Planar sets are aspherical

(dt.: Teilmengen der Ebene sind asphärisch)

Habilitationsschrift

— revidierte Fassung mit integrierten Zeichnungen —

vorgelegt an der

Fakultät für Mathematik

der Ruhr-Universität Bochum

von

Andreas Zastrow

Bochum, 1997–1998

(Zeichnungen erst 1999/2000 integriert)

TABLE OF CONTENTS

0. List of Theorems **2**

Part A: Reduction of the problem to spaces in Normal Form 5

1. The Tame Case **5**

2. First Reduction Steps **7**

3. Further Reduction Steps: Regularizing to Normal Form **10**

4. Relevant Pathologies **16**

5. Combings and Homotopy Combings **23**

6. Proof of the Main Theorem modolo the Combing Theorem **26**

Part B: Construction of a homotopy combing via word sequences 28

1. The (multivalued) Mappings between Paths and Word Sequences . **29**

2. Equivalence classes of word sequences **38**

3. Structure of the Proof of the Combing Theorem **44**

Part C: Proofs of Section B.1 — Properties of the multivalued Reading- and the Realizing-Mapping 45

1. Combinatorial Description of word sequences **45**

2. Reading word sequences from paths **53**

3. Realizing word sequences by paths **63**

4. Some observations on the Reading- and the Realization process ... **77**

5. Splitting word sequences at position sequences **83**

Part D: Proofs of Section B.2 — Topological Meaning of our Equivalence Relations 98

1. Addendums to the combinatorial definitions of our equivalence relations **98**

2. Verifying that the strong equivalance reflects the geometric identity 112

3. Essential definitions for and on the structure of the next subsections 125

4. A shortening algorithm for words 137

5. Compatibility of the shortening algorithms with the elementary processes 143

6. The obstruction to uniqueness of reduced word sequences 157

7. The uniqueness theorem for firm equivalence classes of reduced word sequences 167

8. On Realization Diagrammes 176

9. Combinatorial characterization of nullhomotopy 193

10. Combinatorilly describing corridor entrances and exits 200

11. Analyzing the cancellations when multiplying (reduced) word sequences 207

12. Combinatorial Characterisation of Relative Homotopy and Summary of Part D 218

Part E: Proofs of Section B.3 — Construction of the Lifting by a synchronous parametrization of the Combing Paths .. 221

1. Motivation and initial Definitions and Lemmata 221

2. Construction of the synchronous parametrization of our homotopy combing 226

3. The Basic Set-up for the Continuity proof of the Lifting Theorem 235

4. Preparing the final Estimates for the proof of the Lifting Theorem 244

5. Concluding the proof of continuity of our contraction process 259

Although as long as a book the following text is more written in the style of a paper than in that of a book. The goal is proving the theorem that is stated in the title. Plenty of efford was put in motivating the various steps that have to be performed in the course of that proof. Wherever in that context auxiliary techniques have to be developed, their development is not started before it can be explained why and what for they are needed. For that reason there do not exist introductory chapters that are devoted to building up the corresponding techniques, but the corresponding development happens in all parts of the papers, sometimes just in subparagraphs of lengthy proofs of some lemmata, or as auxiliary definitions within a sequences of propositions that are devoted to prove some lemma. Hence there do not exist "Chapters" in this paper, just Parts which are subdived into sections. The Parts A and B are devoted to introduce, motivate and to expose the structure of the paper, while Parts C–E correspond to the main steps of the proof.

The paper contains a huge variety of cross-references; at almost every place where techniques, ideas or terminology of other parts of the paper are used such references are inserted. Apart from that references to the places of definition of the most common symbols can be found below, for most common phrases they are listed in a register at the end of the paper.

$\overleftarrow{()}$, $\overrightarrow{()}$ C.1.9
$\mathcal{A}^*_?$, $\overline{\mathcal{A}^*_?}$ A.1.4
BD . . . . . B.1.2(i)
$\mathcal{L}$, $\mathcal{L}_?$, $\widetilde{\mathcal{L}}$ . . . A.5.2
$\mathcal{N}$, $\mathcal{N}_?$ B.1.1
non-(Q_η) . . . E.3.3
Q_η E.3.2
$\mathcal{QH}_i$ . . . . A.3.7(iii)
$\widehat{X_x}$, $\widehat{Y_y}$ . . . A.5.2(i)
$0^{\mathbb{N}}$ . . . . . C.5.3(iii)
$\mathcal{A}^{**}$ A.1.6
C_X , $\widetilde{C_X}$ A.5.5
$\mathbb{L}$ C.1.12
$\mathbb{N} = \{1, 2, 3 . . . . . \}$
$\mathcal{PH}$ A.3.1
(Q_η)- E.3.3
$()^{\mathrm{v}}$ . . . B.1.1, B.1.3
Φ . . . . . B.1.6(2)
$\mathcal{A}^{(-1)}_?$, $\overline{\mathcal{A}_?}$ . . . B.1.3
$\mathcal{B}$, $\mathcal{B}_?$ C.1.3
$()^{\mathrm{h}}$ . . . B.1.1, B.1.3
$\ell^{\mathbb{N}}$ . . . . . C.5.3(iii)
$\mathbb{N}_0 = \{0, 1, 2, ... \}$
$\mathcal{P}X$ A.3.4
$\mathcal{QH}$ . . . . . A.3.7(1)
X, $X_?$ A.3.3–4
Ψ . . . . . B.1.6(1)

Planar Sets are Aspherical

— revised version with integrated pictures —

Andreas Zastrow

Abstract:
In order to prove this fact, firstly an embedding theorem is proven: Any compact and totally and locally pathwise connected subset of the plane can be homotopy-equivalently embedded into a "Modified Sierpinski Carpet", i.e. a space which can be obtained by sometimes omitting to take out a square in the scheme of constructing the classical Sierpinski carpet. Such a space offers the opportunity of defining infinite combinatorial objects that surject on the variety of all paths. The corresponding objects are defined and the equivalence relations that (a) corresponds to the kernel of this surjection and (b) describe when two paths are relatively homotopic are described from the combinatorial viewpoint. On a combinatorial basis minimal elements in these equivalence classes are be selected, and by using these minimal representatives a system of paths can be selected the connects each point of the plane in each relative homotopy class to the base point. Such a system of paths provides a natural source for how to contract a continuous image of a higher dimensional sphere. Thanks to the combinatorial control of our system of paths and to the properties of minimal representatives it can be shown that the above construction actually gives a continuous contraction process. This proves nullhomotopy in an arbitrary situation, and so nothing is left that could generate higher homotopy groups. Hence asphericity.

In order to prove the statement of the title, one has to deal with possibly very pathological phenomena. This can partially be done by reducing the situation to spaces in Normal Form (this situation will be reached at the end of Part A already, cf. Thm.B); however the main bulk has to be spend on combinatorially describing the remaining pathology and on analyzing the combinatorial data to such an extend that they give how to build a contracting homotopy for an arbitrary continuous image of a higher dimensional sphere. There are good reasons to believe that this has been a fairly old problem. However, I was unfortunately not able to verify the corresponding rumour in literature.

ACKNOWLEDGEMENT: I thank Professor Epstein for suggesting the subject, for making me acquainted with some of the relevant mathematical theories, and for simplifying my original definition of a homotopic distance. I also wish to thank Lucien Guillou and Alexis Marin of their interest even in fine points of the proof and asking stimulating

The main part of the initial research of this paper was done, while the author was on a two-year-leave to Warwick, England. This leave was supported by the European Community within the 'Human Capital and Mobility'-Programme, No. ERBCHBICT930859.

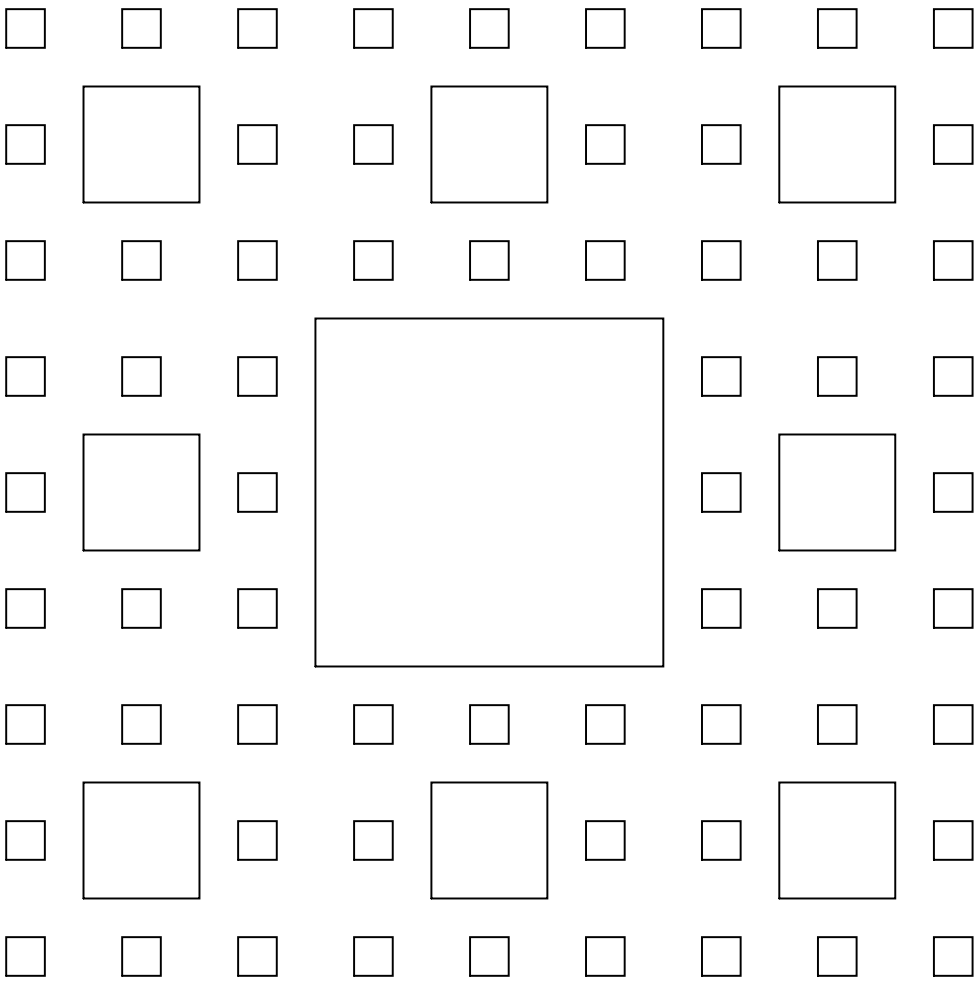


Fig. a (belonging to Theorem B):
The figure pictures the third approximation step of a space which was firstly constructed in [Sier] is nowadays called the Sierpinski Carpet. This space which can be regarded as a two-dimensional version of the Cantor Set is constructed as follows: Take the unit square, impose a (3×3)-subdivision, remove the open middle square of the unit square, (3×3)-subdivide the remaining eight squares, remove from each of them the middle one, and repeat these two steps infinitely many times.

questions. Finally I wish to thank Professor Abresch for improving the presentation of the material by suggesting the structure of this revised version.

§0. List of Theorems

As indicated by the title already, this paper has been written to prove the following theorem that essentially claims that in any subset of the Euclidean plane any continuous

image of a higher dimensional sphere is contractible:

Theorem A: *Let Y be a subset of $\mathbb{R}^2$ with the induced topology and let $y_0 \in Y$. Then $\pi_k(Y, y_0) = 0$ for $k \geq 2$.*

It will take a fairly long way to prove Theorem A, and the proof will not be complete before E.5.11. While doing this, some other results have to be proved as well, some of which might have an interest of their own. They are listed below and here stated in such a way that any use of technical terms as they will be established later is avoided:

Theorem B: *Any compact and totally and locally pathwise connected subset of $\mathbb{R}^2$ can be homotopy equivalently embedded into a space obtained from the Sierpinski Carpet by filling in an appropriate subset of its holes* (cf. Fig.a, A.3.5 and A.3.6(1)).

The proof of Theorem B will use the "Modified Sierpinski Carpet", i.e. a space which has the same universal property as the Sierpinski Carpet but is constructed according to a similar but different scheme of removing holes from a correspondingly subdivided unit square. The Modified Sierpinski Carpet with a filling in of a subset of its holes will be considered as a "space in Normal Form".

Theorem C: *For each subset of $\mathbb{R}^2$ in Normal Form there exits a system of sequences of finite combinatorial objects that can be mapped onto the variety of paths in that space* (cf. B.1.5/B.1.6(2), C.2.1/C.3.1).

These sequences of finite combinatorial objects will be modelled on an alphabet which depends on which of the holes of the Modified Sierpinski Carpet have been filled in and which have not. Based on knowing the appropriate alphabet, the relevant set of sequences can be completely characterized by a finite set of such rules that only apply to one or to two adjacent entries of that sequence and (cf. C.1.5–C.1.9 for the definition of these objects).

The surjective mapping as constructed for Theorem C induces two equivalence relations on our sequences by regarding any two sequences as strongly and as weakly equivalent if they are mapped to the same or to relatively homotopic paths, respectively. The paper will combinatorially characterize these equivalence relations. Their common structure is outlined in the following paragraph:

If two of our sequences ω and ω' are in one of our equivalence relations, then there has to exist a sequence of $(\infty \times k_i)$-matrices ($k_i \in \mathbb{N}$ with $k_i \overset{\text{monotonously}}{\longrightarrow} \infty$ as $i \to \infty$), such that the first and the last column of all of these matrices are the two sequences ω and ω'. The ith and the $(i+1)^{\text{st}}$ matrix have to be in a particular relation that can be described as follows: All columns of the ith matrix appear in the same order in some columns of the $(i+1)^{\text{st}}$ matrix, while the remaining columns can be constructed by using within the i first lines copies of the entries of one of the adjacent columns. In the higher indexed lines of the second type of columns entries may be found that did not appear in the ith matrix. There are two types of rules which have to be fulfilled by such a sequence of $(\infty \times k_i)$-matrices: At first, a finite set of rules has to be satisfied by each (2×2)-submatrix of adjacent entries in the stabilizing part of the sequence. Secondly, certain types of transformations may only occur in a bounded number within the overall system of rows (clearly this number stabilizes for $i \to \infty$ and each fixed row). In this context "transformations" are to be understood as follows: We are thinking of entries of

two neighbouring colums that in the stabilizing part of our $(\infty \times k_i)$-matrices occur in adjacent positions of any of the lines in such a way that we are allowed to "transform" the corresponding left entry into the right entry, and on the technical level the above mentioned restrictions on the (2×2)-matrices describe what kind of transformations are permitted.

Theorem D: *The fact when two of our sequences ω and ω' of combinatorial objects describe the same path in Y can be combinatorially characterized within the framework as outlined above* (cf. B.2.10(ii)/D.12.3)

Theorem E: *A similar combinatorial description exists for describing when two of our infinite sequences of combinatorial objects are mapped to relatively homotopic paths. The corresponding description differs from the one of Thm.D just by weakening the boundedness conditions* (cf. B.2.10(i)/D.12.1).

By subdividing the transformations that occurred in Thm.D and Thm.E into shortening and enlarging ones, the natural concept of reducible and irreducible is established for our infinite systems of combinatorial objects. According to that system and the equivalence relation of Thm.E the following will be proved:

Theorem F: *Each of our sequences is equivalent to an irreducible one. In addition, the irreducible ones of any of the Thm.E-equivalence classes form a subclass according to an equivalence relation that is even stronger as the one described in Thm.D* (cf. B.2.12/D.7.9).

Theorem F can be understood as having characterized "algebraically shortest" paths within each relative homotopy class of paths in a space obtained from the Modified Sierpinski Carpet by filling in an arbitrary subpattern of its holes.

Theorem G: *Let Y be a space as in Thm.B and let $y \in Y$. Let $\widehat{Y_y}$ be the set of all pairs (y', α) such that $y' \in Y$ and such that α is a relative homotopy class of paths that connect y to y'; and let $\mathcal{L}_y(Y)$ be the set of all compact paths in Y with start point y. Consider Y as a metric space, topologize $\mathcal{L}_y(Y)$ by the supremum metric of paths, and metricize $\widehat{Y_y}$ by letting $d((y', \alpha'), (y'', \alpha''))$ be the infimum of all diameters of all subsets of Y that contain a path in the relative homotopy class of $\alpha'^{-1} \cdot \alpha''$. Then there exists a continuous section $\widehat{Y_y} \to \mathcal{L}_y(Y)$* (cf. A.5.2/A.5.5(iii)/A.5.6(iii))

Theorems B–G will be reformulated in this paper, then stated more precisely by using the language as developed by then. In particular *existence claims for sets of rules as made by these Theorems will not just come out in the abstract sense; these rules will be concretely described within this paper.*

0.1 The Skeleton of the Proof of Theorem A can already be guessed from the above theorems:

The pathology that occurs in this paper is tackled twofold: Some pathological phenomena can be circumvented by using essentially topological arguments that reduce the problem to spaces in Normal Form. The necessary arguments are essentially the same that prove Theorem B, and this step of the Proof of Theorem A will already be completed at the end of Part A (cf. A.3.6). The remaining pathologies will then require more effort to be tackled. Roughly speaking, a combinatorial concept of objects that are almost bijectively associated to the system of paths in a space in Normal Form is

build up, and, even though there are uncountable many spaces in Normal Form, our combinatorics will allow to respond individually to each of them. The combinatorial objects will be word sequences (B.1.4), and it will be an important step of this proof to understand that those word sequences which actually can describe paths can be characterized by combinatorial rules that only apply to finite parts of these sequences (C.1.5). Similarly it can be combinatorially described when word sequences describe relatively homotopic paths (B.2.10(i)/B.2.9(i)) and the "almost" in "almost bijective correspondence" can be combinatorially understood (B.2.10(ii)/B.2.9(ii)). Even though the fundamental groups of wild subsets of the plane need not be free any more (even not for the Hawaiian Earrings, cf. A.4.4) and, although the cancelling of inverse letters is not the only equivalence transformation for word sequences (cf. B.2.2), for our infinite combinatorical objects enough analogy to the tame case of the finitely punctured plane will be left for finding almost unique reduced representatives within each relative homotopy class of paths. That way we are able to establish a system of paths that contains for each point in our space in Normal Form and for each relative homotopy class precisely one path that connects it to the base point or our space. Such a system of paths ("homotopy combing", cf. A.5.5(ii)) can then be used as guideline for how to contract an arbitrary higher dimensional sphere within that space. For that purpose the paths of the homotopy combing need to be parametrized (finding the corresponding parametrization will be a step of its own most of which happens in Part E), but it turns out that word sequences provide enough control to construct such a parametrization and to make sure that the accordingly constructed contraction does actually come out as a continuous mapping. This may be understood as the very rough outline of the proof of Theorem A.

Abschnitt A: Reduction of the problem to spaces in Normal Form

§1. The Tame Case

As indicated in the first paragraph of this paper already, the main effort will have to be spend to appropriately handling those subsets of the space which have "pathological" or "wild" phenomena (i.e. are in particular not homeomorphic to a simplicial complex). In the complementary "tame" case the proof of our Theorem A can easily be obtained. It is the purpose of this section to give the corresponding proof for one of the possible assumptions of tameness below:

1.1 Lemma: *If Y is open, then Theorem A holds.*

Proof: Given some continuous map $f : S^k \to Y$ for $k \geq 2$. Since the spheres have trivial fundamental groups in that dimensions, there exists a lift $\widetilde{f} : S^k \to \widetilde{Y}$ to the universal

covering of Y. Since Y was assumed to be open, it can be interpreted as a non-compact Riemann surface, and consequently $\widetilde{Y}$ is a Riemann surface with trivial fundamental group. Hence by the Riemann Mapping Theorem it has to be biholomorphic to $\mathbb{C}$ or to $\mathbb{D}$, and so in any case it is homeomorphic to $\mathbb{R}^2$. It follows that there exists a contraction of $\widetilde{f}$ within $\widetilde{Y}$, and via the covering projection $p : \widetilde{Y} \to Y$ some contraction of f within Y is obtained. Thus there cannot exist a mapping f that could generate a non-trivial $\pi_k(Y)$. □

1.2 Remark on why this proof cannot be extended to the non-tame case:
Hardly any of these arguments of Lemma 1.1 can be used in the general case: Any point in an open set can never be the accumulation point of holes of a space, but this phenomenon can for example be found at the Hawaiian Earrings (cf. 4.5(Fig.b)). In addition, the Hawaiian Earrings are not semilocally simply connected, and thus the classical theory does not permit the construction of a universal covering space. Hence there is not any chance to reduce the general problem to such a powerful tool as the Riemann mapping theorem.
Another gap that obstructs using a classical line of asphericity proofs is the fact that we cannot apply the Eilenberg-Steenrod coincidence theory of homology groups (cf. [E–S].III.10.1). For simplicial complexes this theory guarantees that simplicial and singular homology groups coincide and hence that the higher singular homology groups vanish for dimension reasons because the simplicial groups do. However, since wild subsets of the plane are not homeomorphic to simplicial complexes, we cannot conclude that their higher homology groups have to vanish for dimension reasons. Hence we cannot use the following classical line of an asphericity proof that with some variations is always used when in the course of the proof of the Waldhausen Theorem asphericity has to be shown (cf. [Wald].1.1.5, [ZBG].10.8)):
Use some individual properties of the given space to compute that all lower homology groups vanish for that space or for its universal covering. Now apply Eilenberg-Steenrod and dimension arguments to conclude that the remaining homology groups also vanish. Then use Hurewicz' theorem to conclude that for the space under consideration all homotopy groups vanish, also. Finally, if our space of consideration should have had a non-trivial π_1, this line of arguments could still have been applied by skipping to the universal covering in the first step and by using the invariance of homotopy groups with respect to covering spaces as the last step of this line of conclusions. — As explained before, there are at least two gaps in this classical line of arguments (cf. C.4.10).

1.3 Remark on the basic structure of our Proof of Theorem A (cf. 0.1):
It is desirable to keep in mind the picture of f being retracted inside the Riemann surface Y as in Lemma 1.1, since in principle the paper is about how to construct for a given f a contracting homotopy, even if it will take other methods than just going over a contractible covering space. Such a contraction is explicitly constructed for spaces in Normal Form. The system of paths on which during such a contraction process each point is slid down to the base point will be constructed as a prerequisite for defining the contraction (*"homotopy combing"*, cf. Definition 5.5). Our spaces in Normal Form

are obtained from the unit square by marking a dense pattern of subsquares and by removing an arbitrary subset of these squares. Hence they are essentially determined by naming the system of squares to be removed. This combinatorial information can then be turned into another type of combinatorial information that controls the variety of paths in such a space in normal form (*"word sequences"*, cf. Definition B.1.4). By discussing how homotopy is mirrored in the combinatorial data of word sequences and finding reduced word sequences, we will manage to select the abstract combing paths (cf. B.3.1). By going that way we are essentially showing that the possible pathology of subsets of the plane cannot obstruct that there exists such a contracting homotopy for an arbitrary f (cf. Alexander's Sphere 4.1).

§2. First Reduction Steps

2.1 Lemma: *In order to prove Theorem A one can restrict oneself to considering closed pathwise connected and locally pathwise connected subsets of the plane.*

Proof: Let $Y \subset \mathbb{R}^2$. The statement of Theorem A is equivalent to the assertion that for $k \geq 2$ and an arbitrary continuous mapping $f : S^k \to Y$ there exists a contracting homotopy for f. Note that by elementary topology we have that $f(S^k)$ is compact and totally pathwise connected, in addition the theory of Peano Continua (cf. [Will].(Thm.31.1)&31.4–5) gives that it is locally pathwise connected, also. Since finding a contraction for f inside its image set implies finding one in an arbitrary set, the lemma follows. □

2.2 Remark: Recall that we did not require any pathwise connectivity for our space Y in Theorem A. For a not pathwise connected Y asphericity is to be understood as the vanishing of all homotopy groups π_k of all path-components of Y for $k \geq 2$. However, by Lemma 2.1 the cases which are not pathwise connected need not be considered any further.

2.3 Lemma: *In order to prove Theorem A one can restrict oneself to considering spaces which result from the closed unit square by removing open disjoint squares.*

Proof: Let Y be a compact and totally and locally pathwise connected subset of the plane (cf. Lemma 2.1). The proof will be given by the following propositions 2.5–2.7, where the connected components of $\mathbb{R}^2 - Y$ will be analyzed. It will be proven that they are topological disks with such a tame behavior of their boundary that by retracting these "holes" to square shape and synchronously enlarging the space Y we will end up with some space Y' that is homotopy equivalent to Y:

2.4 Definition: Let Y be a subset of the plane. Then a *"hole of Y"* is a path-component of $\mathbb{R}^2 - Y$.

2.5 Proposition: *Let Y be a compact totally and locally pathwise connected subset of the plane. Then each hole of Y is homeomorphic to D^2, i.e. to the open unit disk (or,*

for the exterior hole, to the once-punctured disk).

Proof: Let G be a path-component of the complement of Y (i.e. what in any part of this paper apart from 2.3–2.7 will be merely considered as a "hole"). For not having to treat the exterior region as a special case, in the following we use the one-point compactification $\mathbb{R}^2 \subset \widehat{\mathbb{R}^2} \approx S^2$. We regard G as being embedded into $\widehat{\mathbb{R}^2}$, and we refer to $\widehat{\mathbb{R}^2}$ as the basic set for all questions of (non)-nullhomotopy in the subsequent proof. The Riemann Mapping Theorem restricts the problem to proving that $\pi_1(G) = 1$. Assume the contrary, i.e. that there exists a non-nullhomotopic path $u : [0,1] \to G$. Since G is open, without loss of generality u is a polygon. Interpret $u([0,1])$ as a graph, and observe that this graph cannot be a tree, since otherwise u would have all of its trace on a tree and thus be nullhomotopic. Hence $u([0,1])$ is imposing a non-trivial cell-subdivision on $\widehat{\mathbb{R}^2}$. At least one of the two-cell boundaries has to be a non-nullhomotopic curve in G, because otherwise the contracting homotopies of all cell-boundaries could be used for homotopying u to a curve onto some tree. Hence by considering an appropriate cell-boundary we find a simple closed polygonal non-nullhomotopic curve $v : [0,1] \to G$. By Schönflies' Theorem v separates $\widehat{\mathbb{R}^2}$ into two regions, each of which has to contain points that do not belong to G, and thus contains points that belong to Y, also. This is a contradiction to the pathwise connectedness of Y, and hence the proposition. □

2.6 Proposition: *Use the same assumptions on Y as in Proposition 2.5. If Y has infinitely many holes, then their diameters form a null-sequence.*

Proof: Assume the contrary, i.e. that the diameters of our holes would not converge to zero. This implies that within the plane there exist sequences of points $(P_i)_{i\in\mathbb{N}}$, $(Q_i)_{i\in\mathbb{N}}$ such that for each $i \in \mathbb{N}$ the two points P_i and Q_i are contained in the same hole G_i such that $\operatorname{dist}(P_i, Q_i) > \varepsilon$ for all i and that $G_i \neq G_j$ for $i \neq j$. We choose polygonal paths $u_i : [0,1] \to \mathbb{R}^2$ with $u_i(0) = P_i$, $u_i(1) = Q_i$ which take on all their values in G_i. By skipping to a subsequence we get that $(P_i)_{i\in\mathbb{N}}$ and $(Q_i)_{i\in\mathbb{N}}$ both converge. The limits are denoted by P and Q, respectively. Without loss of generality we can adjust our coordinate system of the plane in such a way that $P = (0,2)$ and $Q = (0,-2)$. In addition, by truncating finite initial subsequences of our sequence we have that all P_i and all Q_i are contained in $U((0,2),1)$ and in $U((0,-2),1)$, respectively. We denote by R_i the first intersection with the x-axis that u_i sees on its way from P_i to Q_i. The R_i form a bounded sequence on the x-axis, and hence there have to exist accumulation points. We now again skip to a subsequence of $(P_i, Q_i, u_i, R_i)_{i\in\mathbb{N}}$, such that the sequence $(R_i)_{i\in\mathbb{N}}$ converges monotonously. Our next considerations are based on joining the subsegments of u_i, u_{i+1} that connect P_i with R_i and P_{i+1} with R_{i+1}, respectively, with the corresponding segment $[R_i, R_{i+1}]$ on the x-axis, and with some suitable curve that joins P_i with P_{i+1} within $U((0,2),1)$. Hence we get a simple closed curve within the plane which by the Schönflies Theorem bounds a cell Δ_i'. We in particular ask, under what conditions segments of the curves $\{u_j \mid j \in \mathbb{N}\}$ may be found inside Δ_i'. Based on our construction there are just two possibilities for u_i-curves to enter this cell:

- ▷ Within $U((0,2),1)$ the initial segments of other u_j may be found in the interior of Δ_i', but these curves are not of interest for our discussion in the following.
- ▷ There may be some u_j-curves which enter Δ_i' by crossing the segment $[R_i, R_{i+1}]$. By assuming general position, these curves have to leave Δ_i' by another intersection of $[R_i, R_{i+1}]$ again. Hence these u_j-segments may be interpreted as combinatorial semicircles that could be found within Δ_j'. They may be partially nested and partially lined up on $[R_i, R_{i+1}]$.

We now construct another subcell of the plane (this time denoted by Δ_i) whose boundary is given by joining the entire polygons u_i and u_{i+1} with appropriate connections of P_i and P_{i+1} within $U((0,2),1)$ on the one hand, and of Q_i and Q_{i+1} within $U((0,-2),1)$ on the other hand. Based on the above observation we see that there is a possibility to travel from R_i to R_{i+1} via the interior of Δ_i only, namely by using the corresponding segments of the x-axis, and appropriate routes that are parallel to the uppermost combinatorial semicircles. Since such a journey from R_i to R_{i+1} takes us from the interior of one hole of Y to another, we will come across some points belonging to Y while we are on one of the segments of the x-axis. Hence $\exists S_i \in Y \cap [R_i, R_{i+1}] \cap \overset{\circ}{\Delta}_i$. Since Y is compact, $\lim_{i\to\infty}(S_i) = R \in Y$. We now want to find a contradiction by showing that Y cannot be locally connected at R:

Assume that we want to travel by only using points of Y from R to some point S_i in a δ-neighbourhood of R. On such a journey we have to intersect the boundary of Δ_i that on the one hand had been composed by curves u_i that have an empty intersection with Y, and on the other hand by such curves that are entirely contained in $U((0,2),1)$ or in $U((0,-2),1)$. This proves that our journey has to take us at least at distance one apart from the x-axis, and hence there is no chance for Y to be locally pathwise connected at R. □

2.7 Proposition: *Use the same assumptions on Y as in Proposition 2.5. Then Y is a deformation retract of some set Y', where Y' is obtained by removing open squares from a closed square in $\mathbb{R}^2$.*

Proof: Since Y is compact, there exists a big square Q_0 in $\mathbb{R}^2$ containing all of Y. Because the holes G_i of Y are open, each of them contains at least a little square shaped region Q_i. Let Y' be obtained by removing these open regions from Q_0. It remains to be proved that Y is a deformation retract of Y'. I.e. we are essentially left with two tasks:

(i) firstly, to prove that for each hole G_i there exists a deformation retraction that projects $\overline{G_i} - Q_i$ to ∂G_i, and,

(ii) secondly, to prove that by plugging all these deformation retractions together we still get a continuous mapping.

The exterior region does not require separate consideration, since it can be understood as a special kind of hole.

Ad (i): Let $\mathbb{D}$ be the open unit disk and $f_i : \mathbb{D} \to G_i$ a biholomorphic mapping whose existence is guaranteed by the Riemann Mapping Theorem and by 2.5. By

[Pomk].(Thm.2.1), since Y is locally pathwise connected, f_i can be continuously extended to a mapping $f_i : \overline{\mathbb{D}} \to \overline{G_i}$ which imposes continuous coordinates of the unit disk on G_i. These coordinates enable us to retract the once-punctured interior of G_i (or any complement of a tame disk $\subset \overset{\circ}{G}_i$) to ∂G_i.
Ad (ii): Since there may be infinitely many G_i which accumulate at some point of Y, the continuity of a map obtained by plugging together the infinitely many deformation retractions as obtained in (i) is not trivial at such an accumulation point. Nevertheless it can be obtained in our case by using 2.6: Since 2.6 is telling us that almost all of the deformation retractions are moving the points just an arbitrary small distance apart, an ε-δ-estimate as it has to be performed in the course of a standard continuity proof is hardly more difficult than in the case where only finitely many maps are plugged together; and in that case continuity is given by standard arguments. □

Since homotopy groups are invariants with respect to homotopy equivalence, Proposition 2.7 implies Lemma 2.3. □

2.8 Remark on our Normal Form for topological spaces: (cf. Thm.B and 1.3)**:**
A topological space in "Normal Form" is essentially a space like Y' in 2.7. However, for the Normal Form there are additional demands which did not occur in 2.7. Namely, for the unit square a certain dense pattern of possible hole-regions is prescribed (cf. 3.1), and spaces in Normal Form must have all its holes in the areas designed by the pattern for the purpose. I.e. for a space Y in Normal Form a subset of those areas that are listed in the pattern are actually holes, while the other areas and the complementary region of all pattern-areas belong to Y. It will be the purpose of Section §3 to define the pattern and to prove that it only takes a refinement of the arguments of 2.7 to retract the holes of Y is such a way that the resulting space Y' has Normal Form.

§3. Further Reduction Steps: Regularizing to Normal Form

Spaces in Normal Form will be defined as subsets of the unit square $[0,1] \times [0,1] = [0,1]^2 \subset \mathbb{R}^2$.

3.1 Definition of the pattern of possible holes for a space in Normal Form.

$$\begin{aligned}\mathcal{PH} := \{&(\tfrac{2\cdot k-1}{3^i}, \tfrac{2\cdot k}{3^i}) \times (\tfrac{2\cdot m-1}{3^i}, \tfrac{2\cdot m}{3^i}) \mid i,k,m \in \mathbb{N}, \\ &k,m \leq 3^i/2\,,\ \nexists i', k', m' \in \mathbb{N},\ k', m' < 3^i/2\,,\ i' < i \text{ such that} \\ &(\tfrac{2\cdot k-1}{3^i}, \tfrac{2\cdot k}{3^i}) \times (\tfrac{2\cdot m-1}{3^i}, \tfrac{2\cdot m}{3^i}) \subset (\tfrac{2\cdot k'-1}{3^{i'}}, \tfrac{2\cdot k'}{3^{i'}}) \times (\tfrac{2\cdot m'-1}{3^{i'}}, \tfrac{2\cdot m'}{3^{i'}})\}.\end{aligned}$$

For any $\mathcal{PH}$-square $(\frac{2\cdot k-1}{3^i}, \frac{2\cdot k}{3^i}) \times (\frac{2\cdot m-1}{3^i}, \frac{2\cdot m}{3^i})$ the index i is regarded as the *"level-index"* of that area.

3.2 Properties of $\mathcal{PH}$:
The following are immediate:
(i) Any closures of two different $\mathcal{PH}$-areas are disjoint.

(ii) Each point on the boundary of a $\mathcal{PH}$-area is the accumulation point of an infinite sequence of other $\mathcal{PH}$-areas.

(iii) The system of $\mathcal{PH}$-areas is dense, i.e. any neighbourhood of any $P \in [0,1]^2$ intersects at least one $\mathcal{PH}$-set.

3.3 Definition of a space in Normal Form:
X is called a *"Space in Nomal Form"* if it is obtained from $[0,1]^2$ by removing some of the open squares that are contained in the system $\mathcal{PH}$. Any (finite or infinite) subset of $\mathcal{PH}$ may be removed; even the spaces $[0,1]^2$ (for the empty subset) and the space where all $\mathcal{PH}$-squares are removed are also considered as spaces in Normal Form. The latter space is called the *"Modified Sierpinski Carpet"* (cf. 3.5).

3.4 Additional Notation for spaces in Normal Form: Let $\mathcal{P}X$ be the set of holes (the exterior one excluded) of X, i.e. $X = [0,1]^2 - \{x \mid \exists Q \text{ with } x \in Q \in \mathcal{P}X\}$. Denote by X_i the space which is obtained from X by filling in those holes which have an index that is bigger than i, i.e. $X_i = [0,1]^2 - \{x \mid \exists Q \text{ with } x \in Q \in \mathcal{P}X \text{ and } \mathrm{ind}(Q) \leq i\}$. Hence $X_0 = [0,1]^2$. Since $\mathcal{PH}$ contains only finitely many areas for each index, a space X_i can only have finitely many holes and hence is of polyhedral type. By definition $\bigcap_{i=0}^{\infty} X_i = X$. By considering the natural inclusions $X_{i+1} \hookrightarrow X_i$ we see that X can also be described as the inverse limit of the X_i. In addition, regarding X as a limit-object and the X_i as the approximating spaces makes us think that in the course of some approximation process the finer holes become some later state visible as the bigger holes and, when they become visible, then suddenly "crop up".

3.5 Remark: In 1916 Sierpiński had in [Sier] introduced a similar scheme for removing a set of open squares from the unit square, and the resulting space is nowadays called the "Sierpinski Carpet" (cf. 0.0(Fig.a), 3.3 and [PJS].§2.2(Fig.2-20)). In 1958 Whyburn characterized the universal property of the Sierpinski Carpet (cf. [Why].(Thm.3&4) and [C–P].(page 36) for the meaning of the word "curve" in [Sier] and [Why]) and from Whyburn's work it follows that our Modified Sierpinski Carpet has the same universal property. By Whyburn's result we obtain equivalent statements if we substitute "Sierpinski Carpet" by "Modified Sierpinski Carpet" in Thm.B, and this observation is a part of the proof of our Thm.B. For the purposes of this paper the modified Sierpinski Carpet was chosen, since, even though it takes more effort to describe the pattern of holes to be removed as for the Sierpinski Carpet, it has the more symmetric position of holes and hence it is more suitable for the remaining constructions, in particular for putting a corridor network on spaces in Normal Form (cf. B.1.1).

3.6 Lemma: *In order to prove Theorem A one can restrict oneself to considering spaces in Normal Form.*

Proof: By virtue of 2.1 it suffices to construct a homotopy equivalence between an arbitrary compact and locally and totally pathwise connected subset Y of the plane, and a space X in Normal Form. This will be done in two steps according to

$$Y \xrightarrow{\varphi} X' \xrightarrow{\psi} X, \tag{1}$$

where φ will be a homotopy equivalent embedding, ψ be a homeomorphism and X' be a space in Prenormal Form. φ and ψ will be defined below (cf. 3.9, 3.8), after having defined the *Prenormal Form* in 3.7. Since, in particular, the proof of 3.9 will be lengthy, the proof of 3.6 will not be complete before 3.12.

3.7 Definition:

(i) Let $J := \{\frac{j}{3^i} \mid i,j \in \mathbb{N}_0 \ , \ 0 \le j \le 3^i\}$ and let $\{a_j \mid j \in J\}$ and $\{b_j \mid j \in J\}$ be two sets of nodes satisfying that $0 = a_0 = b_0 \ , \ 1 = a_1 = b_1 \ , \ a_{j'} < a_j$ and $b_{j'} < b_j$ for $j' < j$ and that $|a_{k/3^i} - a_{m/3^i}| < |k-m| \cdot (\frac{3}{4})^i \ , \ |b_{k/3^i} - b_{m/3^i}| < |k-m| \cdot (\frac{3}{4})^i$ for all naturals $k, m, i \in \mathbb{N}$. For such two systems of nodes we define the "pattern of possible holes of spaces in Prenormal Form" in analogy to 3.1:

$$\begin{aligned} \mathcal{QH} := \{&(a_{(2\cdot k-1)/3^i}, a_{2\cdot k/3^i}) \times (b_{(2\cdot m-1)/3^i}, b_{2\cdot m/3^i}) \mid \ i,k,m \in \mathbb{N}, \\ &k,m \le 3^i/2 \ , \ \nexists i', k', m' \in \mathbb{N} \ , \ k', m' < 3^i/2 \ , \ i' < i \text{ such that} \\ &(\tfrac{2\cdot k-1}{3^i}, \tfrac{2\cdot k}{3^i}) \times (\tfrac{2\cdot m-1}{3^i}, \tfrac{2\cdot m}{3^i}) \subset (\tfrac{2\cdot k'-1}{3^{i'}}, \tfrac{2\cdot k'}{3^{i'}}) \times (\tfrac{2\cdot m'-1}{3^{i'}}, \tfrac{2\cdot m'}{3^{i'}})\}. \end{aligned} \tag{1}$$

(ii) X' is called a "Space in Prenormal Form", if it is obtained from $[0,1]^2$ by removing some of the open squares that are contained in a system $\mathcal{QH}$ that satisfies all the conditions of (i). In the previous sentence "some" is to be understood as in 3.3.

(iii) Sometimes a space which has Prenormal Form with respect to an area system $\mathcal{QH}$ (cf. (i)) will have to be approximated by spaces whose holes are at slightly different coordinates than $a_{k/3^i}, b_{m/3^i} \quad (0 \le k,m \le 3^i \ , \ i \in \mathbb{N})$, namely at coordinates $a_{k,i}$, $b_{m,i}$ where $\lim_{n\to\infty}(a_{k\cdot 3^n, i+n}) = a_{k/3^i}$ and $\lim_{n\to\infty}(b_{m\cdot 3^n, i+n}) = b_{m/3^i}$. The areas that are bounded by such approximating coordinates of a fixed level index are then gathered together in systems that are denoted by

$$\begin{aligned} \mathcal{QH}_i := \{&(a_{(2\cdot k-1)\cdot 3^n, i'+n}, a_{2\cdot k\cdot 3^n, i'+n}) \times (b_{(2\cdot m-1)\cdot 3^n, i'+n}, b_{2\cdot m\cdot 3^n, i'+n}) \mid \\ &n + i' = i \ , \ k,m \in \mathbb{N} \ , \ 1 \le k,m < 3^{i'}/2 \ , \ n \in \mathbb{N}_0 \ , \\ &\nexists k', m', i'' \text{ with } n \in \mathbb{N}_0 \ , \ i'', k', m' \in \mathbb{N} \ , \text{ such that} \\ &(\tfrac{2\cdot k-1}{3^{i'}}, \tfrac{2\cdot k}{3^{i'}}) \times (\tfrac{2\cdot m-1}{3^{i'}}, \tfrac{2\cdot m}{3^{i'}}) \subset (\tfrac{2\cdot k'-1}{3^{i''}}, \tfrac{2\cdot k'}{3^{i''}}) \times (\tfrac{2\cdot k'-1}{3^{i''}}, \tfrac{2\cdot k'}{3^{i''}})\} \end{aligned}$$

3.8 Proposition: *Any space X' in Prenormal Form is homeomorphic to some space X that has Normal Form.*

Proof: The homeomorphism $\psi : X' \to X$ is induced by mapping $(a_j, b_{j'}) \mapsto (j, j')$ (cf. 3.7(i)). Since the node systems $\{a_j \mid j \in J\}$ and $\{b_j \mid j \in J\}$ are dense and monotonous, the above above description can uniquely be extended to an autohomeomorphism of $[0,1]^2$. By using the coinciding combinatorial patterns of the area-systems $\mathcal{PH}$ and $\mathcal{QH}$ (cf. 3.1 and 3.7(i)), this map also gives a homeomorphism $X' \to X$, if X is defined as that space in Normal Form, that has the combinatorially analogous hole system as X'. □

Proposition 3.8 gives one part of the proof of Lemma 3.6; the following proposition will then be devoted to construct the mapping φ of 3.6(1). However, the proof of this proposition will be more lengthy and also require the subsequent propositions 3.10–3.11.

3.9 Proposition: *Any compact and totally and locally pathwise connected subset Y of the plane can be homotopy equivalently embedded into some space X' that has Prenormal Form.*

The proof of this proposition will use the same arguments as for Proposition 2.7, however, it requires additional care to make sure that the coordinates of the squares to which the holes are retracted satisfy all demands of 3.7(i). However, the arguments why all holes can be retracted to square shape need not to be repeated here. Since a contracting similarity can always be added to some φ, other than in 2.7 we now assume that Y is already contained inside the unit square so that $X'_0 = [0,1]^2$.
Similarly as in 3.4 for our spaces in Normal Form the desired space X' will arise as $\bigcap_{i=0}^{\infty} X'_i$. However unlike as in 3.4, the approximating spaces X'_i will not be constructed out of X'_{i-1} by only making the additional holes of index i, but also by possibly shrinking holes of indexes less than i to some smaller size. For that purpose, apart from the nodes a_j and b_j that give the coordinates of the holes of X', we will for each of the pairs (k,i) and (m,i) that satisfy the conditions of Definition 3.7(i) also define approximating nodes $a_{(2\cdot k-1)\cdot 3^n,i+n}$, $a_{2\cdot k\cdot 3^n,i+n}$, $b_{(2\cdot m-1)\cdot 3^n,i+n}$ and $b_{2\cdot m\cdot 3^n,i+n}$ $(n \in \mathbb{N}_0)$ as in 3.7(iii) that satisfy the monotony conditions that guarantee shrinking holes (cf. 3.10(ii)).
Having established this notation for the coordinates of the holes of the approximating spaces $X'_i = [0,1]^2 - \{x \mid \exists Q \text{ with } x \in Q \in \mathcal{QH}_i\}$, we can now proceed by explaining our process for how to choose the desired coordinates; this will be an induction over i. The property

$$0 < a_{2\cdot k,i} - a_{(2\cdot k-1),i} < (\tfrac{3}{4})^i \tag{1}$$

will be ensured during the construction. The zero-step ($i = 0$) of this induction is trivial and given by considering the unit square where the outside boundaries

$$0 = a_0 = a_{0,i} = b_0 = b_{0,i} \text{ and } 1 = a_1 = a_{3^i,i} = b_1 = b_{3^i,i} \tag{2}$$

are the only nodes to be considered.
Hence we may now assume that up to a certain level i_0 nodes $a_{k,i}$ and $b_{m,i}$ have already be chosen such that for the relevant (cf. 3.7(iii)) triples (k,m,i) with $i \leq i_0$ the regions $(a_{2\cdot k-1,i}, a_{2\cdot k,i}) \times (b_{(2\cdot m-1),i}, b_{2\cdot m,i})$ are either squares to which our holes can be contracted, or that these regions entirely belong to our space X'_{i_0}.
Then for all integers k with $1 \leq k \leq 3^{i_0}$ and all $\nu \in \{0,1,2,3,4,5,6\}$ we draw the lines $\{(x,y) \mid x = \frac{\nu}{6}\cdot a_{k-1,i_0} + \frac{6-\nu}{6}\cdot a_{k,i_0}\}$ and the lines $\{(x,y) \mid y = \frac{\nu}{6}\cdot b_{k-1,i_0} + \frac{6-\nu}{6}\cdot b_{k,i_0}\}$ into the unit square. For the following steps we restrict our considerations to the subgrid of lines with even k and $\nu = 3$, and with odd k and $\nu \in \{1,5\}$. We surround each of these subgrid-lines by an ε_{i_0}-strip, where ε_{i_0} is chosen so small that the outward boundaries of two adjacent ε_{i_0}-strips are not further apart than $(\frac{3}{4})^{i_0+1}$. By Proposition 3.11 there exist η_{i_0} and δ_{i_0} determining the coordinates of a $(2\cdot\delta_{i_0})$-strip inside each ε_{i_0}-strip, such that each of the intersection regions of those $(2\cdot\delta_{i_0})$-strips either entirely belongs to X'_{i_0} or entirely belong to the complement of X'_{i_0}. Recall that our task of approximating a space whose holes have to be chosen according to 3.7(i) implies that

each of the regions $(a_{2\cdot k-1,i_0+1}, a_{2\cdot k,i_0+1}) \times (b_{2\cdot k-1,i_0+1}, b_{2\cdot k,i_0+1})$ must either entirely belong to X'_{i_0} or entirely to the complement. However, by having found $(2\cdot\delta_{i_0})$-strips that intersect in a similar way, we now know that nodes $a_{?,i_0+1}$ and $b_{?,i_0+1}$ with the corresponding properties do exist. For the purposes of our construction we choose for any even $k \in \{0, 1, 2, \dots 3^{i_0+1} - 1\}$ our new nodes as follows

(i) $a_{3\cdot k+1,i_0+1} \in [\frac{4}{6}\cdot a_{k,i_0} + \frac{2}{6}\cdot a_{k+1,i_0}, \frac{3}{6}\cdot a_{k,i_0} + \frac{3}{6}\cdot a_{k+1,i_0} + \eta_{i_0} + \delta_{i_0}]$ as small as possible,

(ii) $a_{3\cdot k+2,i_0+1} \in [\frac{3}{6}\cdot a_{k,i_0} + \frac{3}{6}\cdot a_{k+1,i_0} + \eta_{i_0} - \delta_{i_0}, \frac{2}{6}\cdot a_{k,i_0} + \frac{4}{6}\cdot a_{k+1,i_0}]$ as big as possible,

(iii) $a_{3\cdot k+3,i_0+1} \in [a_{k+1,i_0}, \frac{5}{6}\cdot a_{k+1,i_0} + \frac{1}{6}\cdot a_{k+2,i_0} + \eta_{i_0} + \delta_{i_0}]$ as small as possible,

(iv) $a_{3\cdot k+4,i_0+1} \in [\frac{5}{6}\cdot a_{k+1,i_0} + \frac{1}{6}\cdot a_{k+2,i_0} + \eta_{i_0} - \delta_{i_0}, \frac{4}{6}\cdot a_{k+1,i_0} + \frac{2}{6}\cdot a_{k+2,i_0}]$ as big as possible,

(v) $a_{3\cdot k+5,i_0+1} \in [\frac{2}{6}\cdot a_{k+1,i_0} + \frac{4}{6}\cdot a_{k+2,i_0}, \frac{1}{6}\cdot a_{k+1,i_0} + \frac{5}{6}\cdot a_{k+2,i_0} + \eta_{i_0} + \delta_{i_0}]$ as small as possible,

(vi) $a_{3\cdot k+6,i_0+1} \in [\frac{1}{6}\cdot a_{k+1,i_0} + \frac{5}{6}\cdot a_{k+2,i_0} + \eta_{i_0} - \delta_{i_0}, a_{k+2,i_0}]$ as big as possible.

$a_{?,i_0+1}$-values whose first index is bigger than or equal to 3^{i_0+1} are not chosen.

Analogous formulae are used to define the b_{m,i_0+1}.

Our continuing to define nodes $a_{?,i'}$ and $b_{?,i'}$ for $i' > i$ does not introduce additional subregions to the existing $\mathcal{QH}_{i'}$-regions, even though for suitable k' and m' the crossing region $(a_{2\cdot k'-1,i'}, a_{2\cdot k',i'}) \times (b_{2\cdot m'-1,i'}, b_{2\cdot m',i'})$ will be a subregion of $(a_{2\cdot k-1,i_0}, a_{2\cdot k,i_0}) \times (b_{2\cdot m-1,i_0}, b_{2\cdot m,i_0})$. However since inside a region that belongs to X'_{i_0} holes cannot be found (and vice versa), our iterative process of detecting holes and retracting them into square shape will never introduce substructures into such a crossing region like $(a_{2\cdot k-1,i_0}, a_{2\cdot k,i_0}) \times (b_{2\cdot m-1,i_0}, b_{2\cdot m,i_0})$; it will restrict itself to readjusting the boundaries of such a region according to $a_{k,i} \mapsto a_{3\cdot k,i+1} \mapsto a_{3^2\cdot k,i+2} \mapsto \dots$ and to $b_{k,i} \mapsto b_{3\cdot k,i+1} \mapsto b_{3^2\cdot k,i+2} \mapsto \dots$. Thus we see that in that way our process is compatible with the combinatorial structure of an area system of a space in Normal Form or in Prenormal Form, where any two areas of such a system are never contained in each other or do have any overlaps.

Since (1) guarantees that the systems of $a_{?,?}$- and $b_{?,?}$-nodes are dense and since the complement of X' is open, all holes of X' are detected and are "integrated" into our pattern that way. Hence in order to complete the proof of this proposition and thus of Lemma 3.6, we just have (apart from proving 3.11) to show that the above constructions make sure that the $a_{?,?}$ and $b_{?,?}$ do well-behave, i.e. that their limits exist and define a non-degenerate X'. The latter task will be completed by proving the items (iv) and (v) of the following proposition. Thus the proof of 3.9 is complete apart from the addenda 3.10–3.12. □

3.10 Proposition: *The nodes $a_j, b_j, a_{k,i}$ and $b_{k,i}$ as inductively defined in the course of the proof of Proposition 3.9 satisfy the following properties:*

(i) $a_{k',i} < a_{k,i}$ and $b_{k',i} < b_{k,i}$ if and only if $k < k'$.

(ii) For all pairs (k, i) and (m, i) as in 3.7(i) we have that $(a_{(2\cdot k-1)\cdot 3^n,i+n})_{n\in\mathbb{N}}$ and $(b_{(2\cdot m-1)\cdot 3^n,i+n})_{n\in\mathbb{N}}$ are monotonously increasing sequences and that $(a_{2\cdot k\cdot 3^n,i+n})_{n\in\mathbb{N}}$, $(b_{2\cdot k\cdot 3^n,i+n})_{n\in\mathbb{N}}$ are monotonously decreasing.

(iii) All sequences as considered in (ii) do converge.

(iv) $a_{j'} < a_j$ *and* $b_{j'} < b_j$ *for* $j' < j \in J$ *(cf. 3.7(i)).*
(v) $0 < a_{2\cdot k/3^i} - a_{(2\cdot k-1)/3^i} < 3\cdot(\frac{3}{4})^i$

Proof: (i) and (ii) are immediate from the construction, (iii) then follows since by the inequalities of (ii) each of these sequences is monotonous but bounded by any neighbouring grid-line for some fixed level-index i. (iv) can be obtained as follows, without loss of generality let k be odd: By (i)–(ii) we have that $a_{k\cdot 3^n,i+n} < a_{(3\cdot k+1)\cdot 3^{n-1},i+n} \leq a_{3\cdot k+1,i+1} < a_{3\cdot k+2,i+1} \leq a_{(3\cdot k+2)\cdot 3^{n-1},i+n} < a_{(k+1)\cdot 3^n,i+n}$, and hence we see that $a_{3\cdot k+2,i+1} - a_{3\cdot k+1,i+1}$ is a lower bound for the difference of the outer members of our inequality and thus for $a_{(k+1)/3^i} - a_{k/3^i}$ which arise as their limits, too. Finally (v) follows from the following observation: By 3.9(1) we have that $a_{k+1,i} - a_{k,i} \leq (\frac{3}{4})^i$. If now we want an estimate for $a_{(k+1)/3^i} - a_{k/3^i}$ and hence skip to considering $a_{(k+1)\cdot 3^n,i+n} - a_{k\cdot 3^n,i+n}$, the values of $a_{(k+1)\cdot 3^n,i+n}$, $a_{k\cdot 3^n,i+n}$ may have gone into the adjacent strips of the grid of the level index i, but not further beyond. Hence for the limits a similar estimate as 3.9(1) if weakened by an additional factor "3", holds. □

3.11 Proposition: *Let* $a_{k,i_0,\nu}$ *and* $b_{m,i_0,\nu'}$ *denote the coordinates of the grid-lines that have in the course of the iterative process of 3.9 be surrounded by an* ε_{i_0}*-neighbourhood. Then there exists* $\delta_{i_0} > 0$, η_{i_0} *and* η'_{i_0} *which do not depend on* k, m, ν *and* ν' *such that* $|\eta_{i_0}| + \delta_{i_0} < \varepsilon_{i_0}$, $|\eta'_{i_0}| + \delta_{i_0} < \varepsilon_{i_0}$ *and such that each of the regions* $[a_{k,i,\nu} + \eta_{i_0} - \delta_{i_0}, a_{k,i,\nu} + \eta_{i_0} + \delta_{i_0}] \times [b_{k,i,\nu} + \eta'_{i_0} - \delta_{i_0}, b_{k,i,\nu} + \eta'_{i_0} + \delta_{i_0}]$ *either does or does not belong to* X'_{i_0}.

Proof: For the purposes of this proof we might simplify our notation which in particular implies getting rid of the level indices. I.e. we just consider a subset Y of $[0,1]^2$ and assume that a finite number of horizontal and vertical ε-strips are marked within $[0,1]^2$. Use C to denote the complement of Y within $[0,1]^2$. By natural translations the crossing regions of all ε-strips can be identified, and the resulting space will be denoted by $[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}] \times [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$. The intersection of $\overset{\circ}{Y} \cup C$ with the crossing regions of ε-strips defines an open and dense set in each of the crossing regions, and hence for $[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}] \times [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$ finitely many of these open and dense sets are defined. By Baire's Result the intersection of these open and dense sets is still dense. In our concrete situation of a finite intersection, this gives a non-empty open set within $[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}] \times [-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}]$ that has to contain a square shaped subset $[\eta - \delta, \eta + \delta] \times [\eta' - \delta, \eta' + \delta]$. These are the desired coordinates to be constructed that define vertical and horizontal $(2\cdot\delta)$-strips with the desired crossing behavior. For, by construction, points of ∂Y cannot be contained in the $(2\cdot\delta)$-square in any of the crossing regions, and hence for any of those the $(2\cdot\delta)$-square has entirely to be covered either by $\overset{\circ}{Y}$ of by C. □

3.12 Remark: The purpose of this remark is pointing out one particular aspect that might occur in the course of the iterative construction as performed in the proof of 3.9. Recall that in that process we made sure that each region $(a_{2\cdot k-1,i}, a_{2\cdot k,i}) \times (b_{2\cdot m-1,i}, b_{2\cdot m,i})$ either entirely does belong to X'_{i-1} or to its complement. It now may happen that one of the holes of X'_{i-1} covers more than one of these regions. Hence when by using the techniques of 2.7 such an X'_{i-1}-hole is retracted so some X'_i-hole,

only one of the corresponding $(a_{2\cdot k-1,i}, a_{2\cdot k,i}) \times (b_{2\cdot m-1,i}, b_{2\cdot m,i})$-regions will actually become a hole of X_i', the other(s) of them will then entirely belong to X_i'. Hence within X_i' some of these regions may end up with another type than at the time where the corresponding nodes had been adjusted. Due to that phenomenon for a fixed level index i the retraction of our holes into square shape must not be performed simultaneously, but consecutively for each of the holes. This was the last addendum to the proof of 3.8; hence by now this proposition and with it Lemma 3.6 are completely shown. □

§4. Relevant Pathologies

The purpose of this section is to discuss certain examples for pathological situations in order to get some awareness for the variety of phenomena that have to be dealt with in this paper. Our particular interest will be understanding those phenomena that might still occur in spaces in Normal Form.

4.1 Example: The purpose of the first example is just to recall the fact that there are theorems in topology that just become false statements if one deletes the tameness conditions. The most famous of these observations is due to Alexander 1924. In [Alx1] he proved that if one tamely embeds a two-sphere S^2 into $\mathbb{R}^3$, then any curve in $\mathbb{R}^3$ which is disjoint from this S^2 can be contracted without touching the embedded S^2 during the entire homotopy. However, in the same journal in [Alx2] he also presented his famous example of a horned sphere (for a figure see [HO–Y].§4.6), i.e. he found a way of embedding S^2 into $\mathbb{R}^3$ with so much pathology that there exist curves in the complement of this embedded S^2 which cannot be contracted without intersecting this S^2 during the contraction process.

4.2 Remark: Examples like 4.1 are not restricted to dimensions three and higher; here is a two-dimensional theorem where the tameness conditions cannot be abandoned: "The measure of the image-domain of any piecewise smooth curve in the plane is zero". Without tameness-conditions such as "piecewise smoothness" Peano Curves would give counter-examples.

Recall (1.3) that we essentially want to prove Theorem A by as a first step defining a contracting homotopy for a space in Normal Form explicitly by constructing the system of paths (cf. 5.5) along which each point is slid down to the base point. Of course, such a construction can only yield the desired results, if the path-system satisfies the appropriate continuity conditions. The purpose of the next example is to remind us that even without continuity assumptions the existence of such a path-system is not immediate in a non-tame context:

4.3 Example: Consider an arbitrary pathwise connected space Y_0 and try to find a "prefix-closed combing" i.e. a system of paths $\{c_y \mid [0,1] \to Y$ such that $c_y(0) = y_0$, $c_y(1) = y\}$ where $y_0 \in Y$ is a base point, and up to reparametrization $c_y|_{[0,t]}$ coincides with the combing path $c_{c_y(t)}$ for all c_y and all $t \in [0,1]$. The latter property is what is called "prefix-closed" (cf. [ECHLPT].(Def.1.2.10)). The existence of a mere combing

(without requiring any properties of continuity or prefixed-closedness) is trivial by the axiom of choice; however requiring only prefixed-closedness makes finding a combing a non-trivial task. In a simplicial complex one can, simplex by simplex, choose directions for how to comb, and one will always find a prefix-closed one. On the other hand, observe that $((\mathbb{R}\times\mathbb{Q})\cup(\mathbb{Q}\times\mathbb{R}))\cap([0,1]\times[0,1]) =: Y$ has by $\overline{(0,0)\ (x,0)}\cup\overline{(x,0)\ (x,y)}$ a partial prefix-closed path-system which does not reach the points of Y with irrational x-component, but which cannot be extended since this would imply finding the closest rational next to some irrational. Thus we see that by the standard applications of Zorn's Lemma the existence proof for a prefix-closed combing cannot be extended from the tame to the non-tame context. Hence, even though for that particular space a prefix-closed combing can be constructed by using the countability of $\mathbb{Q}$, this example should justify the care and effort that this paper will spend on proving the existence of a continuous homotopy combing for those spaces that have Normal Form.

By having to construct path-systems as in 4.3, some understanding of the structure of fundamental groups of non-tame spaces is helpful. The purpose of the following example is to recall the structure of the simplest space with accumulation phenomena, whose fundamental group has been successfully described since the fifties (e.g. [Gr2].(Thm.6.3), [Higm].(Proof of Thm.6), [MM].(Thm.4.1), [dSM].§3, [Eda].(Thm.A.1), [C–C], [B–S], [Zas4]).

4.4 Example: The *"The Hawaiian Earrings"* Z_0, cf. 4.5(Fig.b), are a topological space whose fundamental group is usually described as a subgroup of the inverse limit $\varprojlim(F_m)$ of the finitely generated free groups where $F_m = \langle \alpha_1,\dots,\alpha_m \mid \text{—} \rangle$ such that α_i corresponds to the ith loop of the Hawaiian Earrings. This inverse-limit-description implies that each homotopy class of a path inside Z_0 can be represented as a sequence of group-elements, the ith entry of that sequence taken from the group F_i. In [Zas2].§2 a slightly different combinatorial concept of word sequences has been suggested for describing this group. This concept is based on the idea that one should resist representing each group-element of F_i by a reduced word and use non-reduced words also, in order to end up with a sequence where the ith word is derived from the $(i-1)^{\text{st}}$ word by only inserting letters $\alpha_i^{\pm 1}$ into the $(i-1)^{\text{st}}$ word. This turns out to be of advantage when paths and homotopies have to be constructed out of combinatorial data, since the combinatorial data of a word sequence can directly be turned into a sequence of paths that form a uniform Cauchy-sequence. This does not hold for the classical description of inverse-limit elements by sequences of reduced words, since for $i<j$ a substring $\alpha_i\alpha_j\alpha_i^{-1}$ of the jth word will just be represented by the empty string according to the classical description, independently of how big $j-i$ is.

Unfortunately the comparatively easy description of the fundamental group of the Hawaiian Earrrings has been using the fact that this space has one accumulation point, only. However, even if we just look at the second simplest example, we experience the following:

4.5 Example: Consider the space Z_1 which is pictured in Fig.b, i.e. two Hawaiian earrings which are joined by a line. In this space a continuous path can run over any combination of the loops of any of the two Hawaiian Earrings (finitely many times over

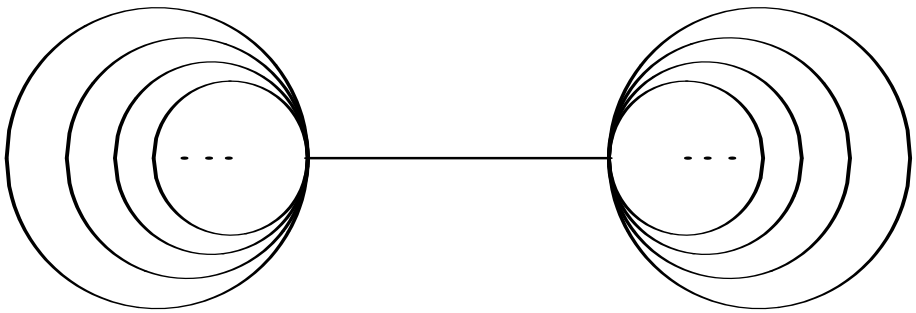


Fig. b (belonging to 4.5):
This picture shows the topological spaces Z_1 which arises as a union of two Hawaiian Earrings Z_0 and of a line joining their accumulation points.

each of the loops, only), but it can only run a finite number of times between the rings. As a consequence of this, while for the Hawaiian Earrings we are able to arrange any letters in any order that we liked in a word sequence, (as long as we made sure that each word is finite and do respect the insertion condition (cf. 4.4)), a similar concept of word sequences as in 4.4 that associates each loop with a generator cannot declare those word sequences as legitimate that would force a path to run infinitely many times between the different Z_0-subspaces. In particular, these conditions cannot be phrased in such a way that each of them applies to a finite part of the word sequences, only, since still any finite initial sequence of words respecting the insertion structure may be part of a legitimate word sequence. Hence for spaces in Normal Form it seemed desireable to have another concept of word sequences that is based on combinatorial rules that only apply to finite aspects of word sequences. Consequently, is was decided to build up a new concept of word sequences in this paper. In addition, spaces in Normal Form are in general much more complicated then Z_1, for they may have so many holes that there are in principle infinitely many different Hawaiian Earrings embedded into such a space.

For purposes explained in 4.5 we will build up another concept of word sequences for describing the variety of paths in spaces in Normal Form. This concept will be less that of a fundamental group, but more of the fundamental groupoid:

4.6 On the idea of the concept of word sequences as used in this paper:

(i) Brief description of the meaning of our words: We will call the rectangular region between two adjacent holes a *"corridor"*. Thus two opposite sides of our rectangle coincide with boundary lines of hole, and according to our new concept of word sequences each letter will imply that our path *"crosses* such a *corridor"*, i.e. goes all the way through the corridor so as to connect its other pair of opposite boundary lines. For this concept of word sequences again not any combination of letters is valid, but the corresponding new rules are combinatorially manageable, since they do not rely on accumulation phenomena, but only apply to a finite subpattern of a word sequence. Essentially just the fact whether two corridors are situated next to each

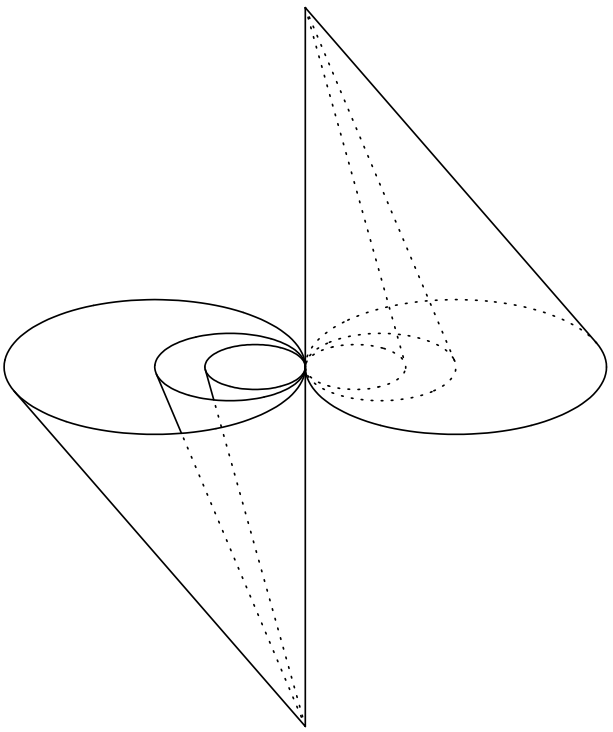


Fig. c (belonging to 4.7):
This picture shows a one-point union of two coned Hawaiian Earrings and presents them as an example where an infinite composition of nullhomotopic paths generates a non-nullhomotopic path (cf. [Zas1].(0.12&3.11). This Example goes back to [Gr1].(3.4).

other in $\mathbb{R}^2$ gives whether the corresponding letter can or cannot appear in adjacent positions within the word of a realizable word sequence.

(ii) Brief description of the internal structure of our word sequences: Another consequence of working with this new concept of word sequences will be that the word sequences now will not have an insertion structure any more, since the entry "corridor C_1 is crossed" is not suitable for being just copied on all sufficiently fine levels as it was for the Hawaiian Earrings the entry "loop α_1 is passed". Hence we will have to draw finer corridors into the areas that are covered by bigger corridors, and in the higher indexed words we will replace the C_1-entry by the crossing records of those finer corridors. Again the rules which letters may appear as a replacement of which letters in a legitimate description of a path do not refer to accumulation phenomena, but only to which fine corridors can be found inside the areas of the less fine ones. In particular these rules do only involve two adjacent words of a word sequence.

4.7 Example: Consider Fig.c which shows the one-point union of two Hawaiian Earrings, each of which is homotopically trivialized by having a cone attached to it. Define a continuous curve as alternately running over each of the loops of the two Hawaiian Earrings. This curve w can be regarded as an infinite composite of curves each of which is homotopically trivial. On the other hand it is shown in [Zas1].(0.12&3.11), that the resulting curve is not null-homotopic. If such phenomena as in Fig.c would occur in the plane also, our concept of proving Theorem A would be fated to fail, since essentially word sequences just give the approximate description of the positions of a path. Hence even our corridor-based concept of word-sequences (cf. 4.6), when applied to our path w, would for each word of a describing word sequence only detect the first finitely many loops of w. Consequently, the resulting word sequence would indicate nullhomotopy in each of its words, even if w is not nullhomotopic. Thus the obstruction for w being nullhomotopic would not be in the word sequence, and this would be a severe obstruction for our concept of proving Theorem A. Fortunately, this situation does not arise in the plane. The moral reason for that (even if at no state of the paper we are forced to prove this as a lemma) is given by the fact that one can always contract a nullhomotopic path inside the region enclosed by its trace. Hence tiny paths only need tiny nullhomotopies, and hence, if infinitely many tiny nullhomotopic paths are composed, the nullhomotopies can also be plugged together in order to give a nullhomotopy for the composite path, also. Observe that our counter example of Fig.c is based on the fact that even though the loops α_i get arbitrarily small, the contracting disks do not.

4.8 Remark: Example 4.7 shows that an attempt to extend the concept of word sequences to dimension three would imply that whatever combinatorial description is used in dimension 3, it would not only have to take care of the position of the paths, but also of the position of the contracting disks. This is not necessary in dimension two. Roughly speaking, the cancellation facilities for word sequences describe how a nullhomotopic path can be contracted to one point. Fig.d is devoted to illustrate this fact. Observe that a path like w may be contracted in various ways if contained in a unpunctured plane, but the hole H restricts these possibilities to essentially one. The information how w can be contracted is then contained in the facility how the describing word can be cancelled, and in Fig.d this facility is also graphically illustrated by by the arch-system.

The purpose of the following remark is *pointing out that certain aspects of the reals are fairly similar to phenomena that we experience when working with word sequences:*

4.9 Remark: Recall (4.5/4.6) that word sequences will be introduced in this paper as a combinatorial description of the variety of paths in a space in Normal Form. The basic structure of a word sequence is that each of its words is finite, and that the first finitely many entries of a word sequence give an approximate description of each path. The more words of such a finite initial path one considers, the more precise the description becomes. When working with word sequences we will need the property that a sequence of paths that is uniformly converging can be described by a sequence of word sequence which is converging by stabilization on each level index. For that purpose we need to define the variety of paths to which certain entries in the in the initial finite part of

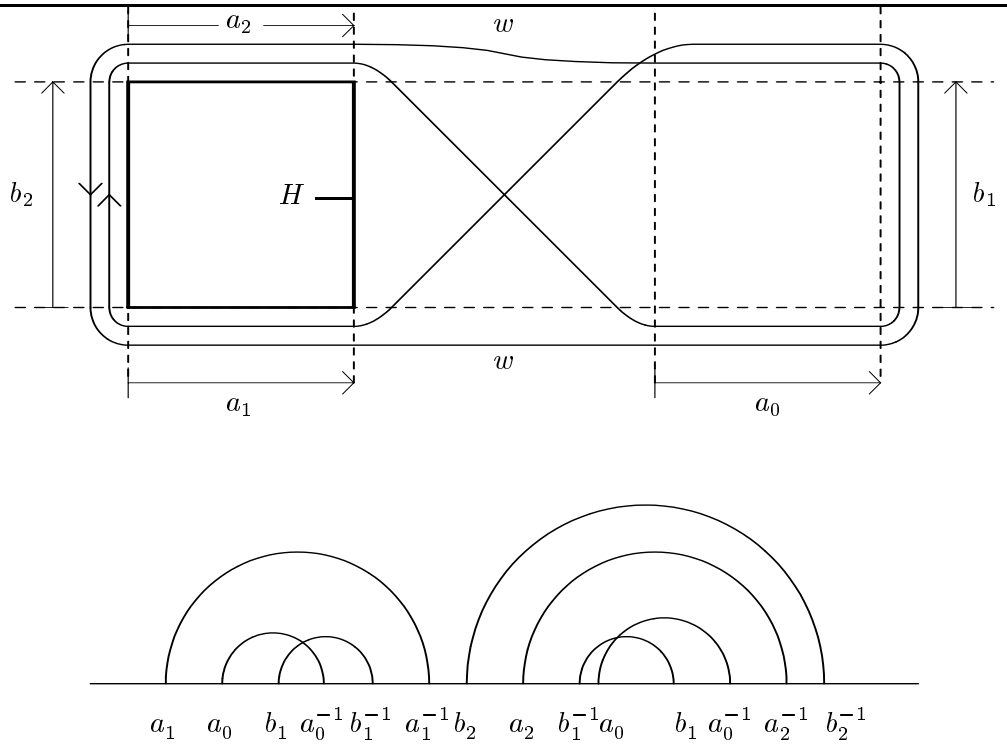


Fig.d (belonging to 4.8):
This figure shows a nullhomotopic path w in some corridor lattice, the word that can be read from w and by an arch-system visualizes the cancellation facilities of that word. Observe that these cancellations facilities mirror the way that one has to take if one wants to contract w.

word sequences correspond to be a closed subset in the variety of paths. Since the entire variety of paths must be considered as an essentially continuous domain and has to be covered by finitely many closed sets which belong to the possibilities for initial parts of word sequences, these domains have to overlap. These overlaps correspond to paths which can in a legitimate way be described by word sequences where even their initial part has different entries. The situation is indeed similar to the the concept of describing reals by digit sequences in the decimal system: The first finitely many entries (with respect to the decimal point) describe an interval of the reals where the number could lie, and an equation like "$1 = 0.999\ldots$" reminds us that there are also overlaps which correspond to different legitimate combinatorial descriptions of the same number. For word sequences the origin of this non-uniqueness can be almost as easily understood as the nine-case can be understood for the reals (cf. B.1.4 and C.1.3). As for the reals, it would not be hard for word sequences to set up the definition in such a way that in each of the non-uniqueness-cases just one description is declared to be a legitimate word sequence. However, the price for forcing uniqueness that way would be that in a lot of internal constructions we would be in permanent danger of running into illegitimate word sequences, and for that reason we decided not to exclude the non-uniqueness-case in the definition of word sequences in this paper. The situation is analogous to the reals:

If one tries to forbid the digit-sequence "0.999..." and regards "1.000..." as the only legitimate way for describing the number one, one starts missing "0.999..." if one seeks a combinatorial way for describing what happens if one multiplies 0.333... by three.

4.10 Remark: The Hawaiian Earrings (cf. 4.4) are also good for demonstrating another property that for pathological spaces one always has to be aware of. This is the phenomenon of of non-rectifiable paths. It already occurs for the Hawaiian Earrings. For this space there does exist a continuous path which runs around each of the loops exactly once, and, when using the standard metric of the Hawaiian Earrings, the total length is $\pi \cdot \sum_{i=1}^{\infty}(\frac{1}{i}) = \infty$. Hence this path is not rectifiable, and so will be any path that belongs to the same homotopy class. In particular this means that the arc-length cannot be used to identify the shortest path in this homotopy class. This phenomenon does not make our task of of choosing a homotopy combing easier, because finding a homotopy combing will essentially imply having to select a system of paths, one representative for each homotopy class. And, as just seen, this cannot be done by using the natural concept of arc-length. However, we will overcome the technical obstruction with the help of our concept of word sequences, since word sequences will allow to choose "algebraically shortest paths" instead of the geometrically ones. We take the opportunity to point out that this obstruction cannot be overcome by a change of the metric. Even if we should decide not to use the standard metric for the Hawaiian Earrings Z_0, (i.e. that $\frac{1}{i}$ is the radius of the ith loop), but if we metricize Z_0 in such a way that $\frac{1}{i^2}$ is the radius of the ith loop, the path running around each of the loops once will then rectifiable, but the path running around the ith loop i times will not. That way homotopy classes that only contain non-rectifiable paths will be found for every possible metric of the Hawaiian Earrings. Observe that this observation is also relevant for spaces in Normal Form, since at every accumulation point of holes (cf. 3.2(ii)) Hawaiian Earrings are embedded (at least, up to homotopy equivalence) into a space in Normal Form.

4.11 Example: Consider a path $c_1 \cdot c_2 \cdot c_3$ (cf. 5.1) which is composed out of a tame path c_1, a path c_2 running around each of the loops of some Hawaiian-Earring subspace exactly once as in 4.10, and finally a tame path c_3 again. Thanks to the c_2-component, the entire path is not rectifiable. This makes it impossible to parametrize this path by a constant finite velocity. For similar reasons, on the other hand the velocity cannot be infinite everywhere since most parts of the path are indeed tame. This tells us that any attempt to find a valid parametrization of our path has in principle to be given by a non-constant velocity, because the velocity has to take on a pole when our path is passing the point of non-rectifiability at the end of c_2.
Recall that in 0.1/1.3 we indicated that we want to prove Theorem A by constructing a system of paths along which a continuous image of a sphere is contracted by some combinatorial method. This combinatorial method will not provide us the paths with an explicit parametrization, but such a parametrization is necessary to define a contraction along this system of paths. Hence constructing this parametrization will be one of the tasks that are to be solved in this paper. And the above example shows that the most natural idea for getting such parametrizations, namely that of a constant velocity, will not work. We will call the need to let the velocity tend to infinity in the neighbourhoods

of the non-rectifiable segments the *"speed-up phenomenon"*. Besides this, we see that we will be only able to parametrize our paths by some method that will individually respond to the concrete behavior of each of the paths.

4.12 Example: Consider $\{(x,y) \mid y < \sin(\frac{1}{x}) \text{ for } x \neq 0 \ , \ y < -1 \text{ for } x = 0\}$ as a hole in the one-point compactified Euclidean plane which gives as its boundary $\{(x,y) \mid y = \sin(\frac{1}{x}) \ , \ x \neq 0\} \cup \{\infty\} \cup \{(0,y) \mid -1 \leq y \leq +1\}$. The remarkable aspect of this example is that here we have a hole which is a topological disk and whose boundary cannot be parametrized by a continuous curve. Recall that in 3.6/2.7(i) we could only reduce the proof of Theorem A to spaces in Normal Form by quoting [Pomk].(Thm.2.1) and that [Pomk].(Thm.2.1) was only applicable since a continuous image of the sphere must be locally pathwise connected. The purpose of this example has been to recall how essential for the entire concept of proof our ability to reduce the problem to the locally pathwise connected case is.

Note that by reducing the proof of our Theorem A to spaces in Normal Form in 3.6, we essentially retracted the holes of our space into nice shape. One could have also tried the opposite strategy, i.e. retracting our space to a graph-like space that does not have inner points while hoping that in such a space natural combing paths can be determined more easily (cf. [Zas3].3.20). However, the following example shows that such retractions cannot be done in a homotopy equivalent way:

4.13 Example: Consider Fig.e. It pictures a space Z_2 which is essentially given as a subset of the plane by the points that are painted black in Fig.e. More precisely, Z_2 consists of a solidly filled square in the middle, and four peripheral isosceles right-angled triangles that are only filled by a graph-like pattern of lines. This pattern of lines is constructed by a infinite iterative process, and, of course, only the first finitely many iteration steps are painted in Fig.e.

Now assume, that we select two points at opposite boundaries of the inner square and choose inside the neighbourhood of each of these points Hawaiian Earrings (i.e. a subspace of the corresponding outer triangle which is homotopy equivalent to Z_0 (cf. 4.5(Fig.b)). If we try to get rid of the inner points of Z_2 by collapsing the inner square to a single point, it would be possible to define a continuous path which runs over all of these loops of the Hawaiian Earrings alternately. Such a path could not have been constructed before the collapse of that inner square, and hence this collapse is not a homotopy equivalence.

§5. Combings and Homotopy Combings

5.1 Convention of our Notation:

In this paper we do use the words "curve" and "path" synonymously and usually think of them as objects which are only defined up to monotonous reparametrization. Nevertheless we do usually not distinguish between a curve and its parametrization mapping, i.e. if we denote a curve by a variable, this variable is also used for the parametrization mapping. If the operations "$()^{-1}$" and "$\circ$" are applied to such variables, then they are

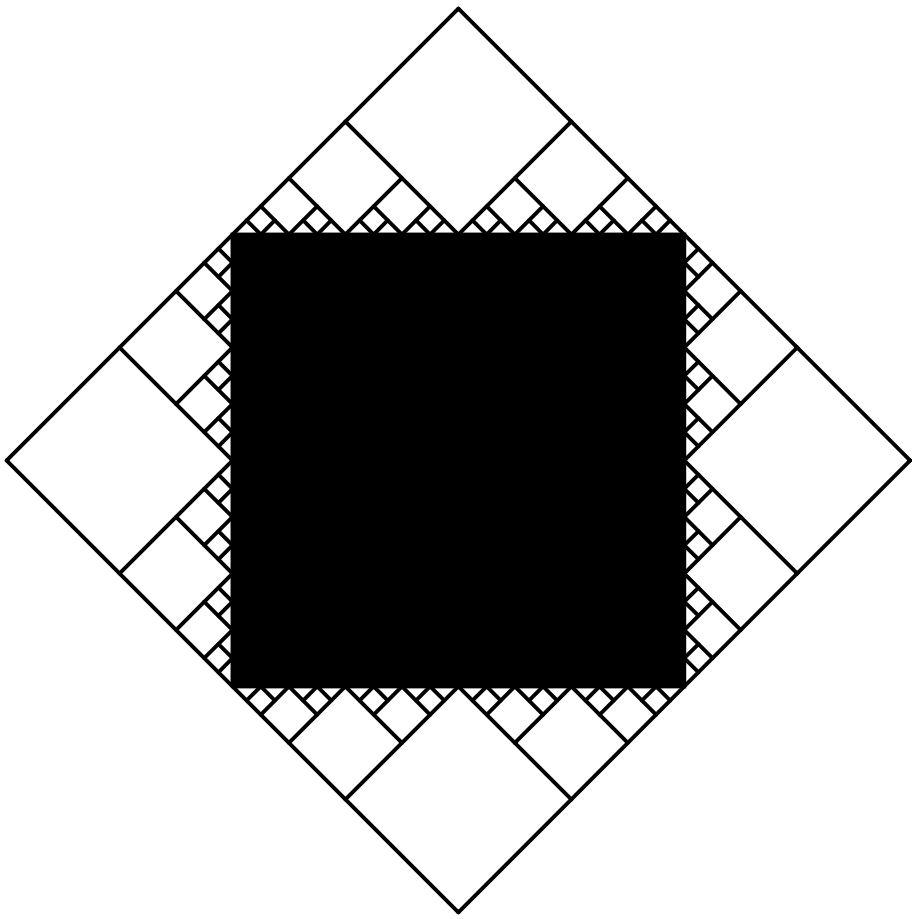


Fig. e (belonging to 4.13):
This is a picture of a compact space of the plane which apparently cannot be retracted to a space with empty interior.

to be understood as referring to the parametrization mapping, whereas "$\cdot$" indicates that curves have to be plugged together. Reversing the orientation of a curve will be denoted by putting a bar over the curve's name.
However, there will be parts in this paper where we need to treat curves in such a way that a mere change of the parametrization would make them a different object. In those situations we will call them *"parametrized curves"*. Conversely we will use the attribute "abstract" to point out that we are back in the standard situation: For an *abstract path* a parametrization is either not given, or it is arbitrarily chosen out of many ones that describe this object.

5.2 Definition: Let X be a pathwise connected metric space. Since in this paper we usually just need to consider spaces in Normal Form, we denote the metric of X as the

absolute value of the difference of points. Then

(i) $\widehat{X_x} := \{(P,\alpha) \mid P \in X\,,\ \alpha$ is a relative homotopy classes of paths that connects x to $P\}$,

(ii) $\mathcal{L}(X)$ is the set of all abstract paths inside X; the subset of those paths that start at the base point x will be denoted by $\mathcal{L}_x(X)$.

(iii) $\widetilde{\mathcal{L}}(X), \widetilde{\mathcal{L}_x}(X)$ are the spaces of all parametrized paths with the same conditions on their start and end points as in (ii).

$\widetilde{\mathcal{L}}(X), \widetilde{\mathcal{L}_x}(X)$ will always be regarded as metric spaces, where the metric is essentially given by the ordinary supremum metric. More precisely:

$$\begin{aligned} d(c_1,c_2) = \sup(\{|c_2(t)-c_1(t)| \mid 0 \le t \le \min(b_{c_1},b_{c_2})\} \cup \\ \cup\{|c_2(b_{c_2})-c_1(t)| \mid c_1 \le t \le c_2\} \cup \\ \cup\{|c_2(t)-c_1(b_{c_1})| \mid c_2 \le t \le c_1\}). \end{aligned} \tag{1}$$

On the other hand, we will in this paper refrain from introducing a metric on $\mathcal{L}(X)$ and on $\mathcal{L}_x(X)$ and just treat $\mathcal{L}(X), \mathcal{L}_x(X)$ as sets (cf. 5.4). Finally $\widehat{X_x}$ will also be metricized, here the *"homotopic distance"* as defined in the following will give a metric (cf. 5.3).

$$\begin{aligned} &d((P_1,\alpha_1),(P_2,\alpha_2))) := \inf\{\mathrm{diam}(U) \mid U \subset X\,,\ U \text{ contains a path } u:[0,1] \to \\ &X \text{ with } u(0)=P_1 \text{ and } u(1)=P_2 \text{ such that } u_1 \cdot u \cdot \overline{u_2} \simeq 0 \text{ in } X, \text{ where } u_1 \text{ and} \\ &u_2 \text{ are any two paths representing the relative homotopy classes } \alpha_1 \text{ and } \alpha_2, \\ &\text{respectively}\} \end{aligned} \tag{2}$$

5.3 Remark: It is immediate that 5.2(2) defines a pre-metric; however, it is not so obvious that it actually defines a metric. For that purpose one has to prove that it cannot happen that for $P_1 = P_2$ and for $\alpha_1 \neq \alpha_2$ the nontrivial homotopy class $\alpha_1^{-1} \cdot \alpha_2$ can be represented by a sequence of paths that fit into neighbourhoods whose diameters tend to zero. Instead of giving a lengthy elementary proof for this fact now, we postpone it till it can be reduced as a mere corollary from our theory of word sequences (cf. D.12.5).

5.4 Remark: Of course, there exist sensible ways of defining a metric on our space $\mathcal{L}_x(X)$. However, the main technical bulk of this paper will have to be spend on defining a continuous section $\widetilde{C_X} : \widehat{X_x} \to \widetilde{\mathcal{L}_x}(X)$ that reverses the natural projection. This $\widetilde{C_X}$ will be found from lifting some $C_X : \widehat{X_x} \to \mathcal{L}_x(X)$ that will have been previously constructed. $\widetilde{C_X}$ will be continuous if we manage to find for two homotopically near pairs (P_j,α_j) such parametrized paths $c_j = \widetilde{C_X}(P_j,\alpha_j) \quad (j \in \{1,2\})$ that are close to each other with respect to the supremum metric. I.e. the big technical difficulty will rest in finding a definition for $\widetilde{C_X}$ such that the distances between the points $c_1(t)$ and $c_2(t)$ get small. Now, since $\mathcal{L}_x(X)$ consists of abstract paths only, any sensible metric on $\mathcal{L}_x(X)$ can only rest on measuring the distances between the point $c_1(t)$ and some points $c_2(s)$ where the parameter values s might differ from t in a way that is technically difficult to control. Given the task of getting distances of type $|c_1(t)-c_2(t)|$ small, it is of little help to know that distances of type $|c_1(t)-c_2(s)|$ are small. Hence we decided not to introduce any metric on $\mathcal{L}_x(X)$ and not to discuss whether C_X will or will not

be continuous. Thus all mappings that are defined on or go into $\mathcal{L}_x(X)$ are only to be understood as set-theoretic maps.

5.5 Definition: Let X be a pathwise connected space and choose $x_0 \in X$ as a base point.

(i) A system of paths

$$C_X = \{c_x : [0, b_{c_x}] \to X \mid x \in X\}$$

is called a *"combing of X"*, provided $c_x(0) = x_0$, $c_x(b_{c_x}) = x$ for all $x \in X$. This definition is an adaptation of a concept that has become quite standard in combinatorial group theory (cf. [ECHLPT].(Def.3.6.1)). A combing can be defined as an *"abstract"* or as a *"parametrized combing"* in the sense these attributes have been introduced in 5.1. According to which of these attributes is chosen, a combing C_X also defines a set-theoretic mapping $X \to \mathcal{L}_{x_0}(X)$ or $X \to \widetilde{\mathcal{L}_{x_0}}(X)$. However, in general we cannot expect this mapping to be continuous.

(ii) A system of paths

$C_X = \{c_{x,\alpha} : [0, b_{c_{x,\alpha}}] \to X \mid x \in X$, α is a relative homotopy class of paths connecting x_0 to $x\}$

is called a *"homotopy combing of X"*, provided $c_{x,\alpha}(0) = 0$, $c_{x,\alpha}(b_{c_{x,\alpha}}) = x$ and $[c_{x,\alpha}] = \alpha$ for all $x \in X$. Homotopy combings can be defined abstract or parametrized, and accordingly they can also be interpreted as mappings $\widehat{X_{x_0}} \to \mathcal{L}_{x_0}(X)$ or $\widehat{X_{x_0}} \to \widetilde{\mathcal{L}_{x_0}}(X)$. Observe that the use of the mere phrase "homotopy combing" does not imply any continuity conditions.

(iii) A parametrized homotopy combing that defines a continuous mapping $\widetilde{C_X} : \widehat{X_x} \to \widetilde{\mathcal{L}_x}(X)$ will be called a *"continuous homotopy combing"*.

5.6 Combing Theorem: *Let X be a space in Normal Form. Then*

(i) X admits a combing.

(ii) X admits a homotopy combing C_X.

(iii) X admits a continuous homotopy combing, $\widetilde{C_X}$ such that $c \mapsto b_c$ is continuous on $Im(\widetilde{C_X})$ (cf. 6.2).

Remark: (i) and (ii) are trivial by the axiom of choice since spaces in Normal Form are pathwise connected. However, proving (iii) will be the major part of proving our Theorem A and will not be complete until E.5.11. Parts B–E will be essentially devoted to proving (iii), so that this theorem must be understood as the main contribution of the proof of our Theorem A. Our $\widetilde{C_X}$ will arise as a lift of our abstract homotopy combing C_X whose construction will already be completed at the end of Part D (E.1.3/B.3.1). I.e. this C_X cannot be obtained by the axiom of choice, but it will also be constructed with great care, so that it will admit the continuous lifting that proves (iii).

§6. Proof of the Main Theorem modolo the Combing Theorem

Before we can actually give this proof in 6.5, we need some preparatory lemmata:

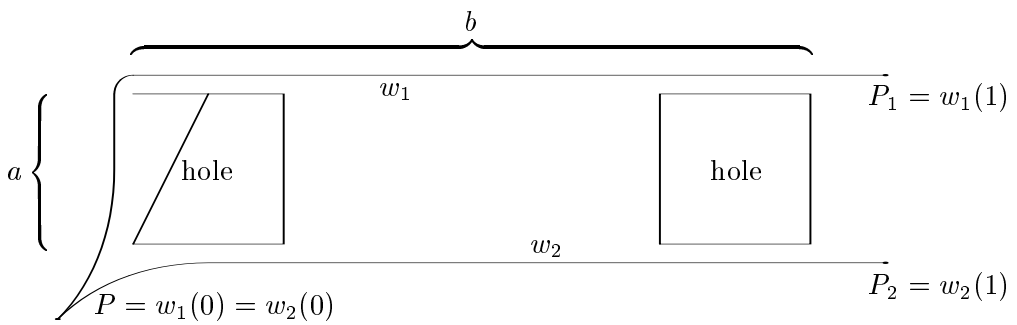


Fig. f (belonging to 6.1):
This figure shows an arrangement where two paths w_1 and w_2 have huge homotopic distance ($\approx b$), but comparatively small ($\approx a$) distance in the supremum metric.

6.1 Lemma: *Consider the natural projections*

$$\widetilde{\mathcal{L}_x}(X) \xrightarrow{p_1} \mathcal{L}_x(X) \xrightarrow{p_2} \widehat{X_x} \xrightarrow{p_3} X,$$

where p_3 is the projection $(P, \alpha) \mapsto P$. Then $p_3 \circ p_2 \circ p_1$ and p_3 are continuous mappings (cf. 5.4), but $p_2 \circ p_1$ is not.

Proof: The continuity claims are immediate since $|P_1 - P_2|$ is a lower bound for the homotopic distance $d((P_1, \alpha_1), d(P_2, \alpha_2))$ for any α_1 and α_2 and for the supremum metric of any two paths leading to P_1 and P_2. The origin of the claimed non-continuity of $p_3 \circ p_2$ is illustrated in Fig.f: For paths as pictured in Fig.f the homotopic distance is approximately b, while the distance of paths according to the supremum metric is essentially a. An arrangement as in Fig.f shows that b can sill be fairly big while a tends to zero. □

6.2 Remark: *Observe that the function $\widetilde{\mathcal{L}_x}(X) \to \mathbb{R}_+ \cup \{0\}$, $c \mapsto b_c$ is not continuous.* In order to see this consider two paths c_0 and c_1 where c_0 is an arbitrary path and c_1 is a path that is defined on a fairly long parameter interval but remains entirely in the neighbourhood of the endpoint of c_0. Further, think of $c_0 \cdot c_1$ as being parametrized by a mere shift of values without rescaling (i.e. $c_0 \cdot c_1 : [0, b_{c_0} + b_{c_1}] \to X$ such that $c_0 = (c_0 \cdot c_1)\big|_{[0, b_{c_0}]}$. If for such a construction c_1 is chosen accordingly, it is not hard to see that $d(c_0, c_0 \cdot c_1)$ can be arbitrarily small while $b_{c_1} = (b_{c_0} + b_{c_1}) - b_{c_0}$ can be made arbitrarily large.

6.3 Lemma: *Let K be a compact space and let $f : K \to \widetilde{\mathcal{L}_x}(X)$ be a continuous map. Further, let $b : \widetilde{\mathcal{L}_x}(X) \to \mathbb{R}_+ \cup \{0\}$, $c \mapsto b_c$. If $b \circ f$ is continuous, then f is nullhomotopic.*

Proof: Standard compactness arguments and the assumption of this lemma give that the parameter intervals in $f(K)$ are bounded by a bound b_0. Hence, after extending the paths by a constant value as in 5.2(1), we may assume that all paths in $\mathrm{Im}(f)$ are defined on the same parameter domain $[0, b_0]$. Now the contracting homotopy of f is

given by $H(z,s) = (f(z))\big|_{[0,s]} \cdot ((f(z))(s))\big|_{[s,b_0]}$, where $(f(z))(s)$ is to be understood as a notation for the constant path that takes on the value $(f(z))$ only. □

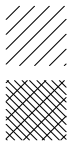
6.4 Lemma: *Let $k \geq 2$ and let $f : S^k \to X$ be continuous. Then f can be lifted to a continuous function $\widehat{f} : S^k \to \widehat{X_x}$* (cf. 6.1).

Proof: In order to prove that lemma we need to define the α-component for a given P-component and for a preimage $z \in S^k$ with $f(z) = P$. This is done as follows: Without loss of generality let us assume that a base point $z_0 \in S_k$ is chosen such that $f(z_0) = x_0$. For an arbitrary preimage z we connect z_0 to z by a path w in S^k and then our lift $\widehat{f}(z)$ can be defined as $(f(z), [f \circ w])$. This setting is well-defined since due to the trivial fundamental group of S^k all possible paths w are relatively homotopic to each other, and so are their image paths $f \circ w$. In addition, the lift turns out to be continuous, since in the standard set-up of a continuity proof (i.e. $z' \in U(z,\delta)$, $f(z') \in U(f(z),\varepsilon)$ with $f(U(\delta,z)) \subset U(\varepsilon, f(z))$) a path can be found which is an image under f and and connects $f(z)$ to $f(z')$ inside $U(f(z),\varepsilon)$; and the existence of such a path proves that the homotopic distance of $\widehat{f}(z)$ and $\widehat{f}(z')$ does not exceed ε. □

6.5 Proof of the Main Theorem modolo the Combing Theorem:
By virtue of 3.6, this needs only to be proved for a space in Normal Form. Such a space has by 5.6(iii) a continuous homotopy combing such that $c \mapsto b_c$ is continuous in $\mathrm{Im}(\widetilde{C_X})$. Now, given a continuous map $f : S^k \to X$, it can by 6.4 be lifted to $\widehat{f} : S^k \to \widehat{X_x}$ and by 5.6(iii) be extended to $\widetilde{C_X} \circ \widehat{f} : S^k \to \widetilde{\mathcal{L}_x}(X)$. Lemma 6.3 gives that the latter map is nullhomotopic and by 6.1 this nullhomotopy can be projected down to X where it becomes an ordinary nullhomotopy for X. Hence f cannot have represented a non-trivial element in $\pi_k(X)$. □

Abschnitt B: Construction of a homotopy combing via word sequences

By A.6.5 we are left with the task of proving our combing theorem A.5.6. As already indicated in 0.1/A.1.3 and and in A.4.6/A.4.10 our main tool for constructing the desired continuous homotopy combing will be a translation of this geometric problem into the combinatorial problem for word sequences. In order to make the structure of the proof more transparent, the purpose of this part of the paper is only giving the formal definition of word sequences and of their relation to paths and of proving easy facts, while the more difficult constructions or proofs will be postponed to Parts C and D. Based



Area belonging some corridor

Area belonging to some hole

Boundary line of a hole of the ith corridor lattice

Boundary line of a corridor of the ith lattice

Boundary line of a hole of the $(i+1)^{\text{st}}$ lattice

Boundary line of a corridor of the $(i+1)^{\text{st}}$ lattice

Boundary line of a hole of the $(i+2)^{\text{nd}}$ lattice

Boundary line of a corridor of the $(i+2)^{\text{nd}}$ lattice

Fig. g0 (belonging to 1.1):
The four main pictures belonging to the series of Fig.g-drawings are devoted to showing adjacent corridor lattices of some part of our space X according to the same scale.

on the results and statements of this part we will be able to explain how to construct a homotopy combing that satisfies A.5.6(iii) for spaces in Normal Form.

§1. The (multivalued) Mappings between Paths and Word Sequences

1.1 Definition of the corridor network $\mathcal{N}(X)$ of a space X in Normal Form:
In this definition we associate with each space X in Normal Form a set $\mathcal{N}(X)$ of areas of $[0,1]^2$ *("corridor network")* each of which will be called a "corridor". We let $\mathcal{N}(X) := \bigcup_{i=0}^{\infty} \mathcal{N}_i(X)$ where this union is disjoint and the index i is associated with the level index of holes (cf. A.3.1) and gives a graduation on the corridor network. $\mathcal{N}_i(X)$ is called the "i^{th} corridor lattice of X". Each $\mathcal{N}_i(X)$ is a disjoint union $\mathcal{N}_i^{\text{h}}(X) \cup \mathcal{N}_i^{\text{v}}(X)$, where the symbols $()^{\text{h}}$ and $()^{\text{v}}$ stand for "vertical" and "horizontal". Now $\mathcal{N}_i^{\text{h}}(X)$ consists of the closures of the connected components of $X_i \cap \{(x,y) \mid 0 \le x,y \le 1 \ , \ \exists m \in \mathbb{N}$ with $\frac{2\cdot m-1}{3^i} < y < \frac{2\cdot m}{3^i}\}$, and similar we let $\mathcal{N}_i^{\text{v}}(X)$ be the connected components of $X_i \cap \{(x,y) \mid 0 \le x,y \le 1 \ , \ \exists k \in \mathbb{N}$ with $\frac{2\cdot k-1}{3^i} < x < \frac{2\cdot k}{3^i}\}$. Note that we always have that $\mathcal{N}_0(X) = \emptyset$.

1.2 Terminology related to and easy properties of a corridor network:
(i) The elements of {vertical, horizontal} are called *"basic directions"*. Since there are only these two of them, we may use phrases such as "the corridor C' which belongs to the other basic direction than C does." In order to address the basic direction of

some corridor in formulae, we hereby introduce an operator: $BD : \mathcal{N}(X) \to \{\text{vertical, horizontal}\}$ that in 1.3 will be extended to $\mathcal{A}(X)$, also.

(ii) The areas $[\frac{2\cdot k-1}{3^i}, \frac{2\cdot k}{3^i}] \times [0,1]$ and $[0,1] \times [\frac{2\cdot m-1}{3^i}, \frac{2\cdot m}{3^i}]$ are regarded as *"vertical"* and *"horizontal lines of corridors* on the ith level". Note that any corridor on that level is contained in one of those areas. The remaining areas (i.e. $[\frac{2\cdot k}{3^i}, \frac{2\cdot k+1}{3^i}] \times [0,1]$ and $[0,1] \times [\frac{2\cdot m}{3^i}, \frac{2\cdot m+1}{3^i}]$ are then called *"gap-lines"*.

(iii) Recall that by definition two opposite of the four sides of the rectangular region that is considered as a corridor lie by construction on boundary points of X. These two boundary lines will be called the *"end segments* of the corridor", while the term *"sides of the corridor"* will be exclusively reserved for the two other boundary lines. Observe that a vertical corridor has vertical sides and horizontal end segments, and vice versa for horizontal corridors. In addition, only these sides are usually referred to if for some corridor the expression "boundary (lines)" is used.

(iv) Observe that the grid $(\{0, \frac{1}{3^i}, \frac{2}{3^i}, \ldots, \frac{3^i-1}{3^i}, 1\} \times [0,1]) \cup ([0,1] \times \{0, \frac{1}{3^i}, \frac{2}{3^i}, \ldots, \frac{3^i-1}{3^i}, 1\}$ subdivides $[0,1]^2$ into square shaped regions such that each of these line segments is either given by the boundary lines of holes or of corridors of the level index i. To each of these square shaped regions we attach precisely one of the following five attributes according to the following rules:

- ▷ *"hole"*, if this region does not belong to X_i.
- ▷ *"vertical"* or *"horizontal corridor segment"* if this region is part of a corridor of the corresponding basic direction, but is not part of a corridor of the other basic direction.
- ▷ *"crossing region of corridors"*, if this region belongs to a vertical and to a horizontal corridor.
- ▷ *"pure lattice square"*, if the region is in neither of the above categories.

The phrase *"extended lattice square"* is used for an area which consists of a pure lattice square and of the eight adjacent or diagonally adjacent squares in the grid of a fixed level index. Observe that by construction (cf. A.3.12) we know crossing regions of corridors to belong to X (i.e. these regions cannot contain holes on finer level indices) and conversely all holes do definitely not belong to X.

(v) If $C_1 \neq C_2 \in \mathcal{N}_i(X)$ and belong to the same basic direction, then they must be disjoint; if $C_1 \in \mathcal{N}_i(X)$, $C_2 \in \mathcal{N}_j(X)$ and $i < j$ then C_2 may also be a proper subset of C_1. If $C_1 \in \mathcal{N}_i^{\text{h}}(X)$, $C_2 \in \mathcal{N}_i^{\text{v}}(X)$, these corridors may either be disjoint, or they intersect in a crossing region of corridors, or their intersection consists precisely of one point, which is also the corner point of some hole (*"corner point intersection"*).

(vi) If C_1 and C_2 belong to the same basic direction and $C_1 \subset C_2$, then C_1 will be called a *"split-up corridor of* C_2*"*, and conversely C_2 a *"root-corridor of* C_1*"*. By construction (cf. 1.1) split-up corridors exist for any higher level to each corridor and are never unique, while root-corridors do not always exist, but if they do, they are unique (cf. (v)). Hence we will also use the expression "root-corridor" with the definite article, in particular, if the level on which we are looking for the root is given, or if we are referring to the root on the preceding level.

1.3 Definition of the alphabet that is associated with a space X in Normal Form:
Let $\mathcal{A}(X), \mathcal{A}^{-1}(X)$ be disjoint sets each of which is bijectively associated to $\mathcal{N}(X)$, and let $\overline{\mathcal{A}}(X) = \mathcal{A}(X) \cup \mathcal{A}^{-1}(X)$. While $\mathcal{N}(X)$ is defined as a set of subsets of $\mathbb{R}^2$, $\mathcal{A}(X)$ is interpreted as a set of symbols that will be used to compose combinatorial objects, namely words and word sequences (cf. A.4.6). In particular $\mathcal{A}^{-1}(X)$ will be understood as the set of "inverted symbols" with respect to $\mathcal{A}(X)$ (i.e. $l \in \mathcal{A}(X) \iff l^{-1} \in \mathcal{A}^{-1}(X)$). There will be only a few selected lemmata where it is appropriate to distinguish the geometric objects of $\mathcal{N}(X)$ and the algebraic of $\mathcal{A}(X)$, in the majority of all cases we may identify $\mathcal{A}(X)$ and $\mathcal{N}(X)$, and then in particular we will use the same variable to the denote an element of $\in \mathcal{N}(X)$ and its associated symbol. Since $\mathcal{A}(X)$ and $\mathcal{N}(X)$ are bijectively associated, the superscripts $()^{\mathrm{h}}$ and $()^{\mathrm{v}}$ and level indices are also defined for the symbols $\mathcal{A}(X), \overline{\mathcal{A}}(X)$ and $\mathcal{A}^{-1}(X)$. The same holds for all attributes that are in 1.2 defined for corridors.

1.4 Definition of word sequences and readings:
Fix a space X which has Normal Form and use $\mathcal{N}, \mathcal{N}_i, \mathcal{A}, \overline{\mathcal{A}}, \ldots$ as abbrevations for $\mathcal{N}(X), \mathcal{N}_i(X), \mathcal{A}(X), \overline{\mathcal{A}}(X), \ldots$. Let $\overline{\mathcal{A}_i^*}$ be the set of all finite words whose letters are taken from $\overline{\mathcal{A}_i}$. Then an element from $(\overline{\mathcal{A}_i^*})_{i\in\mathbb{N}_0}$ is called a *"sequence of words"*. Such a sequence of words is called *"word sequence"*, if it is a *"reading of u"*, i.e. if it is related to a path $u : [a, b] \to X$ according to the following rules for how to choose certain intervals in $[a, b]$ and to associate them with letters of $\overline{\mathcal{A}}(X)$:

(i) For all $i \in \mathbb{N}$ we mark all intervals $[a', b'] \subset [a, b]$ such that $u([a', b'])$ is contained in some corridor $C \in \mathcal{N}_i$ and that either $a' = a$ or $u(a')$ lies on a boundary line of C, and similarly either $b' = b$ or $u(b')$ lies on the boundary line of C'. All points in the interior of a corridor must be contained in one of those marked intervals, points in the interior of crossing regions of corridors even in two intervals so that each of the two crossing corridors is associated to some interval. One-point intervals $[a', a'] = \{a'\}$ are in general not excluded from being marked.

(ii) If a path has general position with respect to the grid-lines of $\mathcal{N}_i$ then the intervals are uniquely defined by the above demands. If not, there is some freedom of choice which is in principle described here in (ii), but is then implicitly restricted by the following demands (iii)–(xiii):

If a path spends a segment on a boundary line of a corridor, but does not go into the interior, then this segment may or may not be enregistered in a marked interval. In particular, if a path spends a segment on the boundary line of a corridor before going into the interior, then the start point for the corresponding interval may be chosen anywhere on the boundary line segment. If $u_{|(t-\varepsilon,t+\varepsilon)} \subset C$ and if $u(t)$ is a point on the boundary line of C, then either $(t-\varepsilon, t+\varepsilon)$ is some subinterval of a marked interval, or $(t-\varepsilon, t]$ is the end-segment of one marked interval and $[t, t+\varepsilon)$ the start segment of another *("tangent point case")*. If there are more points than t inside $(t-\varepsilon, t+\varepsilon)$ where $u(t)$ is taken on the boundary line of t, then marked intervals may be adjusted accordingly even in more complicated scheme.

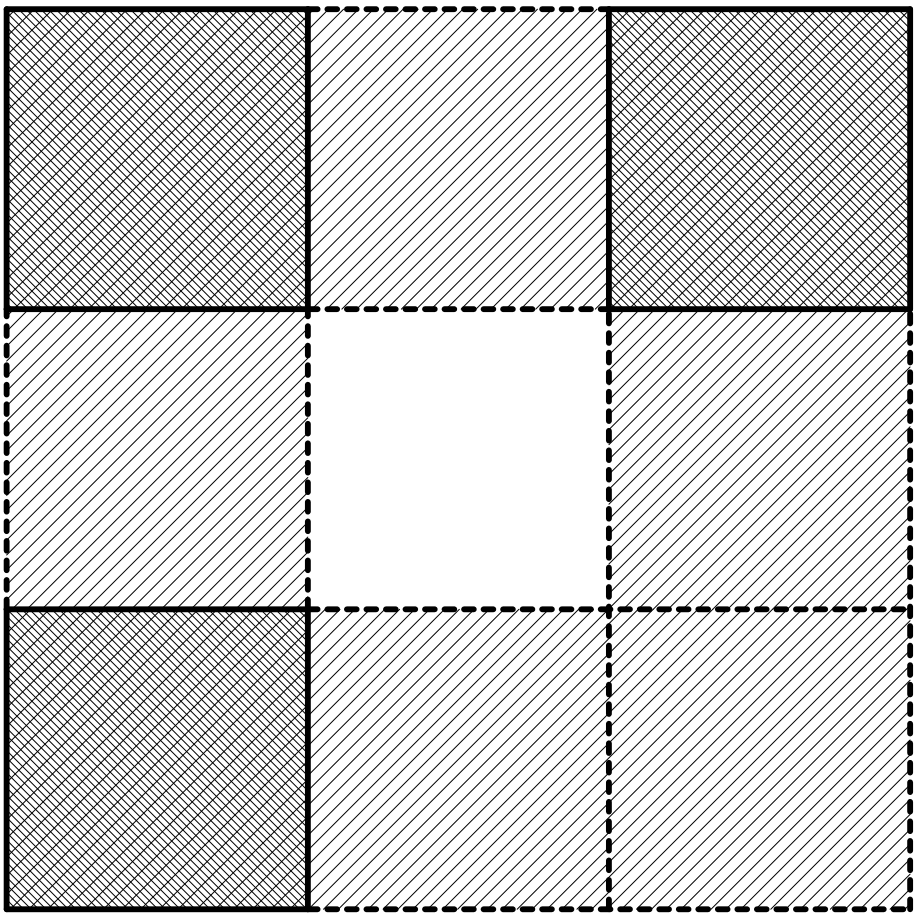


Fig. g_1 (belonging to 1.1):
This figure pictures the corridors of a fixed (say i^{th}) lattice within this part and the holes as far as they have been integrated into that lattice (cf. the final remark of A.3.9).

(iii) The overlap of marked intervals that are associated with the same level-index i is restricted to three cases, namely to the one-point overlap at a tangent point as described in (ii), or to paths which pass through crossing regions of corridors or through corner points of holes (i.e. through those places where different corridors do overlap).

(iv) It now must be decided which of the marked intervals $[a', b']$ are "essential" and which are inessential. The following rules must be obeyed:

(1.) If neither $a' = a$ nor $b' = b$, and if $u(a')$ and $u(b')$ lie on different sides of C, then $[a', b']$ must be treated as essential.

(2.) If neither $a' = a$ nor $b' = b$ and if $u(a')$ and $u(b')$ lie on the same boundary side of C', then $[a', b']$ must not be treated as essential.

(3.) If $a' = a$ or $b' = b$, then $[a', b']$ may or may not be treated as essential.

In the following the "essential intervals" are regarded as *"reading domains"*. The subsequent items (v)–(xiii) only deal with reading domains.

(v) A reading domain is usually interpreted as a subsegment of a path that *"crosses a corridor"* (cf. (1.)). Rule (2.) says that if a path enters a corridor but does not cross is, this is not to be acknowledged by a reading domain. Rule (3.) allows, if a path ends or begins inside a corridor C, to treat the path as if it would or would not cross C. I.e. "incomplete acts of crossing" at the beginning or at the end of a path may be treated as if they were just enterings in the sense of (2.) or completed crossings in the sense of (1.).

(vi) Let $a' \leq b' < c' \leq d'$ be parameter values which on a fixed level i come from adjacent reading domains $[a', b']$ and $[c', d']$ that are associated with corridors of the two split-up lines (cf. 1.2(ii) & 1.2(vi)) of the same root-line ("adjacent" in this context does not exclude that reading domains with corridors of the other basic direction can be found in or intersect with $[a', d']$). Then $[a', d']$ must be a reading domain on the $(i-1)^{\text{st}}$ level.

(vii) If $[a', d']$ is a reading domain on the ith level and $a' \neq a$, then there exists $b' \in [a', d']$ so that $[a', b']$ is a reading domain on the $(i+1)^{\text{st}}$ level with a corridor of the corresponding split-up line. Similarly for $d' \neq b$ there exists $c' \in [a', d']$ so that $[c', d']$ is on the level $(i+1)$ a reading domain with a corridor of the other split-up line. The attributes "split-up" and "root" are hence also extended to reading domains. Observe that in particular this rule implies that in general each reading domain has two split-up domains; excepted are just the first and the last letter of each basic direction which may have only one split-up domain. For paths which only exist inside one corridor, by combination of the above rules they may have no split-up domain at all.

For each fixed level index i the reading domains $[a', b'], [c', d'], \ldots$ are put into some ordered tuple $([a', b'], [c', d'], \ldots)$. The rules according to which this order is chosen are as follows:

(viii) If two intervals are disjoint, so that without loss of generality we have $a' \leq b' < c' \leq d'$, then $[a', b']$ comes before $[c', d']$.

(ix) In case of a one-point overlap due to the corner-point intersection (cf. 1.2(v)) or to the tangent-point-case (cf. (ii)) so that we have $a' \leq b' \leq c'$ the order of $[a', b']$ and $[b', c']$ is also chosen naturally, if at least one of the two "$\leq$"-signs is a "$<$"-sign.

(x) There is no demand by this rule in which order two intersecting reading domains are found in our ordered tuple, if their intersection is due to the fact that our path goes through a crossing region of corridors. The freedom of choice is even given, if in the latter case the two intervals should just intersect in one point.

(xi) Let I and J be two reading domains for the index level i, and let I_n and J_m $(n, m \in \{1, 2\})$ be their split-up domains, respectively. In this case, where both reading domains have two split-up letters, we require that some of them come in the same order as their root-domains, i.e. for some m, n the order of I_m and J_m is the same as for I and J on the preceding level.

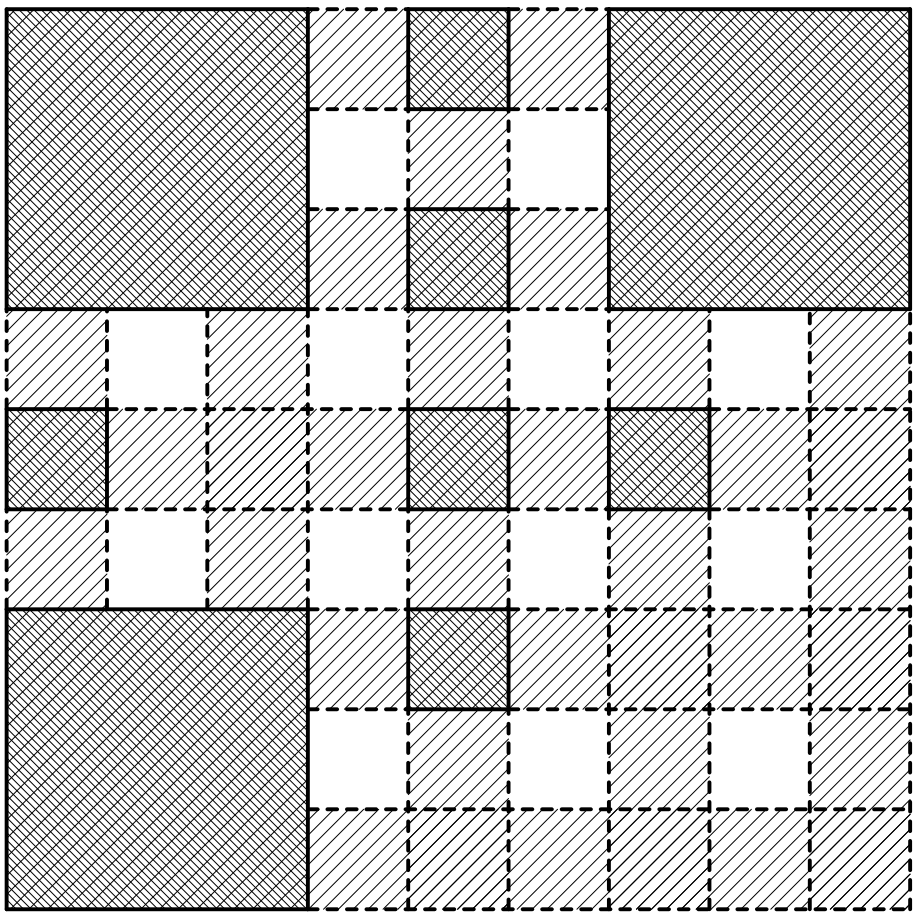


Fig. g2 (belonging to 1.1):
This figure covers the same part of X as Fig.g$_1$. It shows the corridors of the $(i+1)^{st}$ lattice and the corresponding holes.

(xii) In principle the same rule as (xi) is to be applied if one our reading domains should only have one split-up domain, but the other has two. However, in this case the "missing" split-up letter is to be added as an "imaginary letter" either before the first entry of the tuple or after its last entry. According to the "may"-option in (vii) it is clearly defined where the missing domain is to be inserted, and with such an imaginary insertion the same rules as in the preceding case imply.

(xiii) If two reading domains as in (xi) or (xii) both have just one split-up letter, then their order has to be the same if the two letters belong to readings domains of the tangent-point case or the corner-point case, and otherwise their order is free (in the other cases the order is determined anyway).

Having selected and ordered our reading domains, we now can define our word sequences:

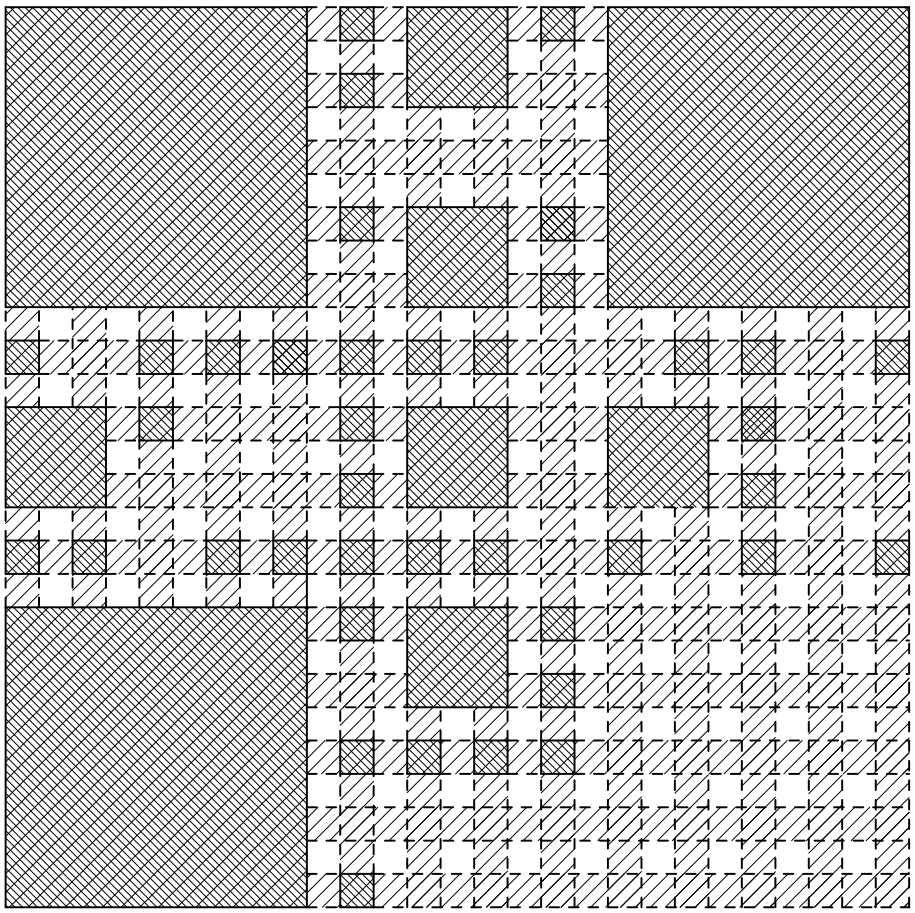


Fig. g3 (belonging to 1.1):
This figure shows how the $(i+2)^{\text{nd}}$ corridor lattice might look like in the same region that has been pictured in Fig.g_1 and Fig.g_2 already.

By (iv)–(v) each reading domain is associated with a (possibly incompleted) act of crossing a corridor into a certain direction, and hence it can be associated with a letter from $\overline{\mathcal{A}}$. We herby introduce the convention that acts of crossing vertical corridors from the left to the right and horizontal corridors into upward direction are converted into letters of $\mathcal{A}$, while acts of crossing into the opposite directions are converted into letters from $\mathcal{A}^{-1}$. That way for each level i the ordered tuple of reading domain is directly converted into a word from $\overline{\mathcal{A}}_i^*$, and, if all tuples are simultaneously considered, into a sequence of words from $(\overline{\mathcal{A}}_i^*)_{i\in\mathbb{N}}$.

1.5 Lemma: *For each path $u : [a,b] \to X$ there exists at least one word sequence which is related to u as in 1.4(i)–(xiii).*

Remark: The proof of this statement is fairly lengthy and rests on the discussion of all combinatorial possibilities for showing that the Conditions 1.4(i)–(xiii) are not contradictory. For that reasons it is postponed to Part C (cf. C.2.1/C.1.9). Note that the context is even that tricky that there exist phenomena such as "non-extendable readings", i.e. there exist paths which permit to adjust reading domains on the levels $1, \ldots, i$ that are compatible with all demands, but so that reading domains on the level $(i+1)$ that still satisfy all demands cannot be found without readjusting some of the domains on the previous levels (cf. C.2.5).

1.6 Definition of the Realization-Mapping and the multivalued Reading-Mapping:
The relation between a path $u \in \mathcal{L}(X)$ and a word sequence that is its reading can be interpreted as a multivalued mapping

$$\Psi : \mathcal{L}(X) \to (\overline{\mathcal{A}_i^*})_{i \in \mathbb{N}}. \tag{1}$$

Unfortunately this mapping is only multivalued, since the reading process as described above does not give a unique description by word sequences, in particular when paths are tangent to corridor boundaries, pass through corridor crossings or start or end inside a corridor. However we will hardly make use of the notation Ψ, since multivalued mappings are not a part of standard mathematical notation. Instead of this we will prefer to call the $\Psi(u)$-images the *"readings of u"*. However, we denote $\mathrm{Im}(\Psi)$ by $\mathcal{A}^{**}$, and call $\mathcal{A}^{**}$ the *"set of all word sequences"* that is defined on the alphabet $\overline{\mathcal{A}}$. It will turn out that Ψ has an inverse mapping

$$\Phi : \mathcal{A}^{**} \to \mathcal{L}(X), \tag{2}$$

and that this mapping is univalent almost everywhere (cf. C.1.2). However, a true understanding of Φ cannot be achieved without being able to combinatorially characterize the subset $\mathcal{A}^{**}$ in $(\overline{A_i})_{i \in \mathbb{N}_0}$. This will be one of our major tasks of Part C (cf. C.1.5). Another task will be to combinatorially describe Φ as a mapping that finds a path that *"realizes a word sequence"* (cf. C.3.1).

1.7 Remark: On how to understand the expression *"letter"* in this paper:
When elements of $\overline{\mathcal{A}}$ are composed to words or even to word sequences (e.g. $w = l_1 l_2 l_3 l_4$ $w \in \overline{\mathcal{A}_i^*}$), then they are regarded as "letters". When single letters of a word or of word sequences are identified by a variable, this identification may be by simple counting of letters (as in the above example: $l_3 := 3^{\mathrm{rd}}$ letter of w), but also a counting relative to the properties "horizontal" or "vertical" or to split-up relations or to the meeting of a certain line of corridors will occur. For that reason a variable which denotes a letter should in general be understood as an operator which is applied to a certain word or to a word sequence and returns two types of information, namely

(1.) the alphabet-component, i.e. which value in $\overline{\mathcal{A}}$ the corresponding letter has (this will be called the *"value of the letter")* — and

(2.) the position information, i.e. the precise information at which place inside the word or word sequence we can find the corresponding letter.

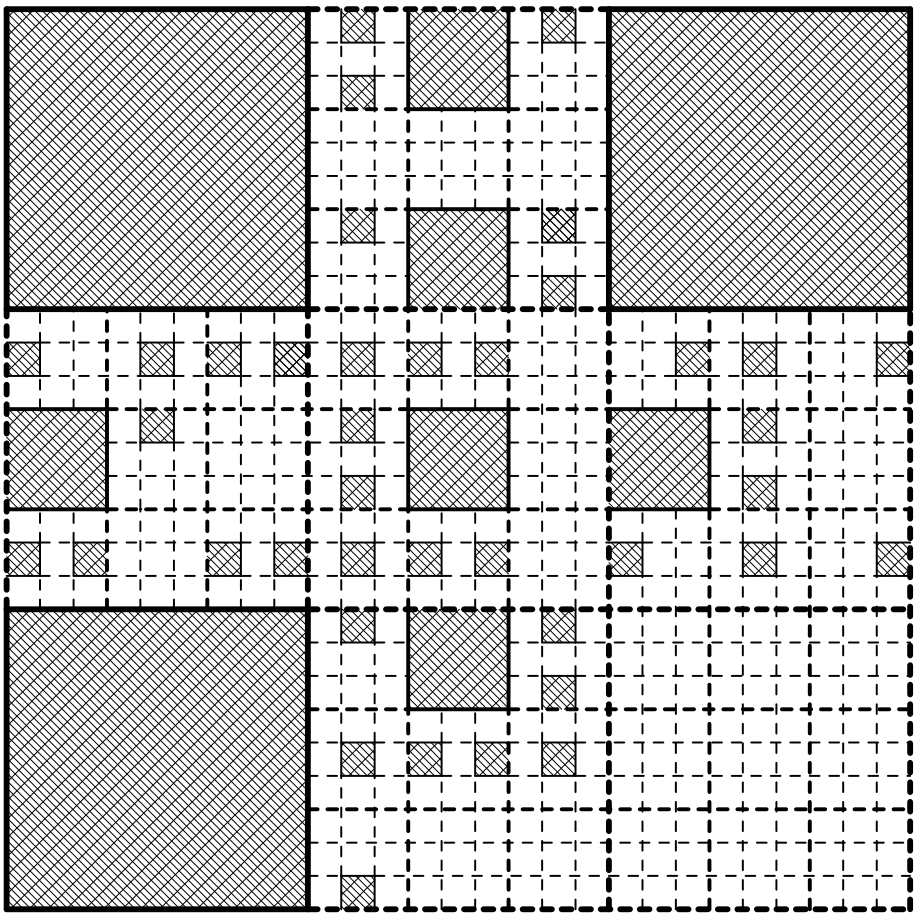


Fig. g4: (belonging to 1.1):
This figure tries to give some impression of the the entire corridor network. For that purpose the three corridor lattices as they have been pictured in Fig.g1–Fig.g3 have been drawn into this figure simultaneously.

When writing up this paper it was decided to keep the language as close as possible to everyday-language, i.e. we want to be able to talk about our words by using the same phrases as when talking about the spelling of ordinary words in the English language. For that purpose it was decided to refrain from introducing projections that map the information returned by a letter-operator to the alphabet- or to the position component, and instead of that we will stick to the following rules:

(i) The relations "=" and "≠", if applied to letter, exclusively refer to the alphabet-component. In particular we are allowed to write $l_i = l_j^{-1}$ independently from where in our word these letters are placed.

(ii) The relations "<", "≤", ">", "≥" exclusively refer to the position component

of letters, i.e. $l_1 < l_2$ means that l_1 is found to the left of l_2 on the same level. Comparing the positions of letters which occur at different levels of word sequences is more complicated and cannot be simply expressed in formulae in this paper.

(iii) Whether two letter-operators agree or do not agree as operators (i.e. whether they identify letters according to the same or to equivalent criteria) will never be expressed in a formula in this paper. Accordingly we do also introduce the convention that "l" may be used as an abbreviation for "$l(\omega)$", i.e. the word or word sequence to which a letter-operator is applied may be omitted if it follows from the context.

(iv) Where letter-variables are referred to in plain text we will choose our expressions in such a way that it should be clear which of the two properties (1.) or (2.) are referred to.

These rules will enable us to discuss the properties of words and of word sequences in a most natural way, and, since we once stated them, be yet precise.

1.8 Definition of commuting letters:
Two elements of $\overline{\mathcal{A}_i}$ are defined to be *"commuting"*, if one of them is horizontal, the other vertical, and if they are associated with two corridors that intersect each other in a crossing region. All other combinations of $\overline{\mathcal{A}_i}$-elements are regarded as non-commuting. In this context "commuting" is merely to be understood as an attribute that is associated to particular pairs of $\overline{\mathcal{A}_i}$-elements. This attribute will be used to define the equivalence relations in Section §2. On the other hand, the order of letters in each word or word sequence is always clearly defined, even if commuting letters should be found in adjacent positions!

1.9 Remark: In C.1.11 we will see that $\mathcal{A}^{**}$ has the structure of a generalized inverse limit. The attribute "generalized" must be included due to the fact that due to similar limit-cases as described in 1.4–1.6 the projections $\mathcal{A}^{**} \cap \overline{\mathcal{A}}_i \to \mathcal{A}^{**} \cap \overline{\mathcal{A}}_{i-1}$ can only be defined as multivalued mappings, also. However, until we have not reached a combinatorial understanding of $\mathcal{A}^{**}$, we are not able to introduce these projections, anyway.

§2. Equivalence classes of word sequences

Recall that in Section §1 we introduced word sequences as combinatorial concepts in order to translate the geometric task of constructing a homotopy combing (cf. A.5.6(iii)) of our space X in Normal Form into a combinatorial one. By 1.6 we implicitly defined two equivalence relations in $\mathcal{A}^{**}$ namely:

$$\omega_1 \sim_1 \omega_2 \iff \Phi(\omega_1) = \Phi(\omega_2), \tag{1}$$

$$\omega_1 \sim_2 \omega_2 \iff \Phi(\omega_1) \text{ is relatively homotopic to } \Phi(\omega_2). \tag{2}$$

For reducing the task of chosing the paths of a homotopy combing to a combinatorial problem, It will be necessary to be able to describe "$\sim_1$" and "$\sim_2$" in a combinatorial

way. From a technical point of view we will give combinatorial definitions for the corresponding equivalence relations below, and later prove that the so defined equivalence relations satisfy the properties of "$\sim_1$" and "$\sim_2$" as above (cf. D.12.1/D.12.3).

2.1 Basic criteria for the equivalence relations to be defined: We will need three types of equivalence relations (weak, strong, firm) for word sequences (and later in D.3.6, by refining these definition, we will actually come up with nine different relations). However, the skeleton of the definition of all three relations is the same, and this is what is to be explained in this paragraph:

In all cases we will define that two word sequences ω_1 and ω_2 are equivalent if and only if for all $i \in \mathbb{N}$ the ith word of ω_1 can in such a way be transformed by iteratively applying certain elementary processes into the ith word of ω_2 that the transformation processes on the levels i and $i+1$ can be synchronized for all i and that the nesting depth of certain involved transformation processes is bounded. As stated before (cf. Theorems D and E), the various equivalence relations only differ by specifying the words "certain" in the above definitions. Hence in any case there will be three types of demands on the definition of an equivalence relation:

(i) *"Process demands":* We will have to give the precise list of elementary processes that may be applied for words. Roughly speaking (cf. D.1.1 and for exceptions cf. D.1.1(iii)) these processes will reflect how the reading of a path may change if a path is in such a way slid around in the plane that its intersection behavior with the grid-system changes essentially (cf. D.1.1).

(ii) *"Compatibility demands":* We will have to define when sequences of transformation processes on adjacent levels are synchronizable. It should not be surprising that there have to be such demands: If we are sliding around our path in X, then the equivalence relation has to reflect how the reading of a path may be affected by the same sliding processes for all the levels (cf. D.1.2).

(iii) *"Boundedness conditions":* A natural number is to be computed from the sequence of elementary transformations that brings the ith word of ω_1 into the ith word of ω_2 for each i separately. The boundedness conditions require that, depending on which operations are relevant for the particular relation, this number has to be bounded with respect to all i for a given system of equivalence transformations (cf. D.1.6).

In the following we define processes and phrases that are necessary to state the definition of our equivalence relations:

2.2 List of elementary processes: (cf. D.1.7)

This paragraph is devoted to define those elementary processes that will occur in the equivalence relations to be defined in this section:

(i) An *"elementary cancellation"* is a process which deletes two adjacent letters in a word and leaves the remains of the words unchanged (cf. D.1.1(i)). It is applicable, if these letters are inverse to each other. The inverse process is called an *"elementary insertion"*.

(ii) An *"elementary transposition"* is a process which interchanges two adjacent letters in a word and leaves the remains of the words unchanged (cf. D.1.1(ii)). It is applicable, if these letters are commuting (cf. 1.8).

(iii) An *"elementary truncation"* is a process which deletes either the first or the last letter of a word. Correspondingly we will make a distinction between "truncations to the end of a word" and "truncations to the beginning". These processes need also to be distinguished if a one-letter word is reduced to the empty word. In principle there are not any explicit restrictions for applying this process, however the combination of boundedness- and compatibility conditions will give some implicit restrictions (cf. 2.1/2.9). The inverse process is called an *"elementary appendance"* (cf. D.1.1(iii)).

(iv) A *"far-distance cancellation"* is a process which allows to remove l_0 and l_n from a substring $l_0 l_1 \dots l_n$ of a word ω, provided that l_0 is inverse to l_n and that each of the $l_i \quad (1 \leq i \leq n-1)$ commutes with l_0 (cf. D.1.7)

(v) In analogy to (i) and to (iv) *"far-distance truncations"* are also permitted. They can be applied to take out letters of a word that only have commuting letters as its predecessors or as its successors. The inverse processes will then be called *"far-distance appendances"* (cf. D.1.7).

2.3 Definition: The above five processes are subdivided into *"reducing"* and *"enlarging ones"*. The rules are as follows:

(i) (far-distance) cancellations are reducing, (far-distance) insertions are enlarging.

(ii) (far-distance) truncations are reducing, (far-distance) appendances are enlarging.

(iii) For transpositions the rules are as follows:

Since two letters can only be commuting, if they belong to different basic directions, there are essentially eight types of pairs of adjacent letters to which such a process might be applied: They are determined by whether the letters belong to $\mathcal{A}$ or to $\mathcal{A}^{-1}$ and by which of the two letters is the horizontal one. It depends on which two of these types is to be transformed into another, whether a transposition is regarded as a shortening or as an enlarging process. The following formula lists precisely the four possibilities for a shortening process:

$$(1) \qquad \begin{array}{ll} h\,v \longmapsto v\,h & v^{-1}\,h^{-1} \longmapsto h^{-1}\,v^{-1} \\ h\,v^{-1} \longmapsto v^{-1}\,h & v\,h^{-1} \longmapsto h^{-1}\,v \end{array}$$

where $v \in \mathcal{A}^{\mathrm{v}}$, $h \in \mathcal{A}^{\mathrm{h}}$. The inverse processes are then considered as enlarging ones (cf. D.3.10).

2.4 Remark: For words it is evident that far-distance processes 2.2(iv)-(v) can always be replaced by the ordinary processes 2.2(i)–(iii). For word sequences this result is not obvious, since substitutions of far-distance processes are only valid if they can be compatibly extended throughout the entire word sequence. For that purpose it seems most convenient for our proof having introduced the far-distance processes from the very beginning, so that the problem as described in the previous sentence need not to be discussed in this paper at all. The need to regard the far-distance processes as single processes comes from another technical aspect: examples can easily be constructed where a far-distance cancellation, i.e. a shortening process, can only be replaced by transpositions and an elementary cancellation, if one includes enlarging transpositions. The standard example for such a situation is "$v^{-1}hv$", where v denotes a vertical and h a horizontal variable taken from $\mathcal{A}$. If they are commuting, then this string can be

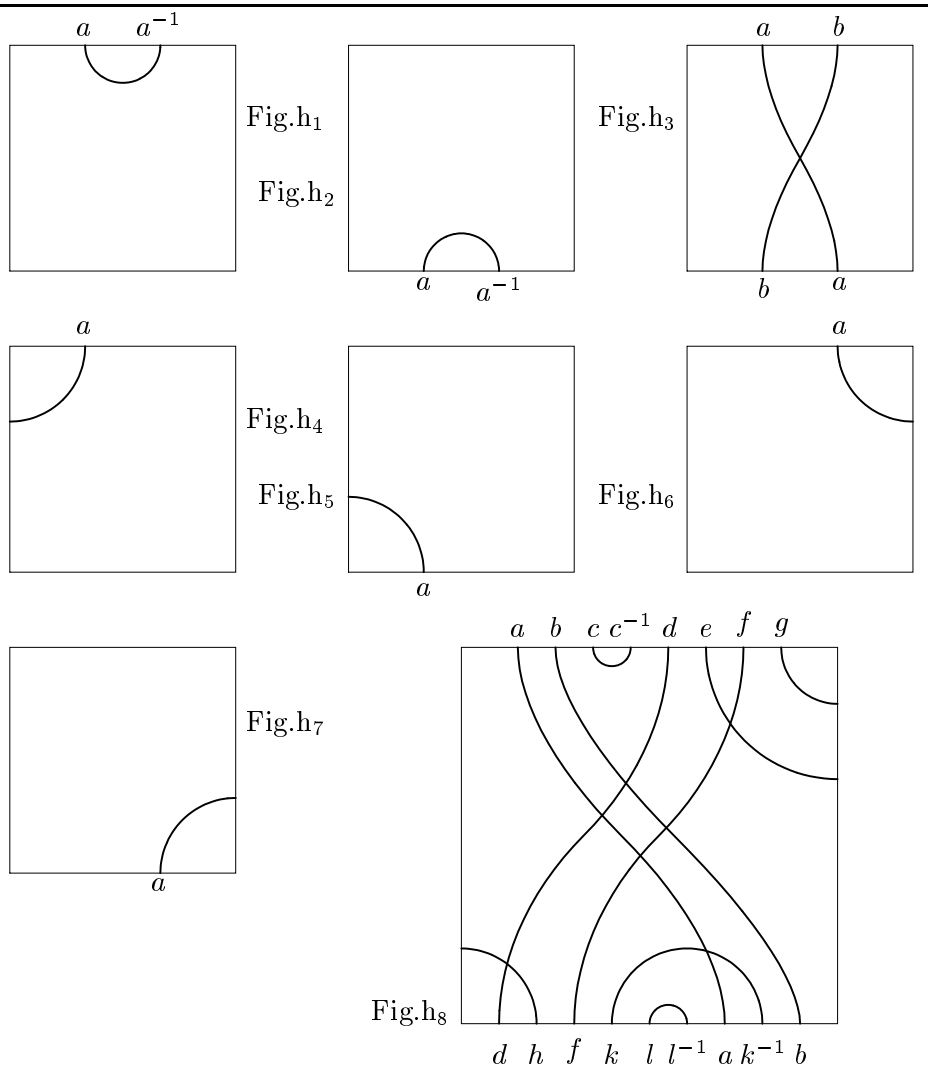


Fig. h (belonging to 2.5):
This figure shows the transformation diagrams for the elementary processes and gives one impression how a combination of these elementary components looks in a concrete example of transforming two words into each other.

reduced to "h", but each of the possible ways of bringing v^{-1} and v into an adjacent position is an enlarging process according to 2.3(1). This fact that such examples exist would make a result such as 2.12(iii) impossible if 2.2(i)–(iii) were the only legitimate operations.

The purpose of the following definition is to visualize a sequence of transformations that are applied to a word. This method of visualization will be restricted to non-far-distance operation 2.2(i)–(iii), however for the subsequent definition of the nesting depth, since it will be defined on a fixed level only, this is not relevant by 2.4:

2.5 Definition of transformation diagrams: Fig.h$_1$–Fig.h$_7$ correspond to the elementary processes, cancellation, insertion, transposition, truncation to the beginning,

truncation to the end, appendance to the beginning and appendance to the end, respectively. By composing these elementary pictures similar as in Fig.h_8 any longer sequence of elementary transformations can be visualized. The resulting graph is then called a *"transformation diagram"* (cf. D.1.3).

2.6 Definition: (cf. D.1.4)

(i) Given a sequence of transformations for a word. Then the *"nesting depth"* is an integer which is computed from the transformation diagram as follows: Each region into which the lines of our transformation diagram have subdivided the domain gives rise to a vertex, and two vertices are to be joined by an edge if they belong to regions that have some boundary segment in common. Those vertices of the dual graph that belong to a region that actually touches a part of the upper bar get a "top"-label, and corresponding "bottom"-labels are also attached. Each edge in the dual of the transformation diagram gets the combinatorial length 1, the length of an edge-path in this graph is given by counting the number of edges. In order to compute the nesting depth we look at all top-vertices and at all bottom-vertices, and for each of these vertices we look for the shortest edge-path to any vertex with the opposite label. The maximum over these shortest edge paths to the nearest partner vertices is then the "nesting depth" of the corresponding transformation process. If some vertex should have both, the "top"- and the "bottom"-label, then "zero" can be used as the length of the shortest path to the vertex with opposite label for these start points.

(ii) The *"truncation depth"* of a sequence of transformations for a word is similar computed as the nesting depth, however in this case all lines that do not end at the side bars of our transformation diagram are removed, i.e. all lines that do not involve truncations. From this diagram the truncation depth is computed by the same method as it was the nesting depth from the original transformation diagram (cf. D.1.5).

Note that by definition the truncation depth is always smaller than or equal to the nesting depth.

2.7 Definition of compatible transformations (cf. D.1.8):

Let $\omega_i, \eta_i, \omega_{i+1}, \eta_{i+1}$ be words in $\overline{\mathcal{A}_i}$ and $\overline{\mathcal{A}_{i+1}}$, respectively. Then the elementary transformation that turns ω_{i+1} into η_{i+1} is called "compatible with respect to the one that turns ω_i into η_i", if apart from (iv) either (i)–(ii) or (ii)–(iii) are satisfied:

(i) $\omega_i = \eta_i$ or $\omega_{i+1} = \eta_{i+1}$.

(ii) η_i either equals ω_i or it differs from ω_i by one of the elementary transformations 2.2(i)–2.2(v) only; the same holds for η_{i+1} and ω_{i+1}.

(iii) If we have $\omega_i \neq \eta_i$ and $\omega_{i+1} \neq \eta_{i+1}$, then the transformation process on the level $i+1$ is of the same type 2.2(i)/(iv), (ii)/(v) or (iii) for the two index levels, and in addition on the index level $i+1$ it only affects one of the split-up variables that had been affected for the index level i.

(iv) In addition we require that in general we only discuss whether a (2×2)-matrix $\left(\begin{smallmatrix} \omega_i & \eta_i \\ \omega_{i+1} & \eta_{i+1} \end{smallmatrix}\right)$ describes compatible transformations if there exist two paths u and v so that ω_i and ω_{i+1} are in $\Psi(u)$ and η_i, η_{i+1} are in $\Psi(v)$.

2.8 Definition of synchronizable sequences of transformations (cf. D.1.16):

Let $(\omega_{i,1},\dots,\omega_{i,l_i})$ and $(\omega_{i+1,1},\dots,\omega_{i+1,l_{i+1}})$ be tuples of words over $\overline{\mathcal{A}_i}$ and $\overline{\mathcal{A}_{i+1}}$, respectively, so that any two neighbouring entries of these tuples just differ by one of the elementary transformations 2.2(i)–2.2(v). Then $(\omega_{i,1},\dots,\omega_{i,l_i})$ and $(\omega_{i+1,1},\dots,\omega_{i+1,l_{i+1}})$ describe two synchronizable sequences of transformation processes if it is possible by a (possibly iterated) doubling of some of the entries of these tuples to enlarge them to $(\omega'_{i,1},\dots,\omega'_{i,l'_i})$ and $(\omega''_{i+1,1},\dots,\omega''_{i+1,l'_i})$ (i.e. $\omega'_{i,1}=\omega_{i,1}$, $\omega'_{i,l'_i}=\omega_{i,l_i}$, and if $\omega'_{i,m}=\omega_{i,k}$, then either $\omega'_{i,m+1}=\omega'_{i,m}$ or $\omega'_{i,m+1}=\omega_{i,k+1}$, similar for the double-primed ω-tuples with first index $i+1$), such that for each j with $2\le j\le l'_i$ we have that $(\omega'_{i,j-1},\omega'_{i,j})$ and $(\omega''_{i+1,j-1},\omega''_{i+1,j})$ are compatible transformations in the sense of 2.7. We call the dashed and double-dashed ω-tuples *"the synchronizised transformation processes"* of the originally given ones.

2.9 Definition of our equivalence relations:
Based on the framework as described in 2.1 and the preceding definitions 2.2–2.8 we can now define that basic three relevant equivalence relations as follows:

(i) Two word sequences are *"weakly equivalent"* if for each level the words can be transformed into each other by a finite sequence of elementary transformations such that on two adjacent levels these sequences are synchronizable in the sense of 2.8 and such that the corresponding sequence of truncation depths is bounded (cf. 2.6(ii)).

(ii) If two word sequences are weakly equivalent with respect to transformation pattern that has also bounded nested depth (cf. 2.6(i)), then these word sequences are even considered as *"strongly equivalent"*.

(iii) If two word sequences are strongly equivalent with respect to transformation pattern that only involves transpositions but not the other elementary processes, then they are even called *"firmly equivalent"*.

As already indicated in first paragraph of Section §2, it will be one of the hard tasks of this paper to prove in Section D the two following

2.10 Theorems: *Let X be a space in Normal Form and let $\omega,\eta\in\mathcal{A}^{**}$ where $\overline{\mathcal{A}}=\overline{\mathcal{A}}(X)$ denotes the associated alphabet. Then*

(i) *ω and η are weakly equivalent word sequences if and only if $\Phi(\omega)$ and $\Phi(\eta)$ are relatively homotopic paths (cf. D.12.1).*

(ii) *ω and η are strongly equivalent word sequences if and only if $\Phi(\omega)=\Phi(\eta)$ as an abstract path (cf. D.12.3).*

Before we can state the theorem that gives the relevance of firm equivalence, we need one more

2.11 Definition:

(i) Let ω and η be two weakly equivalent word sequences. Then ω is called a *"reduction of η"* if some transformation pattern that turns η into ω involves reducing elementary transformations in the sense of 2.3, only.

(ii) ω s called a *"reduced word sequence"* if there does not exist $\omega'\neq\omega$ which is a reduction of ω.

The relevance of the following theorem will be explained in Section §3:

2.12 Theorem: *Let X by a space in Normal Form and let $\overline{\mathcal{A}} = \overline{\mathcal{A}}(X)$ be the associated alphabet. Then*

(i) Any weak equivalence class of word sequences over $\overline{\mathcal{A}}$ permits reduced representatives.

(ii) Any two reduced representatives of the same weak equivalence class are firmly equivalent.

(iii) For each word sequence $\omega \in \mathcal{A}^{**}$ there exists a weakly equivalent reduced word sequence $\overline{\omega}$ such that ω can be transformed into $\overline{\omega}$ by using reducing elementary operations only.

Remark: The proof of this theorem is postponed to Part D (cf. D.7.9). Part (iii) will come out from the method of proof (cf. D.7.10/D.7.13).

§3. Structure of the Proof of the Combing Theorem

3.1 Basic Idea: The particular way to construct an abstract homotopy combing for a space in Normal Form is given by choosing with x_0 an arbitrary base point of X and by letting A_{x_0} be the set of all relative homotopy classes of paths that start in x_0 and may end up anywhere in X. Then

$$C_X := \{\Phi(\omega) \mid \omega \in \mathcal{A}^{**} \text{ is a reduced representative of } \alpha \in A_{x_0}\}. \tag{1}$$

Since by 2.12(i) reduced representatives exist and are all strongly equivalent, by 2.12(ii)/ 2.10(ii) the $\Phi(\omega)$-value does not depend on the precise representative ω. Hence C_X is well-defined as an abstract combing. Thus, once we will have proved Theorems 2.12 and 2.10 in D.12.4, the proof of the Combing Theorem A.5.6 will be reduced to showing

3.2 The Lifting Theorem: *The homotopy combing which can be constructed via the reduced representatives of word sequences can be lifted to a continuous parametrized homotopy combing in the sense of A.5.5(iii).*

Remark: It will be the main task of Part E to prove this theorem. In the following we give a brief outline of what it takes to prove 3.2:

3.3 Remark on the Lifting Problem: Since reduced word sequence only determine abstract paths, proving Theorem 3.2 in principle requires some parametrization for each combing path so that homotopically near combing paths are parametrized in such a way that they come out also close in the supremum metric. Unfortunately we will not be able to chose the corresponding parametrization of combing paths from constructions as they have to be performed in the course of constructing Φ in Part C; our construction of Φ will find some parametrization for our paths, but these parametrizations will not fulfill enough continuity properties to ensure the continuity of the desired lifting $\widetilde{C_X}$. One of the main technical tasks of Part E will be finding this new parametrization of our path-system ("*synchronous parametrization*"). Fortunately it will turn out that word sequences provide enough data for choosing this parametrization. However, the

task still requires great care and effort, mainly because word sequences as combinatorial objects may react by abrupt changes of their entries, while the parametrization to be constructed must respond in a continuous way. However, these problems will be solved in Part E: Here a definition will be given that together with the properties of reduced word sequences makes it possible to give an ε-δ-estimate that will prove that parametrized combing paths that are associated with homotopically near relative homotopy classes remain close to each other with respect to the supremum metric, also. In Part E (cf. E.3.2/E.3.10) this step will conclude the proof of our Lifting Theorem 3.2, and thus of our Combing Theorem A.5.6, and hence of Theorem A.

Abschnitt C: Proofs of Section §B.1

Recall that the purpose of this section is to give a combinatorial description of word sequences which in B.1.6 only had been defined as $\mathrm{Im}(\Psi) \subset (\overline{A_i})_{i\in\mathbb{N}}$, to prove the well-definedness and the existence of the reading mapping Ψ (cf. B.1.5) and to combinatorially characterize its inverse, i.e. the realization mapping. Thus, in brief, this section is devoted to establish the main facts that we need when we want to translate the task of choosing a homotopy combing into a combinatorial task.

§1. Combinatorial Description of word sequences

1.1 Remark: *There will be no complete annotation system for the system of corridors and the associated alphabet as introduced in B.1.1 and in B.1.3, respectively. Such a complete system would have to characterize each corridor by giving the following pieces of information:*
(i) whether the corridor is horizontal or vertical,
(ii) the index of the corridor lattice to which it belongs,
(iii) the numbers of the according line of corridors within the lattice, to which our corridor belongs, and
(iv) the coordinates of the two holes at which the corridor begins and at which it ends.
In addition, if we introduced the system of four or five indices such that each corridor has an individual name that comes along with the complete set of data, we would have also to invent a philosophy for using variables for describing those data of the full index system of a corridor that do not matter or are undefined in a particular context. Overall, it seems not to be of help for anybody to introduce a system for encoding the geometric relations between corridors into such an index system and bother the reader with the need to permanently decode them again.

1.2 Remark: Recall that in B.1.6(2) we did only claim that Φ, i.e. the mapping that associates a path with a given word sequence is univalent "almost everywhere". It will

turn out that the ambiguity is due to paths that are entirely horizontal or entirely vertical (or are even constant), since in a case where only crossings of vertical or only of horizontal corridors are reported, it might not be possible to locate the precise constant coordinate of a path in the plane. In order to overcome this ambiguity we will in this section extend the concept of word sequences to a concept of "triple sequences" (cf. 1.4/1.5), so the the information that is contained in a word sequence is just a subset of the information that is contained in a triple sequence. If extended to triple-sequences, our mapping Φ will be univalent without exceptions. On the other hand, since the information that is contained in word sequences is apart from limit cases usually sufficient, we will mainly argue be using word sequences and will refer to triple sequences only where necessary.

1.3 Remark: Unlike as for the non-uniqueness of Φ as described in 1.2, we will not be able to overcome the non-uniqueness of Ψ (cf. B.1.6(1)) by similar extension facilities for the reasons described in A.4.9, and hence we will have to continue to regard Ψ only as a multivalued mapping. The purpose of this remark is summarizing the geometric origins for the non-uniqueness of Ψ that were hidden in the technical demands of B.1.4. There are essentially three of them:

(i) If a path goes all the way through a corridor till it touches the opposite boundary, but then turns around and goes back into the interior, then the act of crossing this corridor might be regarded as completed at the time the boundary line is touched, or it might not. We will talk of *"acknowledging"* or of *"regarding* a tangency" if we should decide to treat a mere tangent point as the start- or the endpoint of a path that performs an act of crossing, and of *"ignoring"* or of *"disregarding* a tangency" otherwise.

(ii) If a path goes through a crossing region of corridors it in general performs the act of crossing a vertical corridor and of crossing some horizontal one at the same time. Hence two letters are to be put at the corresponding position of our word sequence, but their order can in general not be defined from geometry in such a way that it is consistent with respect to all limit processes.

(iii) If our curve ends in the middle of some corridor, so that it only has partially performed the act of crossing this corridor, might it be necessary even to acknowledge such a partial act of crossing by a letter in the corresponding word sequence? — Since we can easily think of a situation that the concatenation of two curves each of which only partially crosses a certain corridor gives a curve which performs a complete act of crossing, we will have to get used to the idea that if the initial point or the endpoint of a curve falls inside the domain of a corridor, we are in principle given the choice whether to regard the corridor as been crossed or not.

1.4 Remark: In particular for being able to adequately handling the uncertainties imposed by the the latter phenomenon 1.3(iii) and those described in 1.2 we will introduce word-sequences as motivated in A.4.5–A.4.6 and as defined in B.1.4/B.1.6 only as part a part of a more complex combinatorial concept which will be called "triple sequences". This combinatorial concept is an extension of word sequences in so far as the sequence of the middle entries of the triples just gives the word sequence again. The sequence of the first entries of those triples will give a description of the initial

point of the path by a two dimensional interval nesting of extended lattice squares. The third entries of the triples are devoted to analogously describe the endpoint of our path. These descriptions are not unique if the corresponding point lies inside a corridor, since corridors are overlap-regions of the adjacent (extended) lattice squares (cf. B.1.2(iv)). In this case it is a kind of arbitrary decision to assign our point to one of the adjacent lattice squares in the course of the description by the first or third entries of our triples. However triple-sequences will then require consistency between this assignment and the question of 1.3(iii), i.e. whether a partial corridor crossing when performed by the initial or terminal segment of a path should be acknowledged by a letter in the corresponding word sequence. Finally, the sequence of the middle entries of the triples will be the word sequence as defined in B.1.4/B.1.6. Hence in principle it requires a prove that the implicit definition of word sequences by the following definition 1.5 of triple sequences is consistent with was has been defined in B.1.6; however it will be fairly obvious that essentially 1.5 is just a combinatorial rephrasing of the demands of a reading in B.1.4 and of basic properties from the continuous behaviour of paths (e.g. that by intermediate-value properties a path cannot skip a line of corridors). Thus we will come back to this point in the paper.

1.5 Definition of triple sequences:
Let X be a space in Normal Form (cf. A.3.3), $\mathcal{N}$ be its corridor network (cf. B.1.1) and let $\mathcal{A}$ be the associated alphabet. Apart from $\mathcal{A}$ as defined in B.1.3 we introduce a second alphabet: Let $\mathcal{B}_i$ be a set which is bijectively associated with the set of lattice squares of $\mathcal{N}_i$ and let $\mathcal{B} = \bigcup_{i=1}^{\infty} \mathcal{B}_i$. Symbols of $\mathcal{B}$ cannot be inverted. Assume that $u_i, v_i \in \mathcal{B}_i$ and that ω_i is a finite word in the alphabet $\overline{\mathcal{A}}_i$ and that for each $i \in \mathbb{N}$ such u_i, v_i and ω_i are defined. Then $(u_i, \omega_i, v_i)_{i\in\mathbb{N}}$ are a *"triple sequence"* provided they satisfy the following three conditions:
(i) For all $i \in \mathbb{N} - \{1\}$ the u_i and v_i must be subregions of the extended lattice square u_{i-1} and v_{i-1}, respectively, i.e. they can only be a subregion of the corresponding pure lattice square or of one of the adjacent corridors, or of one the regions where those corridors intersect each other.
(ii) The *"internal compatibility"* that will be defined below in 1.6, and
(iii) the *"external compatibility"* that will be explained after the internal one in 1.7.

1.6 Definition of the internal compatibility of word- and triple sequences:
In brief, this condition on word sequences states for each $i \in \mathbb{N}$ that not any combination of letters of $\overline{\mathcal{A}}_i$ and lattice squares of $\mathcal{B}_i$ can give the entries u_i, v_i and the letters of ω_i of the corresponding triple sequence, but only such combinations are permitted that are compatible with the geometry of the corridor-network $\mathcal{N}_i$.
Since by 1.1 we decided not to introduce an index system for describing the alphabets $\mathcal{A}$ and $\mathcal{B}$, we now need to refer to geometry to describe this condition:
The first letter of ω_i must be one of the adjacent corridors of u_i, and it must indicate an outward directed crossing of one of these corridors, since a path having started in u_i can, when firstly crossing a corridor only go into one of the adjacent lattice squares. This lattice square is uniquely determined by u_i and the first letter of ω_i. Since by its next act of crossing our path has to leave that lattice square again, by u_i and by the first letter

of ω_i the second one is again determined up to at most four possibilities. Similarly each letter of ω_i is determined by its predecessors and by u_i up to at most four possibilities. We can only say "at most" since, if the corresponding lattice square is adjacent to one big hole with an index smaller than i, this implies a geometric obstruction for leaving the square on the corresponding side, and in this case the next letter is determined up to only three possibilities, further restrictions may hold near the corners of X_0. On the other hand there is not any condition which prohibits two adjacent letters of ω_i from being inverse to each other; we do not require that ω_i is a reduced word. The principle explained above allows one to compute the lattice square in which our path must end from u_i and all of the letters of ω_i. This lattice square must be v_i. It is the part 1.5(ii) of the definition of a triple sequence that u_i, v_i and the letters of ω_i do only occur in an order and with exponents $()^{-1}$ partially attached to them in such a way, that they are compatible with respect to these geometric relations between the corridors and lattice squares that are associated to the letters of ω_i and to u_i and v_i.
A graph-theoretical picture may be used to express this internal compatibility even more precisely: Construct the dual graph to the given corridor-lattice $\mathcal{N}_i$. I.e. each lattice square gives rise to one vertex in this graph, and any two adjacent (but not diagonally adjacent) lattice squares are to be connected by an unoriented edge, or even better, by a pair of oriented edges, one edge for each of the possible orientations. Each of the oriented edges is then labelled so that the label describes the act of crossing as performed by the edge according to the convention as given in the final paragraph of B.1.4. Observe that hence we may have different edges with the same label in our graph, namely if a corridor goes over various crossing regions, then all the crossing dual edges (there will be a pair for each segment of the corridor) will have the same label. On the other hand there will be never two different edges with the same label the emanate from the same vertex in our dual graph. Hence, given a vertex as a start point, any tuple of edge-labels can describe at most one edge-path in the dual graph. In that language the internal compatibility 1.6 requires that the words of our word sequence may only be valid tuple-descriptions of edge paths in that dual graph with the additional requirement that the start point of the edge-path is that vertex that is associated with the first entry of the triple, and similarly for the endpoint and the third entry.

Recall that the subsequent definition gives together with the preceding one the two essential conditions on triple- and on word sequences (cf. 1.4–1.5). Now 1.7(i) should be understood as the basic condition of the external compatibility, while 1.7(ii)–(iii) are the "exceptional rules", that weaken 1.7(i) near the beginning and the end of each word.

1.7 Definition of the external compatibility of word- and triple sequences:
In the following we describe a process for how out of u_i, v_i and ω_{i+1} a set of words in $\overline{\mathcal{A}}$ can be computed. It is the part 1.5(iii) of the definition of a triple-sequence that ω_i must be contained in this set which, in analogy to B.1.4, will be called the *"set of possible readings* of ω_{i+1}":

(i) Write down ω_{i+1}. Remove all variables that belong to non-split-up corridors (i.e. to corridors that are not contained inside corridors of the prior level, cf. B.1.2(vi)). Then in particular consider one letter a, and note that by 1.6, if not the end of the

word ω_{i+1} is found instead (cf. (ii)–(iii)), the preceding or the subsequent letter of the same basic direction as a must either be a split-up-companion of a (i.e. a letter with the same exponent as a taken from a neighbouring line of $\mathcal{N}_{i+1}$-corridors such that a and this letter belong to corridors that are both covered by the same $\mathcal{N}_i$-corridors), or it must be a letter of the same line of corridors as a and then have the inverse exponent with respect to a.

In the latter case we treat the corresponding pair of letters as a cancelling pair, and do not pay any more attention to them, in the first case we put an overbrace over these two letters. Observe that by geometry such two letters together describe an act of crossing of an $\mathcal{N}_i$-corridor. The $\overline{\mathcal{A}}_i$-letter that stands for the corresponding act of crossing is attached as an annotation to the vertex of that brace. That way (with exceptions near the beginning or the end of a word, cf. (ii)–(iii)) each of those letters which belong to a split-up-corridor is either associated to a cancelling pair or to one of these braces. Note that braces belonging to horizontal letters may overlap with braces belonging to vertical letters. In case all these braces are disjoint, a word in $\overline{\mathcal{A}}_i$ can be uniquely read from their annotations. In case they are not, a certain set of word is determined that way, namely variables belonging to braces with overlapping domains may be put in each of the two possible orders. It is a part of the definition of a triple sequence that one of these words must be ω_i.

(ii) The purpose of this item is to describe a case not treated in (i): When looking in ω_{i+1} for the corresponding split-up-partner or the cancelling partner of a letter, the end of a word may be found instead. In this case we are given an additional degree of freedom, when computing the possibilities for ω_i: We are as well allowed to treat this letter as a letter which has neither a cancelling partner nor a split-up partner, and, apart from this, we are also allowed to treat this letter as a letter which has a split-up-partner and to put some brace between this letter and the corresponding end of our word, and to annotate the brace as if some *"ghost letter"* beyond the end of our word would be our split-up partner (cf. (vii)&(xii) in the description of the reading process B.1.4). Note that only when working with word sequences alone (cf. 1.9), we enjoy the full freedom just described; when working with triple sequences, then only one of the above choices can be compatible with the corresponding value of u_i or v_i, respectively.

(iii) If a word has not any letters of one of the two basic directions, we are, similarly as in (ii), allowed to span a brace between the beginning and the end of a word. Such a brace may be annotated as if the corresponding ghost letters together are split-up partners of the basic direction whose letters are not present in the word. Again, when working with triple sequences, it is up to the choice of u_i and v_i, whether such a brace has or has not to be inserted.

We note that in principle the above conditions (i)–(iii) precisely require that, if from the sequence of $\overline{\mathcal{A}}_j$-letters it can be told that a certain $\mathcal{N}_j$-corridor is crossed, the corresponding $\mathcal{N}_j$-letter has to appear in the jth word. Conversely, other letters than those must not appear on the jth level, apart from such letters that indicate a partial

act of crossing as performed by the initial or by the terminal segment of our path (where one by 1.3(iii)) is given the freedom to or not to acknowledge it by a letter.

1.8 Remark: By 1.7, two adjacent words ω_{i+1} of a word sequence have to be in a certain relation. This relation is precisely described by 1.7(i)–1.7(iii) by giving the rules which determine the possibilities for ω_i when ω_{i+1} is given. In this remark we want to complete the picture by taking the opposite point of view, i.e. by describing how ω_{i+1} is allowed to look like if ω_i is given. Unlike as in Definition 1.5 this cannot done in a precise way an more; however having taken this point of view once seems to be of help for the forthcoming arguments, even if it does not result in a precise equivalent statement to Definition 1.7

(i) The standard case for a letter of ω_i is having two split-up letters in ω_{i+1}. If two adjacent ω_i-letters are not commuting (cf. B.1.8) the split-up letters have to show up in the same order as the ω_i-letters since the geometry resulting from disjoint corridors does not permit other possibilities. In case the two ω_i-letters commute, there is more freedom for the order of the split-up letters: Note that the internal compatibility 1.6 clearly defines which of the split-up letters of one letter comes first and which comes second, so that the order of the two split-up letters that result from one letter of ω_i is determined, even if the precise corridor from which they are taken may not be. This should be kept in mind if in the remainder of this paper we are going to use expressions like *"left split-up letter"* and *"right split-up letter"* (cf. B.1.2(vi) in the network-definition). On the other hand, the order in which the four split-up letters of two commuting neighbouring ω_i-letters may show up is fairly free; any order is permitted as long as not both split-up letters of the right ω_i-letter show up before both ones of the left ω_i-letter (cf. (ii)), and as the above rule concerning the order of the split-up partners of the same letters is obeyed. Note that ω_{i+1} in general consists of more letters than there are split-up letters. In many cases the geometry of corridors determines that an additional non-split-up line of corridors has to be crossed between two split-up lines. Even in cases where such occurrence of non-split-up letters is determined, additional letters from split-up lines and non-split-up lines may show up as long as those letters only describe cancelling pairs or encirclements of holes, which do not imply a crossing of some corridor of the lattice $\mathcal{N}_i$. The following is a general rule without any exceptions: Any combination of $\mathcal{N}_{i+1}$-letters which combinatorially describes an entire crossing of an $\mathcal{N}_i$-corridor must give that the corresponding act of crossing is described by an appropriate letter on the $\mathcal{N}_i$-level of ω. A path can do a lot of fuzz ($\widehat{=}$ fine movements) on the $(i+1)^{\text{th}}$ level, which is not yet visible on the ith level. In addition, note that there are exceptions from the general rule of thumb that each letter on the ith level has to induce two split-up letters. For each of the basic directions the first letter is allowed to have only one of its split-up letters (namely the one taken from those split-up line of corridors which according to the general case should appear in the second position), and similarly for each of the basic directions only the first of its two split-up letters of the last letter must show up. Even more, in case of words which only have one letter (the same for words with only one vertical or only one horizontal letter) it may be that the two rules of the previous sentence

are combined and that way we even get a case where one letter is allowed that none of its split-up letters actually shows up. We introduce the phrases *"full-split case"*, *"semi-split case"*, *"single-split case"* and *"zero-split case"* for the situations where a letter has precisely two, less than two, exactly one, or none of its split-up letters, respectively.

(ii) The purpose of this paragraph is to make one of the aspects of (i) more precise: Assume that A and B are two neighbouring letters of ω_i, and that the corresponding split-up letters of A, B are $a_1\&a_2$ and $b_1\&b_2$, respectively. Taking into account that by (i) a_1 has to come before a_2, and b_1 before b_2, there remain $\binom{4}{2}$ possible orders of those four letters within ω_{i+1}. The following words of type "$\ldots a_? \ldots b_? \ldots$" are devoted to describe these orders of our split-up-letters of consideration. These words are not to be understood as subwords of ω_{i+1}, moreover the corresponding words are derived from ω_{i+1} by ignoring all except the four split-up-letters of consideration. If understood that way, that words $a_1a_2b_1b_2$, $a_1b_1a_2b_2$, $a_1b_1b_2a_2$, $b_1a_1a_2b_2$, $b_1a_1b_2a_2$ and $b_1b_2a_1a_2$ describe the six orders that are consistent with rules of the sentence before the previous one. Rule 1.7(i) only forbids the sixth of these orders; however, in addition we note that for non-commuting A and B the first of these orders must be given.

1.9 Remark on the relation of word- and triple sequences:
Note that in many cases the sequences $(u_i)_{i\in\mathbb{N}}$ and $(v_i)_{i\in\mathbb{N}}$ of a triple sequence $(u_i, \omega_i, v_i)_{i\in\mathbb{N}}$ can be uniquely determined from $(\omega_i)_{i\in\mathbb{N}}$ alone. For, whenever a word ω_i contains vertical and horizontal letters, the computations of 1.6 clearly give the lattice square that contains the segment of our path between acts of crossing corridors that correspond to adjacent letters of different basic directions. Then, as in 1.6, all other lattice squares are uniquely determined, also, and hence u_i and v_i follow. Since this can be considered as a standard-situation, in many parts of this paper it will be sufficient to consider the "word sequence" $(\omega_i)_{i\in\mathbb{N}}$ instead of the entire triple sequence $(u_i, \omega_i, v_i)_{i\in\mathbb{N}}$. However, there are exceptions: A constant path can always be described by the empty word sequence, but an empty word sequence cannot determine the point taken on by a constant path. In this case we need the full triple sequence to determine this point, and this works as follows: Since all ω_i are empty, we have that $u_i = v_i$ for all i, and hence these sequences determine the one point taken on by our constant path. Similarly note that at least it needs a lucky geometry to determine for an entirely vertical non-constant path the corresponding x-coordinate just by giving the data of the horizontal coordinates to be crossed, similarly for horizontal paths.
Despite of the phenomena described in the previous sentences, for the sake of keeping our notation simple we will restrict most of the corresponding considerations to the sequence of middle entries of a triple sequence and we will call them word sequences. When doing this we will usually not exempt the special cases as described above, but we will include them treating the corresponding constant coordinate(s) as if defined by some other means. More precisely speaking, a sequence $(\omega_i)_{i\in\mathbb{N}}$ is called *"word sequence"*, if there exist sequences of lattice squares $(u_i)_{i\in\mathbb{N}}$ and $(v_i)_{i\in\mathbb{N}}$ such that $(u_i, \omega_i, v_i)_{i\in\mathbb{N}}$ is a triple sequence. In this context we let $\overleftarrow{\omega}_i := u_i$ and $\overrightarrow{\omega}_i := v_i$, i.e. we introduce

the reversed and the ordinary arrow-accent as a notation that shall help to relate the corresponding first and last entries of a triple sequence to their middle entries. We will use this notation regardless whether $\overleftarrow{\omega}_i$ and $\overrightarrow{\omega}_i$ are uniquely determined by their $(\omega_i)_{i\in\mathbb{N}}$-sequence or whether they are not. If $(\omega_i)_{i\in\mathbb{N}}$ is abbreviated by ω, then we also let $\overleftarrow{\omega}:=(\overleftarrow{\omega}_i)_{i\in\mathbb{N}}$ and $\overrightarrow{\omega}:=(\overrightarrow{\omega}_i)_{i\in\mathbb{N}}$.

1.10 Remark: *We should note at this place that we can only see overlapping braces as described in 1.7, if the variables that are attached to these braces are of commuting type.* We did not mention this fact in 1.5, since it is not a demand of the definition of a triple sequence, but an easy corollary that can be proved from other demands, namely as follows:
Assume that the corresponding braces are spanned between the level indices i and $i+1$. Denote by A and B the two $\overline{\mathcal{A}_i}$-letters that occur as an annotation of our braces, and denote by a_1, a_2, b_1, b_2 the corresponding split-up letters. In each of the five permitted orders as listed in 1.8(ii) there exists some position between the letters of ω_{i+1} so that a_1 and b_1 can be found to the left of that place, and that a_2 and b_2 are to the right of it. If we apply the computations as required in 1.6 to that situation, this brings us to the middle square at which the lines of corridors $\mathbb{L}(A)$ and $\mathbb{L}(B)$ intersect each other (cf. 1.12). If these lines intersected each other in hole, this would be a contradiction to 1.6. Hence we know $\mathbb{L}(A)$ and $\mathbb{L}(B)$ to intersect in a crossing region of corridors, and this implies by B.1.8 that A and B must be commuting variables.
Similar arguments hold in case of two intersecting braces both of which span ghost-letters. In that case we can conclude that the start- or the end-square of our path (i.e. $\overleftarrow{\omega}_{i+1}$ or $\overrightarrow{\omega}_{i+1}$ must be the middle square of a crossing region of corridors).

1.11 Remark: Recall that in 1.7 we introduced a "set of possible readings of ω_{i+1}", i.e. we offered a process that for the $(i+1)^{\text{st}}$ entry of a word sequence generates a set of possible predecessors in the word sequence, and we required in 1.7 that the actual i^{th} entry in our word sequence was one of these predecessors. If each of the sets of possible readings contained only one element, then the readings process could have been defined as mappings $\overline{\mathcal{A}^*_{i+1}} \to \overline{\mathcal{A}^*_i}$, and then word sequences could have more easily been defined as some kind of inverse limits. However, they do not, and this is essentially due to the same phenomena that we already described in B.1.4, 1.3 and A.4.9 for the situation where word sequences were to be read from paths. In spite of that, even with our description of word sequences as given in 1.4/1.5, the reading process may still be regarded as a multivalued mapping and hence word sequences may be understood as some kind of generalized inverse limit (cf. B.1.9).

1.12 Definition: Let C be a corridor. Then $\mathbb{L}(C)$ denotes the line of corridors (cf. B.1.2(ii)) to which C belongs. I.e. in this paper "$\mathbb{L}$" is used to denote an operator that is defined on $\mathcal{N}$ and that with each corridor associates its line. By B.1.3 this operator can also be defined on $\mathcal{A}$.

§2. Reading word sequences from paths

Recall that in B.1.4 we characterized the demands for how to read a word sequence from a path. However (cf. B.1.5) it is not evident that if one adjusts reading domains level by level and has arrived on the ith level, adjustments on that level can be made so that the adjustments on the levels $1, 2, 3, \ldots, i-1$ can be left unchanged. The purpose of the first three steps of this section is prove the existence of a reading by giving a strategy ("algorithm") of partially giving away the freedom described in 1.3 is such a way that one avoids running into such troubles as explained in the preceding sentence.

2.1 Lemma: *From any continuous path in X a triple sequence can be read in such a way that it satisfies the demands of a reading* (and hence the conditions 1.5(i)–1.7(iii)), cf. 1.4.

The Proof of this claim in principle rests on giving an algorithmic method for how to read a triple sequence $(\overleftarrow{\omega}, \omega, \overrightarrow{\omega})$ from a given path $u : [0,1] \to X$. The algorithm is described below in 2.2, the proof then follows in 2.3.

2.2 An Algorithm for reading a (valid) triple sequence from a given path:

(i) The first step of this algorithm is to fix the sequences $(\overleftarrow{\omega}_i)_{i\in\mathbb{N}}$ and $(\overrightarrow{\omega}_i)_{i\in\mathbb{N}}$. In principle each of these entries can be any lattice square of the corresponding level index that contains the point $u(0)$ and $u(1)$, respectively, such that the choice of the lattice squares $\overleftarrow{\omega}_i$ and $\overrightarrow{\omega}_i$ does not contradict to 1.5(i) with respect to $\overleftarrow{\omega}_{i-1}$ and $\overrightarrow{\omega}_{i-1}$ as chosen by the previous step. There is one situation where we could get such a contradiction, namely if $u(0)$ lies in the outer split-up corridor of an adjacent corridor C of the extended lattice square $Q := \overleftarrow{\omega}_i$. Then, $\overleftarrow{\omega}_{i+1}$ in principle could be adjusted to one of the lattice squares that on the $(i+1)^{\text{st}}$ level is found inside C, or to one of the $\mathcal{N}_{i+1}$-lattice squares that lies outside Q. However, with $\overleftarrow{\omega}_i$ chosen as above, that latter possibility would contradict to 1.5(i). On the other hand note, that no matter which of the possibilities had been chosen for $\overleftarrow{\omega}_i$, at least one possibility for $\overleftarrow{\omega}_{i+1}$-can be found that is compatible with 1.5(i).

Observe that this algorithm that is given as proof of Lemma 2.1 is based on not using the full freedom as explained in 1.3 in order to avoid running into troubles. However, we should note at this place that in the above step (i) it was not necessary to restrict the freedom of reading. *I.e. what is given below can be interpreted as an algorithm that also allows to read a valid word sequence ω, even if $\overleftarrow{\omega}$ and $\overleftarrow{\omega}$ are already given.*

(ii) Whenever our path w crosses a corridor, we associate a certain subsegment of our path with this act of crossing, and choose this subsegment in a minimal way (cf. 1.3(i), i.e. if $w_{\restriction[t_1,t_2]}$ is the corresponding segment, then $w(t_1)$ and $w(t_2)$ are the only two points that lie on the boundary of the corresponding corridor). This condition will turn out to be sufficient for getting a valid reading, but it is not necessary (cf. 2.4).

(iii) The intervals as marked in Step (ii) ("reading domains") are according to the conventions A.4.6(i)/B.1.4 associated with letters, and by putting those letters that correspond to the fixed lattice $\mathcal{N}_i$ in an order that corresponds to the order in which

the intervals appear in the parameter domain of u, we get the word ω_i for our reading. In this context "corresponds" is defined according to some *"central point"* which has to be chosen besides the start- and the endpoint of the reading domain that are used so far. The central point must be a parameter value of some point where our path takes on a value in the interior of the gap line that on the $(i+1)^{\text{st}}$ level (i.e. the next finer level with to those where our reading takes place) crops up in the middle third of the corresponding line of corridors. Apart from that the central points of the same level are chosen in such a way that they never coincide. Both conditions can clearly be fulfilled, in particular since a path when crossing a corridor has also to cross the middle third. Then the order of the corresponding central points on the parameter domain of our path u determines the order of letters.

(iv) Assume that at the beginning or at the end of our word our path is inside an $\mathcal{N}_i$-corridor. According to 1.3(iii) we are given the choice to treat the corresponding intervals as a reading domain or to ignore them for our reading purposes. In this case have to use take care of what has been adjusted as $\overleftarrow{\omega}_i$ or as $\overrightarrow{\omega}_i$, respectively. It can just be one of the adjacent lattice squares of the corresponding corridor. Taking up the corresponding reading domain (or letter) will in the sense of the 1.6-computations of the internal compatibility "bring" us to the other side of our corridor, or we will "stay" where are. Hence one of the possibilities will be compatible with 1.6, the other will not. We have to choose the compatible option.

Some other point must be noted in this exceptional situation: Other than required in (iii) the central point for such a reading domain that corresponds to an uncompleted act of crossing can not always be chosen in the central gap line. If it can, we will do this, but the uncompleted act of crossing a corridor might be so short that our path does not reach the corresponding gap line, but already ends while only having been inside the first split-up corridor. In that case we will define the corresponding central point to be endpoint of our path. Further note that in this exceptional situation we could also have coinciding central points (if a path ends inside an outer square of a crossing region of corridors). In the sense of (iii) the order of such letters is then arbitrary. Finally note that this convention of adjusting the central point of a reading domain is also well-defined in the zero-split case (cf. 1.8(i)): In that case we are given a path which entirely remains in the interior of a corridor, but which according to the adjustments of the first and last entry of the corresponding triple sequence is be be treated like a path that performs an (uncompleted) act of crossing. Due to that adjustment the corresponding split-up lines can clearly be associated with the attributes "first" and "second" (note that is adjustment is independent from what moves the path actually performs inside the corridor, it might even see the "second" split-up corridor before the "first" one!). For the purposes of defining the central point, there are just three possibilities for what our path could have done in the interior of its corridor: It either could have entered the middle gap line, or it could have stayed in one the two split-up lines. In the first case the central point is to be adjusted according to the convention (iii), in the second case our path has to be treated like a path that during its attempt to cross this corridor either has never got

further than the first split-up corridor (and hence the central point is the endpoint), or it is to be regarded as a path which has started its act of crossing so late that it never has seen anything but the second split-up corridor (in that case the central point is to be adjusted at the beginning). Hence the above convention to adjust central points sometimes at one of the ends of the parameter domain is even not contradictory in that case, where a path has both endpoints in the same corridor.

(v) Steps (ii)–(iv) are to be performed for each level index i. Note that the order-condition of (iii) and the condition to choose the intervals minimal as in (ii) are more restrictive than required by a general reading (cf. 1.3/B.1.4). However, by sticking to these rules we can for each level index i proceed independently without having to compare the word we are putting down with what has been read on the previous level or with what will be read on the subsequent one (cf. 2.4).

It remains to be proven that the above algorithm generates in each case a valid triple sequence. This will be done in the following by separately referring to the defining properties of triple sequences as they have been listed in 1.5:

2.3 Proof of 2.1:

Ad 1.5(i): This property is given since our algorithm has taken care of it in Step 2.2(i).

Ad 1.5(ii): This property is given since 1.6 is nothing but a listing of geometrical conditions where a path could go if on the one hand it has to avoid the holes that are integrated into $\mathcal{N}_i$ and on the other hand each act of crossing of an $\mathcal{N}_i$-corridor is to be reported by some letter. Since our path u is bound to these geometric conditions, the words ω_i as read from u have to satisfy the internal compatibility 1.6.

Ad 1.5(iii): In principle this property also is nothing but a rephrasing of geometrical conditions into combinatorial language: Since by B.1.1/B.1.2(vi) each $\mathcal{N}_i$-corridor has two split-up lines of corridors whose outer boundaries reuse the boundaries of the $\mathcal{N}_i$-corridor, one can compute from the acts of corridors of $\mathcal{N}_{i+1}$-corridors whether the $\mathcal{N}_i$-corridors had or had not to be also crossed. And, in principle 1.7(i) just requires that (up to a precise order in case our path goes through a crossing region of corridors), ω_i has to be the record of that $\mathcal{N}_i$-crossings that have to be performed by any path that performs all the given $\mathcal{N}_{i+1}$-crossings. And this also hold for those act of crossings that could be read from u. Thus, all it takes below is to make sure that our order-convention 2.2(iii) and our not caring about the adjacent word according to 2.2(v) cannot lead to orders of letters that contradict to 1.7(i).

By construction funny incompatible orders of letters can only occur if our paths goes through a crossing region of corridors; so let us consider this situation and let us for a moment postpone the exceptional situation of 2.2(iv). In that case it is not hard to see that a total reversion of the orders of letters is ruled out by our reading convention: If according to their order of central points we have put on the ith level the letter A before B, by geometry the central point of the first split-up letter of A must still come before the A-point, the central point of the second split-up letter of B must come after the corresponding central B-point. And this suffices to rule out the total reversion of the order of split-up letters, which according to 1.8(ii) is the only forbidden case. In the 2.2(iv)-case on the one hand similar arguments can be applied, and on the other hand

the following observation is valuable: If the central point of some letter on the ith level has been adjusted according to the convention 2.2(iv), then it is either not a split-up letter, or it is the first split-up letter of some root letter that does not have a second one. However, in the letter case the central point of the corresponding root-letter has been adjusted according to the same convention. Based on these observations a forbidden reversion of orders of the corresponding split-up letters can be ruled out in the semi-split-up cases, also (by the way, such a forbidden order does only exist, of at least one of the involved root-letters has its full set of split-up letters).
Hence we find all conditions for a word sequence satisfied, when we "read" it from a given path according to the conditions above. □

Recall that in B.1.5 and in the introductory remark of this section we indicated the possibility for "non-extendable readings". The following two paragraphs are devoted firstly to analyzing which demands cannot lead to such non-extendable readings and, secondly, to analyzing the remaining single situation that can:

2.4 Analysis of situations that could lead to non-extendable readings:
The phrase "non-extendable reading" has been introduced for a situation where an attempt to adjust reading domains in the sense of B.1.4 gets stuck by finding out that there does not exist any word in $\overline{\mathcal{A}}_{i+1}$ that describes our path and extends the initial finite subsequence $(\omega_j)_{1\leq j\leq i}$ as read before. If we experience such a phenomenon of reading a non-extendible initial word, since 1.5 contained essentially just an encoding of geometrical conditions into combinatorial ones, the following analysis will give that there is a single phenomenon, that can give non-extendible readings:

(1) By 1.7 non-extendible readings can only arise, if according to a certain adjustment that has been made for the ith level we cannot get around to acknowledge on the $(i+1)^{\text{st}}$ level certain acts of crossings that have been not been acknowledged on the ith level, or vice versa, or if we do not see enough freedom to arrange the letters that report the crossings on the $(i+1)^{\text{st}}$ level in an order that is compatible with what we had adjusted on the ith level.

(2) A closer look gives, that such an obstruction can never arise from acts of crossings that on the ith level have (not) been reported because they had been regarded as not completed: If this has happened, then either just one of the two split-up corridors had been crossed, or one of the two split-up corridors has seen some act of crossing that can be regarded as not completed itself. In any case the rules of 1.5 are generous enough to put down a record on the $(i+1)^{\text{st}}$ level that is compatible with what has been put down on the ith level.

(3) Further, the order-condition can never give an obstruction, also: If our path has on the ith level gone through a crossing region of corridors and the corresponding letters had been put into some fancy order, then on the $(i+1)^{\text{st}}$ level our paths must either go through the crossing region of two of the split-up corridors (then these letters can be put into the same order and this suffices to match 1.8(ii)), or it marches in such a way through the five non-corner regions of the (3×3)-subdivided crossing region, that the four split-up letters are coming in such an order that it can be told from that

order that on the ith level our path had actually been in that crossing region. In such a case we see overlapping braces in 1.7(i) and hence any order of letters on the ith level can be regarded as compatible.

(4) Further, if it has been decided to acknowledge an act of crossing on the ith level (we might have been forced to do that or we might have decided to interpret a tangency that way), then our path must on the $(i+1)^{\text{st}}$ level behave in such a way that again the crossing of both split-up corridors could have been acknowledged.

Hence there remains a single phenomenon which could lead to a non-extendible reading, namely if we decided not to acknowledge an act of crossing since it just ended with some tangency (cf. 1.3(i)). In the following we take a closer look at such a situation:

2.5 Example: Think of a vertical corridor of $\mathcal{N}_i$ and of a segment of a path approaching the corridor from the left in its upper third, continuing till it reaches the right boundary, then turning and using precisely the right boundary line to proceed downwards, and when the path has reached the lower third of our corridor, it turns to the left and sticks to the direction until it disappears from the region of consideration. By 1.3(i) we are given the choice of either regarding this as two inverse crossings of the same corridor of the ith lattice, or as regarding this corridor not as crossed at all. Imagine we went for the latter possibility. Further image, that on the $(i+1)^{\text{st}}$ level, when the region of our corridor is (3×3)-subdivided, on the middle row two holes are detected, such that the left and the right square of the middle row have become holes, and hence our corridor of $\mathcal{N}_i$ is split up into four new vertical corridors which take up the upper left, upper right, lower left and lower right of the old $\mathcal{N}_i$-corridor. When now reading the $(i+1)^{\text{st}}$ segment from our path, we cannot avoid regarding each of the four corridors as being crossed: Since our path on its way from the left to the right uses the regions of the upper corridors and on its way from the right to the left it uses the regions of the lower corridors, there is not any chance of disregarding any of these crossings, and we will not get cancellable letters in our word. Obtaining such a substring of our $(i+1)^{\text{st}}$ word would imply by 1.7(i) that on the ith level we would have to regard our corridor as being crossed. However, on the ith level it had appeared that we were given the choice of disregarding these crossings.

There is nothing wrong with what we have said so far, we just have to take into account that careless reading might lead to the phenomenon that words read before may have to be corrected later. This, of course, causes a lot of algorithmic trouble (Can one control the number of such rewriting processes? — Do they stabilize in any case?). In 2.2 we already indicated a method for how to avoid this trouble (to think of the boundary lines of corridors as not belonging to the corridors, similarly to think of extended lattice squares as open domains with their outer boundaries not belonging to them). However, we are now able to precisely understand the geometric origin of that trouble: On the one hand we need to regard all domains as closed, if we want to define word sequences that are consistent with skipping to limits (cf. 1.3/3.8). On the other hand we need to be consistent with respect to the split-up structure if we do not want to run into non-extendible word sequences. In the above example the medium third of the right boundary of the corridor was not contained in the corridors or in its closures of the $(i+1)^{\text{st}}$ level, but

we had regarded it as a part of the closure of the $\mathcal{N}_i$-corridor. Consequently we can conclude that we will not run into any such trouble if we define our corridors as closed domains, namely as such domains which consist of the interior domains and of those boundary points that can be approximated from the interior points that actually belong to X. Hence we should remove those boundary points from corridors (or from the outer boundaries of extended lattice squares) that are the outer boundaries of holes adjacent to the boundaries. Consequently we would not run into non-extendible word sequences if we modified 1.3(i)/B.1.4 in such a way, that one is not given the freedom of not acknowledging an act of crossing a corridor if the points on the opposite boundaries belong to outer boundary points of holes of any higher indexed lattice. Since we have to admit that that this information is not available on an algorithmic basis, we will not proceed as in the above conjunctive-sentences. In addition, having by 2.2 at least one algorithmic method that produces a valid reading turns out to be sufficient for our purposes.

2.6 Analysis *in how far the same word sequence can be read from a path, even if the reading domains have been adjusted differently:*
The purpose of this remark is to explain that in all standard situations it cannot, but that there is one (and only one) exceptional situation in which this can happen. The degrees of freedom in our reading process have been described in B.1.4, and they essentially go back to 1.3. Hence we proceed by looking at the facilities of 1.3(i)–(iii) separately:

Ad 1.3(ii): This degree of freedom is irrelevant in this context since, even if 1.3(ii) to a certain extend allows us to vary the order of letters, 1.3(ii) does not allow us to adjust the corresponding boundaries of the overlapping domains in different ways.

Ad 1.3(iii): Whenever we use the degree of freedom as provided by 1.3(iii), i.e. to or not to treat a partial act of crossing like a completed one, we have to put or not to put the corresponding letter into our word sequence. In addition, there can only be one segment for each of the basic directions at the beginning and at the end of our paths, to which 1.3(iii) can be applied. Hence 1.3(iii) can never be the origin for reading the same word sequence in different ways (at least as long as word sequences are, as they should, be treated as abbreviations of triple sequences, cf. 1.9/D.7.16/D.7.17).

Ad 1.3(i): In the standard situation the freedom to or not to ignore a tangency will or will not give an additional cancelling pair of letters, respectively, and hence, as in "Ad 1.3(iii)", it will give rise to different word sequences. However, another situation would be given, if the segment of our path on one side of the tangency performs a completed act of crossing, but on the other side of the tangency does just go a bit into the corridor and then leaves it by crossing the same boundary line that had contained the tangency, already. Denote by a, b and c the parameter values of the start point, the tangency-point and the final exit point of our path, respectively, and let C be the corresponding corridor. In that case 1.3(i) gives us the freedom to adjust the reading domain for our act of crossing the corridor in different ways: one possibility is $[a, b]$, and the other is $[a, c]$. On the corresponding level on which we see this picture, we do not get a different reading for our different adjustments. Do we on the finer levels? – In the standard situation, as described before, we do! – For, in the standard situation our path will have on $[b, c]$ be gone deep enough into C,

so that it has performed at least two reverse acts of crossing of the iterated split-up corridors of C. If we decided to use $[a,b]$ as reading domain, then b has also to be the endpoint of the act of crossing as performed to the iterated split-up corridor on $[a,b]$, and hence the two acts of crossing that, as discussed before, take place $[b,c]$ have also to be acknowledged by letters. On the other hand, if we treat $[a,c]$ as a reading domain, the acts of crossing as acknowledged by split-up letters must not end at b (cf. B.1.4(vii)). Hence we see fewer (cf. 2.8) split-up letters in our word sequences as if we would have treated b as an endpoint of our reading domain. That way we get different word sequences.

Of course, there is one exceptional situation, when the above arguments cannot be applied: This is, when on $[b,c]$ our path does not go back into C, but just runs along on the boundary of C for a while (more precisely: for the parameter values $[b,c]$). If in that situation the path is not passing along the outside boundary of one of the fine holes that are tangent to the outside boundary of C (cf. 1.4/2.5), we are free to adjust the endpoint of the reading domain for the $[a,b]$-crossing to any of the parameter values in $[b,c]$. This hence remains the only situation where different word sequences can be read from the same path. The following paragraph is devoted to take a closer look at this situation:

2.7 Example: Given the exceptional situation as described at the end of the preceding paragraph, we ask whether a different choice of this adjustment affects the readings of the finer levels? – Well, it affects at the degree of freedom that we have! – For simplicity, let us first assume that our path while on $[b,c]$, crosses some corridor C' that is perpendicular to C and belongs to the same level index; and if the parameter domain for such an act of crossing, still belongs to the reading domain of the act of crossing C, then we are free to put the corresponding letters in any order we like, but if not then we are not. The same argument holds for the finer levels of our word sequence, since by B.1.4(vii) the endpoints of the corresponding reading of the split-up corridors of C must coincide with the endpoint of the reading domain of the main corridor C itself. Note that on a sufficient fine level there have to exist corridors are crossed by the non-constant segments of our path. However, what has been expressed by the preceding three sentences is an "If"-conclusion, there is non "only if"-part! – We might have perfectly decided to adjust the reading domain of the crossing of our main corridor C to $[a,c]$ so that we enjoy the full freedom of putting the letters that correspond to the corridors that are crossed on $[b,c]$ in an arbitrary order with respect to the split-up letters of C. Hence in particular we can choose the same order that would have to be used if we had adjusted our reading domain to $[a,b]$. In that case we get, despite having had different reading domains, the same word sequences for the $[a,b]$-reading as for the $[a,c]$-reading. In addition, we see that, wherever we have such an interval like $[b,c]$ so that any point $b' \in [b,c]$ might be used as the endpoint of the reading domain of the act of crossing C, then, if one is given a particular word sequence that could be read from our path, a sufficiently small b' might not be a possible adjustment of the corresponding reading domain, but the sufficiently big b'-values will be. Hence there is a minimal b' for which $[a,b']$ is a valid adjustment of the reading domain of our C-crossing. *That way*

we can find for any given reading of a word sequence from a path a "normed minimal adjusting" of the reading domains, which will have to be used at certain occasions within this paper.

2.8 Postscript to Analysis 2.6:
When reviewing the arguments of 2.6, some of them might not seem completely rigorous. In particular this holds for the place where we handled an (ordinary) tangency case by arguing that it depends on our reading adjustment at the point of tangency, how many letters we see in on a certain split-up level. We proved that a decision to skip to acknowledging such a tangent point as the endpoint of a reading domain would result in getting "more" letters on a certain split-up level. Hence one could ask whether such an effect could not be undone by simultaneously disregarding another tangency. The following paragraph is devoted to explain why even in such a situation we still must obtain a different word sequence:
According to the arguments as given in the main part of the above proof it should be clear that only, if a path performs several tangencies to the same corridor C while remaining all the time in that corridor, the above arguments might not be complete. Hence let us consider a path which does this. By standard continuity arguments, there can only be finitely many of them or, more precisely, finitely many zones of tangencies which lie alternately on the opposite boundary lines of our corridor. Such a subdivision of our path into segments of "zones of possible tangencies" and "zones where it runs from one zone of possible tangencies to opposite zone" might topologically be defined by adjusting an appropriate covering for the corridor by three open sets, two sets being a neighbourhood of the corresponding the boundary line (these two sets being disjoint from each other) and the third set being a cover for the remains of the corridor with some small overlaps to each of the other sets. Then one might look at the path-components of the complete preimages of these sets with respect to the parametrization mapping of our path, and one might use Heine-Borel property to find a finite minimal covering of the parameter domain $[0,1]$. Such a covering clearly defines "zones of possible tangencies". These zones of tangencies can also be clearly identified by looking at the finer levels of our word sequence. Note that 1.6 can be interpreted in such a way that each word of a word sequence defines a sequence of lattice squares through which our path passes. Another word sequence might give a sequence which partially contains adjacent lattice squares (as a result of additionally acknowledging tangencies or partial acts of crossings, or putting letters in different order); but due to different readings such processes can never be applied iteratively to the same place of a word (in the sense as it is stated in D.2.9 under similar but different assumptions). Hence different sequences of lattice squares can just differ by sometimes going over neighbouring ones. That way by any valid reading of our path on a sufficiently fine level one is able to trace the above zones of tangencies and to clearly identify them. One can also clearly trace the "changing sides"-zones where our path moves between adjacent zones of tangencies at opposite sides of our corridor boundaries. These changing-sides zones can be associated to certain letters from a fine split-up line of corridor that lies in the middle of our main corridor, let us use b as a preferred letter to denote a corridor from that line. If our path is going to

change sides, this may be acknowledged by some letter "b" in one reading of our path, and by a sequence of letters such as "$bb^{-1}bb^{-1}b$" in another reading (if there are a lot of tangencies to the boundaries of the corridor b), but the changing-sides zone can be identified in any case. We then look at the iterated left and iterated right split-up letters of our main corridor that belong to the same level as b, and let a and c be the preferred letters to denote corridors from those lines, respectively. Since a, b, c belong to parallel lines of corridors and are all non-commuting variables, the order of the a-, b- letters and c-letters is clearly defined, and thus each a- and c-letter is clearly associated to a zone of tangencies; i.e. to the interval that is bounded by some blocks of b-letters that mark a change-sides zone. And then we can tell by looking at the a- and c-letters whether some reading domain of some act of crossing our main corridor C does or does not end in a certain zone of tangencies. For each zone of tangencies we have, depending on what zone we are in, to ask whether we find a cancelling pair of type "$a^{-1}a$" or "cc^{-1}" (we said "type", because we have to look for all letters that belong to the same line of corridors as a or as c, respectively); and of course in the first and in the last zone we just might find a single "a", "c", "a^{-1}" or "c^{-1}", respectively. According to the conditions of 1.5/B.1.4, wherever there are neighbouring a- and c-letters in our word, an act of crossing the main corridor by our path must be acknowledged, even if the endpoints of the reading domains of the a- and c-letters are just tangencies. In that context "neighbouring" means that there are not any other letters from one of these lines of corridors in between. If in that sense "$a \dots c$" are neighbouring letters, then the start point of the reading domain of a must also be the start point of the reading domain of some letter that on the corresponding level reports an act of crossing the main corridor, and the endpoint of the reading domain of c must be the corresponding endpoint; similarly when we find neighbouring letters "$c^{-1} \dots a^{-1}$".
In the preceding paragraph we pointed out how we can clearly identify distinct zones of tangencies, and how to tell from the word sequence whether one of these zones contains the endpoint of the corresponding reading domain of the big corridor or whether it does not. Since this could be told from looking at the word sequence, this rules out that there exists paths that permit, by disregarding one tangency and acknowledging another, to read the same word sequence in different ways from them.

2.9 Remark: 2.8 must be understood as a general scheme of arguments that always can be used in a similar context. We had been driven to add postscript 2.8, since the original arguments of 2.6 had been based on pointing out that a certain act of skipping from acknowledging to disregarding a certain tangency always comes along with getting fewer letters in the corresponding word sequence, it might have seemed not entirely clear whether this effect could not have been neutralized by an inverse decision at some other point of tangency. By this postscript we made clear that the corresponding letters taken up or lost at some point of tangency can clearly be distinguished according to the position of their split-up letters on higher levels. Hence we provided an argument that rules out any such "neutralization" effect. This scheme of arguments that is based on going to sufficiently fine levels where the uncertainties of the less fine level have become clear (even if there may be new uncertainties, but they will only matter on the finer

levels) can be considered as rather standardized, and it can be used in many similar situations. Hence we will not hesitate to use the more-letter/fewer-letters arguments in forthcoming proofs, also, and we will then not add similar postscripts.

§3. Realizing word sequences by paths

Note that the following lemma gives the reversion to Lemma 2.1. By these lemmata the correspondence between word sequences and the variety of paths in our space of consideration is established via the reading- and realization mappings Φ and Ψ giving the backbone of this paper. This correspondence is not a bijection and hence Ψ can only be regarded as a multivalued mapping (cf. B.1.6(1)); in how far it differs from a bijection will also be discussed and understood in this paper (cf. D.12.3). However, the lemmata 2.1 and 3.1 already suffice to give Theorem C (as listed in the Introduction).

3.1 Lemma: *For any triple sequence that satisfies all rules of 1.5(i)–1.7(iii) there exists one path one of whose readings is the given triple sequence.*

Skeleton of the Proof: The main bulk of this proof has to be spend on constructing the corresponding path $u : [0,1] \to X$. For the that purpose a whole sequence of paths $(u_i)_{i\in\mathbb{N}}$ with $u_i : [0,1] \to X_i$ will be constructed, and then u will arise from passing to the limit. However, it will not be that easy to see that all desired properties sustain during the passage to the limit, and this is what makes the subsequent proof fairly lengthy so that all propositions of this section are devoted to it. In particular the proof will require the development of a concept of certain auxiliary diagrams "split-up diagrams", and that some properties of those graphs have to be shown. Hence it is subdivided into eleven steps, and we start by giving an overview over all those steps.

(1.) Definition of the approximating paths u_i (Def.3.2)

(2.) Proof of the uniform convergence of u_i in the special "non-reversion"-case (Prop.3.3).

(3.) Description of the associated split-up graphs and statement of a certain property whose proof is postponed (Def.3.4).

(4.) Proof that the u_i uniformly converge in the full-split case of the general case by using the result as stated in (3.) (Statement & Corl.3.5).

(5.) Proof that the limit-path $u := \lim_{i\to\infty}(u_i)$ is contained in X (Prop.3.6).

(6.) Proof, that the approximating paths u_j have the correct reading ω_i for all indices $j \leq i$ (Prop.3.7).

(7.) gives a proof that on each fixed level stable readings do also hold when passing to the limit (Prop.3.8).

(8.) Unfortunately the properties (3.) and (7.) will be not sufficient for the desired conclusion, since they do not give the compatibility of the bounding parameter values of the corresponding reading domains for different indices i within our system $(u_i)_{i\in\mathbb{N}}$. Hence this compatibility is deduced in 3.9 by using geometrical properties of the split-up diagrams.

(9.) Final step of the proof of 3.1 modulo the gap left in (3.) (Prop.3.11).

(10.) Closing the first gap that was left in (3.) (Prop.3.12).

(11.) Separate discussions of the semi-split cases in 3.13–3.17 and hence closing the second gap that was left in (3.).

3.2 Definition of the approximating paths for a given word sequence:
By using the notation of 3.1, u_i is constructed as a parametrized polygon where each segment is linearly parametrized. Hence, what needs to be determined in order to define u_i are on the one hand the "segmentation points" (i.e. those points of the plane where two adjacent segments of the polygon meet and u need not be differentiable) and the corresponding nodes, (i.e. the parameter values at which u_i shall take on these points). If both of these sets are constructed and properly indexed, then our path u_i is uniquely determined by the concept of being a piecewise linearly parametrized polygon. The corresponding rules for obtaining the segmentation points and corresponding nodes for u_i are as follows:
0 and 1 are "exceptional nodes" on each level i, and this in particular implies that they are neither treated as "essential" nor as "inessential" nodes as this will happen for most of the remaining nodes below. The segmentation points for these exceptional nodes are the mid-points of the pure lattice square $\overleftarrow{\omega}_i$ and the mid-point of $\vec{\omega}_i$. Apart from 0 and 1 there will be a second type of exceptional nodes; namely, if the following definitions will give that at two adjacent of the ordinary (essential or inessential) nodes the same value of the approximating path will hold, so that a mere interpolation would result in a path which has a constant segment that appears as tangency to a corridor boundary, then we define a new exceptional node at the arithmetic average between the two ordinary ones. At such an exceptional node our path is defined to take on the mid-point of the corresponding lattice square. None of our approximating paths will at any place be tangent to any corridor boundary of the corresponding lattice thanks to that construction. The fact that these nodes have to be regarded as exceptional implies that at a later state of the proof, when essential an inessential nodes will have to be counted, the nodes of this type are completely disregarded in the corresponding counting process.
For describing how to obtain the other types of nodes and segmentation points, we proceed by only looking at the level $i = 1$ in the following paragraph:
Each letter in ω_1 determines two segmentation points, namely on the opposite boundaries of the corresponding corridor to be crossed. If the corridor does not go over a crossing region with corridors of the same index, then the segmentation points are just the mid-points of the boundaries. If the corridor goes over crossing regions, we take the mid points of the relevant segment to be crossed, namely the segment which by $\overleftarrow{\omega}_1$, $\vec{\omega}_1$ and the letters of the other basic direction of ω_1 is the segment in which our act of crossing shall take place (cf. 1.6). This principle of determining the segmentation points holds for any index i. The corresponding nodes are determined as follows: If ω_1 has n letters, then the nodes are placed at $\frac{j}{2\cdot n+1}$ $(0 \leq j \leq 2\cdot n+1)$ i.e. the parameter domain $[0,1]$ is uniformly subdivided for placing the $2\cdot n$ non-exceptional nodes and the 2 exceptional ones at the parameter values 0 and 1. For bigger i the rules are a bit more complicated: At first we must then differentiate between essential nodes and inessential nodes. Essential nodes are precisely those contained in the following list:
(i) the first node of each first split-up letter,

(ii) the second node of each second split-up letter, and
(iii) For those split-up letters which do not have a split-up partner (cf. 1.7(ii)), both nodes are treated as essential.

Hence, apart from the zero-split case (cf. 1.8(i)), each node is precisely associated to one node of u_i by the relation imposed in 1.7(i): The entering of a corridor of the $(i-1)^{\text{st}}$ level must correspond on the ith level to the entering of the first of split-up corridors, and the exit from a corridor of the $(i-1)^{\text{st}}$ level must correspond on the ith level to the exit from the second split-up corridor. If there is just one split-up letter, there is a similar "natural correspondence" even if in that case the "corresponding" segmentation points are not on the same boundary line as in the standard case. We have called those nodes "essential" that are associated that way to some nodes on the $(i-1)^{\text{st}}$ level. However, observe that according to condition 1.7(i) they do not come necessarily in the same order as for u_{i-1}, hence note that there are two different bijections between the nodes for u_{i-1} and the essential nodes for u_i, namely one given by geometry, and one given by the natural order coming from the order of the letters within ω_i or ω_{i-1}. The case where these to orders do not agree will be called the *"reversion case"*. We now decide that with respect to the natural order of letters the jth essential node of u_i should be at the same position as the jth nonzero-split node of u_{i-1}, (in this context "non-zero split" means that if ω_{i-1} should have a letter that has zero split-up letters as permitted by 1.7(iii), then the two nodes that correspond to this letter have to be exempted when counting). The parameter values of the additional (inessential) nodes of u_i are then fixed by uniformly subdividing the segment between the adjacent essential nodes.
Having defined our paths u_i we now have to make sure that they converge. For that purpose we want to make sure that our sequence of paths satisfies a uniform Cauchy criterion.

3.3 Proposition : *Let u_i and u_{i+1} be the approximating paths for a given word sequence ω as constructed on the levels i and $i+1$ respectively according to 3.2. If between these construction processes reversion in the sense of 3.2 has not taken place, then we have that $\|u_i(t) - u_{i+1}(t)\| \leq 2 \cdot \sqrt{2} \cdot (\frac{1}{3})^i$.* (This estimate will guarantee the desired uniformity for 3.1(2.) in the non-reversion case.)

Proof: Let t_0 be the biggest node of u_i which is smaller than t, and let t_1 be the smallest node of u_i which is bigger that t. From our method of definition we see that the path u_i is for the entire segment $[t_0, t_1]$ in a correspondingly described region of $\mathcal{N}_i$. Further we get from the non-reversion assumption of this proposition that u_{i+1} enters and leaves the same region of $\mathcal{N}_i$ at the same parameter values as u_i. At the inessential nodes of u_{i+1} between t_0 and t_1 our path u_{i+1} may be forced to perform additional acts of crossing corridors, but usually these corridors crossings should just determine in which of the subregions of the corresponding region of $\mathcal{N}_i$ our path has to take on its values. In case this region is a pure lattice square, these additional corridor crossings may also force u_{i+1} to cross the inner split-up-corridors of the adjacent corridor-regions of $\mathcal{N}_i$; but by 1.7(i) u_{i+1} cannot be forced to cross the outer split-up corridors as well, since such an act would have to be reflected by the corresponding letters of ω_i, also.

Hence our segment of consideration of u_{i+1} has to be in the same or in at most in an neighbouring region of $\mathcal{N}_i$ as the corresponding segment of u_i. Since we know that each of these regions is a square with an edge-length $(\frac{1}{3})^i$ (cf. B.1.1), we hence get the estimate $2 \cdot \sqrt{2} \cdot (\frac{1}{3})^i$. In the single-split case (cf. 1.8(i)) by the above construction our path has started in the middle of some pure lattice square, and then the corresponding act of crossing (which has been acknowledged by a letter with just one split-up letter) has either been the first act of crossing one of the bounding corridors, or, by translating the conditions of 1.5/1.10 into geometry the second possibility can be described as follows: The path u_i could also have started by moving in one basic direction only, and then the crossing that it performs after its first turn into the other basic direction might be an act of crossing which is acknowledged by a letter with just one split-up letter. By 1.8(i) "one split-up letter only" means, that on the $(i+1)^{\text{st}}$ level instead of the crossing of the entire corridor just the crossing of the second of its split-up letters is acknowledged. This implies that u_{i+1} has not started in the same pure lattice square as u_i, but it could have only started in one of the lattice squares that on the $(i+1)^{\text{st}}$ level are defined in the regions of the bounding corridors of the u_i-start square (since this condition allows u_{i+1} to avoid crossing the corresponding first split-up corridors). Now, thanks to having understood the geometrical situation, we see that u_i is never further apart from u_i than having gone to the neighbouring $\mathcal{N}_i$-region, and hence the above estimate 3.3 holds in this case, also. The zero-split case can occur according to 1.7(iii) only, if the corresponding letter of ω_i is the only one with that basic direction. This means that u_i sees, apart from one exception, only crossings of corridors in the same basic direction. I.e. u_i on its whole way always runs parallel to some line of corridors, with one change to the other side at some intermediate state. With similar arguments as in the case of one split-up letter we conclude that u_{i+1} can only be located in the interior of this line of corridors. And hence, even in this case u_{i+1} is never further apart from u_i than by having been moved to the adjacent $\mathcal{N}_i$-region. Recall that in this case we had two u_i-nodes that have not been become essential for u_{i+1}. We now see that this does hardly matter: this phenomenon just causes that one of the u_{i+1}-node intervals is three times longer than the standard, but this is irrelevant for our estimate. □

By now all cases of the special non-reversion case have been discussed, so that we ready to continue our discussion by taking a look at the general case, where we do not longer want to rule out as possible reversion of order between the essential nodes of u_{i+1} and the nodes of u_i.

Recall that by 1.7(i) braces may overlap, but must not intersect each other. This condition puts a natural bound to what extend the order of split-up letters within ω_{i+1} might differ from the order of the corresponding root-letters within ω. In order to understand this bound, we develop a new type of auxiliary diagrams which are defined below:

3.4 Definition of *"split-up diagrams"*, (for a figure see 3.13(Fig.i)):
A split-up diagram is related to two adjacent words ω_i and ω_{i+1} of a word sequence ω. It consists of two horizontal bars the lower of which is annotated by attaching the

string of all essential letters of ω_{i+1} to it (cf. 3.2), and the upper bar is annotated by attaching a string of the same letters in the different order induced by ω_i to it. I.e. each letter of ω_i is to be replaced by its split-up letters, and hence the string of letters on the upper bar has essentially the order of ω_i. We then try to visualize the permutation that transforms the two strings of letters into each other by connecting the corresponding letters on the top and on the bottom bar by lines which only transversely intersect each other and which are monotonous with respect to the y-coordinate. By drawing some of the lines white and others black we make a distinction between lines that connect horizontal letters and those lines that connect vertical letters, respectively. Finally we note that at the top all letters come in pairs (that correspond to one letter of ω_i). We should now explain how to translate the situations of the cases 1.7(ii)/1.7(iii) into our split-up diagrams: If we have to deal with word sequences where braces as described in 1.7(ii)–(iii) occur, we put additional ghost letters at the beginning or at the end of the bottom bar, at most two (namely one of each colour) if accordingly needed. Those letters of ω_i which only have zero or one split-up letter are yet replaced by a pair of letters on the upper bar, and the corresponding split-up partners are then connected to the ghost letters. That way we get a picture where without exception each letter of ω_i is replaced by a pair in the annotation of the top bar. We now agree to hatch the area between the two lines emanating from the two partners of such a pair in the same colour (black or white) as the two lines.
By that principles of construction and the conditions of 1.7(i)–(ii) we obtain a geometric picture as described in Proposition 3.12. The conclusion of Proposition 3.12 will then be the key argument that enables us to give the proof that the u_i uniformly converge in the general case, also (cf. Proposition 3.3).

3.5 Statement of a Proposition and Corollary: By translating the conclusion of Proposition 3.12 (which is unfortunately only phrased so as to treat the full-split case) back into the language of nodes and essential nodes as above, we get the following result:

> *If the jth node of ω_i corresponds to the intersection with a certain boundary line, then the jth essential node of ω_{i+1} corresponds either to an intersection with the same line, or it corresponds to intersecting a line which can be derived from the first by removing from the list of horizontal lines to be intersected the last item and by putting one more item to the list of vertical lines to be intersected, or vice versa.*

Hence we get that the corresponding segments of u_i and u_{i+1} should either be in the same region of $\mathcal{N}_i$ or in a neighbouring (diagonally adjacent) region. If we then take in addition into account that one of our two neighbouring regions may be a pure lattice square so that the corresponding path may have entered the adjacent region without acknowledgement by the corresponding word sequence, we get that instead of "$2\cdot\sqrt{2}\cdot(\frac{1}{3})^i$" as in 3.3, in this case we can only obtain the estimate

$$3\cdot\sqrt{2}\cdot(\frac{1}{3})^i \tag{1}$$

However, this estimate is sufficient to conclude that the u_i are a uniformly convergent sequence. Hence there exists a limit-path which we will call "u" — at least in the full-split case where all the ω_i are free of letters that have zero or one split-up letters, only. Despite that gap the subsequent steps will also be relevant in these semi-split cases, cf. 3.13.

3.6 Proposition: *The assumptions of 3.5 imply that when using the notation of 3.1 the limit path u only takes on values inside X.*

Proof: If u would fail to satisfy the conclusion, by A.3.3 it would have to take on values inside some hole, since it is a well-defined path in $\mathbb{R}^2$. The corresponding hole is according to its finite index i integrated into the approximation space X_i (cf. A.3.1/ A.3.4) so that by construction neither u_i nor any approximating path with a bigger index can take on a value inside that hole, and, since X_i is closed, neither can the limit u. □

Hence one item remains to be proved in order to obtain 3.1, and this is that ω, i.e. our word sequence as originally given, is one of the possible readings of u. However, it will take more then one step to achieve this result. The first of them is:

3.7 Proposition: *The assumptions of 3.5 imply that when using the notation of 3.1 each of the approximating paths u_i has the desired reading ω_i for each index $j \leq i$.*

Proof: By its construction in Definition 3.2 the path u_i has the desired reading for the level index i. The fact that it has also the desired reading for each smaller level index is straightforward from the compatibility of a word sequence (cf. 1.7). Recall that these conditions have been just set up in such a way that they require that ω_{i-1} lists those crossings of $\mathcal{N}_{i-1}$ that any path has to perform that performs in $\mathcal{N}_i$ the crossings described by ω_i. For a more detailed exposition of the corresponding arguments see 2.3(Ad 1.5(iii)). In that context observe that our approximating path has in 3.2 been constructed in such a way that they do not perform tangencies. This eases the adaptation of the arguments of 2.3(Ad 1.5(iii)) to our situation here. □

3.8 Proposition: *The assumptions of 3.5 imply that when using the notation of 3.1 u has the desired reading ω.*

Proof: We fix a certain level index j and restrict our consideration to the lattice $\mathcal{N}_j$ aiming at proving that u has at least the desired reading on our level index of consideration. Since j can be chosen arbitrarily, this set-up is sufficient for our purposes. It also suffices to consider the subsequence $(u_i)_{i \geq j}$ which by Proposition 3.5 uniformly converges and by Proposition 3.7 all have the desired reading. On the other hand our conventions for a reading (cf. 1.3/B.1.4) have been set up in a sufficiently generous way so that, when passing to a limit, we should not be forced to read our words from the limit path in an essentially different way as they could be read from each of the approximating paths. In particular, if all approximating paths have a certain segment in the interior of a corridor, but the limit path does touch the boundary, by 1.3(i)/B.1.4(i)–(iv) this

does not force us to adjust the endpoint of the act of corridor crossing to another endpoint than the corresponding endpoints of the approximating path (the bad case 2.5 is ruled out in that context). Similarly, if all approximating paths go through a crossing region of corridors, but if the overlap-domains of the two simultaneously performed acts of crossings get smaller and smaller for $i \to \infty$, by 1.3(iii)/B.1.4(x) we still enjoy the freedom to put the corresponding variables in any order, even if the overlap-domain should be degenerated to a single point. However, despite having recalled this, we are not yet happy with our situation in the course of proving 3.1, since up to the moment we do not have any control how the bounding coordinates of the parameter domains where the different u_i cross the same corridors behave when we consider $i \to \infty$. By construction we cannot assume that they are constant. At this point we could continue our proof by using the compactness of the parameter domain [0,1] and some skipping to such a subsequence where the corresponding parameter domains converge (note that such a proof could not come out straightforward, since we have to cope with a permanently increasing number of nodes when we come to considering bigger and bigger level indices; hence one would have to play the diagonal-sequence trick to make that scheme of proof work). However, for the sake of achieving a complete understanding of the situation, we prefer to avoid skipping to such an artificial subsequence; instead of this we want to proceed by an an extra lengthy proposition where we want to prove that these parameter values even converge in the original sequence. The proof of this proposition will then be completed in 3.10.

3.9 Proposition: *The assumptions of 3.5 imply that when using the notation of 3.1 the bounding parameter values of the reading domains that are related to each other in the sense of 3.2(i)–(iii) do converge for the approximating paths u_i of u.*

Proof: In order to obtain convergence for the bounding parameter values of the reading domains we look at the split-up diagrams as constructed in 3.4. From considering one of the black or white lines on their way from the top-bar to the bottom bar and from counting the numbers and directions of the intersections with other lines, we can compute the difference between the starting position at the top bar and the end position at the bottom bar. By using that picture we want to discuss how many steps to the left or to the right one of our lines can take on its way from the top to the bottom. In that context, the following observation is of importance: For a left boundary line there is hardly any obstruction to take steps to the left, (i.e. if the corresponding line is white and all the adjacent lines starting on the left hand side are black, then our left boundary can cross all of them), but on the other hand there is a bound for a left boundary line to take steps to the right. It can cross at most one other black line, because at the moment it would cross two of them, the corresponding white corridor could not avoid to completely intersecting a black corridor. For the right boundary lines there is a similar result telling us that they can do at most one step to the left. Recall that when parametrizing u_i, the endpoints of the boundary lines of the corridors in these braid-like diagrams correspond to the parameter values of fixed nodes, and hence by recalling the procedure to define these nodes, the one-step condition gives a concrete estimate for how far the start point

of a reading domain can be moved to the right when comparing the situation for u_{i+1} with that of u_i, and how far the endpoint of a crossing domain can be moved to the left, when comparing the same two paths. For $i = 1$ the estimate for the "length of a step" is 1/3, since it takes at least two corridors (one black and one white) to perform such a crossing, and, even if we take into account that the start point of the first and the endpoint of the last corridor might be at exceptional nodes, i.e. at the parameter values zero and one, this subdivides our unit interval into three subintervals. If our corridor should perform a similar step on the next level, the corresponding subinterval of length 1/3 would have to be again subdivided, since the outgrow of each boundary of one level are two boundaries below, and so we again get a subdivision into at least three intervals. Hence it follows that there is an estimate for the length of a step to the left, namely by $(\frac{1}{3})^2$ for the second step, and by $(\frac{1}{3})^i$ for the ith time such a step should be taken to the left. Now assume that a_i is the parameter value one of the nodes of u_i, i.e. some parameter value at which u_i enters a particular corridor C or exits from it. Denote by a_j for $j > i$ the corresponding entrance or exit-points of u_j, i.e. the nodes that correspond to the entrance point of the iterated left split-up letters of C, or to the exit points from the iterated right split-up letters from C. Since we are postponing the semi-split cases to 3.13–3.17 and to the following paragraph, we may assume that all these split-up letters actually show up in the corresponding word of ω. The purpose of this Proposition 3.9 is to prove that $(a_j)_{j\geq i}$ converges. From the above arguments we get the information that, if the a_j had been parametrizing an entrance point, then the sum over the positive differences $(a_{j-1} - a_j)$ is bounded by a convergent sequence. This is sufficient to conclude that $(a_j)_{j\in\mathbb{N}}$ must be a convergent sequence. Namely, since this sequence is defined as a sequence of reals in a bounded interval, this sequence could only be non-convergent if it had more than one accumulation point. But in that case the sequence would have to take on infinitely many values in disjoint neighbourhoods of its accumulation points, and hence for taking infinitely many journeys between these neighbourhoods in both directions neither the the sum over the positive differences $(a_{j-1} - a_j)$ nor the sum over the negative differences $(a_{j-1} - a_j)$ could be bounded. Hence the a_i converge, and the same for any similarly constructed sequence. □

3.10 Completion of the Proof of 3.8:
Consider a fixed lattice $\mathcal{N}_i$ and a sequence of paths $(u_j)_{j\geq i}$ all of which permit to read the same word ω_i from them. By 3.5 we know that the u_i uniformly converge, and by 3.9 we know that the reading domains for each of letters of ω_i converge, too. As indicated in Proposition 3.8 already, it is now straightforward to conclude that the corresponding subsegments of u that are bounded by the limit points of the nodes are contained in the corresponding corridors or in the extended lattice squares, and hence at the limit points of the nodes itself u must take on values on the corresponding boundary lines. And hence it must perform the same acts of crossings in the same order (up to overlap-phenomena) as each of the u_i. For, if one act of crossing must take place on the parameter interval $[a_i, b_j]$ and some other on $[c_j, d_j]$, and if according to the order of ω_i-letters the $[a_j, b_j]$-crossing shall take place before the $[c_j, d_j]$-crossing, this implies that $a_j \leq d_j$ if the original order of ω_i-letters should be a valid reading. Since by assumption

this inequality holds for all j, it must also hold for the limit. □

3.11 Returning to the Proof of Lemma 3.1:
At this state of the proof we now should pay attention to the first and last letters of u_i for each of the basic directions, since, unlike as for the other letters the exceptional rules of 1.7(ii)–(iii)/B.1.4(3.) may apply here. According to those rules these letters may either refer to a partial or to a completed act of crossing, and even our approximating path u_i may behave differently from what has been discussed before (recall that one of the nodes of a single-split letter does not correspond of the same boundary line on the next level by 3.2(i)–(iii)). In this context we may neglect the zero-split case since in this case the compatibility properties of a word sequence just force that all acts of crossing higher corridors take place inside the zero-split corridor. This forces our limit path to lie in the corresponding corridor, so that the zero-split letter can always be read as an act of crossing that is "uncompleted on both sides". In the standard single-split case (more precisely: in any case where we know all the iterated left split-up letters or all the iterated right split-up letters to show up as usual, but not the opposed iterated split-up letters), the compatibility properties of a word sequence and the above construction of u_i give that all (sufficiently high indexed) u_j have a corresponding corridor entrance or a corresponding corridor exit points. And for the nodes of these points the ordinary estimate of 3.9 holds. Hence all paths $(u_j)_{j\geq i}$ have to intersect the corresponding boundary in the corresponding direction, and this property is passed on to the limit u. Hence the corresponding single-split letters can also be read from u_i, it is just not predictable whether they stand for a completed or for a partial act of crossing.
Finally, our attention should be drawn to the phenomenon that by the rules of B.1.4 reading a word sequence requires more than just reading each of its word independently. In particular the rule B.1.4(vii) requires that the start and endpoints of reading domains have to be reused for a reading of the higher indexed words. This can be practiced for u also, since the start- and endpoints of the reading domains have been defined as limits for the corresponding start- and endpoints of the approximating paths u_i, hence these limits are independent from the level index of our reading. □

Hence the proof of 3.1 is complete apart from the gaps left at 3.4–3.5 which will be closed in the remainder of this section:

3.12 Proposition: *Consider a diagram of the following type* that might appear like a braid diagram without making a distinction between overcrossings and undercrossings: $2{\cdot}n$ strictly monotonous lines are spanned between two horizontal lines such that all intersections between those strings are transverse. All the strings are coloured white or black. At the top white strings and black strings start in pairs. The area between the 1st and $2^{\text{nd}}, 3^{\text{rd}}$ and $4^{\text{th}}, 5^{\text{th}}$ and $6^{\text{th}}, \ldots$ string is hatched, and these hatchings are couloured white and black corresponding to the colour of the strings. The hatchings are extended when strings are intersecting each other, and due to these intersections

hatched regions may overlap also. Apart from this principle set-up, the following rules are to be observed:

(i) Only lines of different coulours may intersect each other.

(ii) White and black corridors may partially overlap, but a complete intersection of such corridors is forbidden.

Let a_i be the number of white strings that start within the first i positions and let b_i be the corresponding number of black strings (so that $a_i + b_i = i$). Denote by a'_i and b'_i the corresponding, number of strings that end at one of the first i positions of the bottom line. *Then* $|a_i - a'_i| \leq 1$ *and* $|b_i - b'_i| \leq 1$ *for all* i *and any such "split-up diagram"* (cf. 3.4, for a picture see 3.13(Fig.i)).

Proof: Assume that we managed to construct such a string diagram which does not satisfy the latter condition, and that in particular i is the smallest index for which the condition is not given. Since $a_0 = b_0 = a'_0 = b'_0 = 0$ and $a_1, b_1, a'_1, b'_1 \in \{0, 1\}$, we have that $i \geq 2$, and hence we can think of a non-empty left subdiagram for which our condition is still satisfied. Further note that clearly $|a_j - a_{j-1}| \leq 1$, and similar for b_j, b'_j and a'_j. Hence the minimality property of i implies that $|a_i - a'_i| = |b_i - b'_i| = 2$. Without loss of generality assume that $a'_i = a_i + 2$ and that $b'_i = b_i - 2$.

CASE ONE: Assume that i is even. Since at the top bar black lines and white lines start in pairs, we can conclude that a_i and b_i are even, also. Consequently $\frac{a_i}{2}$ white and $\frac{b_i}{2}$ black corridors start at the segment $[1, i]$ of the top bar, and at the corresponding segment of the bottom bar we see the end of an additional white corridor, but of one black corridor less. We regard the domain that lies between the segments $[1, i]$ on the top-bar and on the bottom bar as "frame of consideration". Since the lines and the corridors of each colour are not allowed to intersect each other, the rightmost white corridor or our frame of consideration has to intersect the right side of that frame, and on the same side one black corridor has to come in and to end on the bottom bar. From the basic set-up of geometry it follows that these corridors are forced to completely intersect each other, and since this contradicts (ii) our assumption must be false in this case.

CASE TWO: Now assume that i is odd. By the minimality property of i we can conclude that the string starting at the ith position of the top bar is black and that the string ending at the bottom at the ith position is white. Since $i - 1$ is even, as in Case One we can conclude that a_{i-1} and b_{i-1} are both even, too. As a consequence of the minimality property of i we must have that $a'_{i-1} = a_{i-1} + 1$ and $b'_{i-1} = b_{i-1} - 1$, and hence a'_{i-1} and b'_{i-1} are odd. This implies that the rightmost white end point of the segment $[0, i-1]$ of the bottom bar is the left boundary of a white corridor which disappears via the right edge from our frame of consideration. From "$a'_i = a_{i+2}$" we learn where the right boundary of the corresponding corridor ends, namely at the ith position of the bottom line. From counting the numbers of black and white endpoints and by using the property that lines of the same colour cannot intersect each other, it follows that the black line starting at the ith position and on the top bar must also leave our region of consideration via the right edge. Consequently the black corridor bounded by the lines that start at the ith and the $(i+1)^{\text{st}}$ position must intersect completely the white corridor described

before and this contradicts to (ii), as in Case One.
Since we got a contradiction in both cases, this proposition is proved and this completes one of the two gaps in the the proof of the Lemma 3.1, also. □

The following five discussions are then devoted to close the remaining gap in the proof of Lemma 3.1:

3.13 Basic Discussion of the semi-split cases that have been postponed in 3.1–3.12:
Recall that the corresponding cases are given, if there are words in our word sequence ω some of whose letters do according to the exceptional rules 1.7(ii)–(iii) not have the full set of split-up letters (and hence in the diagram 3.4 some of the lines have to be drawn so as to end at ghost-letter markings). However, our Proposition 3.12 was stated and proved in such way that we did not take the possible behaviour of those corridors in split-up diagrams into account that arise when a letter does not have its full set of split-up letters. I.e. the statement of Proposition 3.12 holds in this case also, if we draw the corresponding figure in such a way that those lines which end at markings of ghost letters do also end at the bottom bar. On the other hand the endpoints of these lines do not correspond to the positions of essential nodes any more: Recall that the essential nodes of a single-split letter are at the entrance- and the exit point of the reading domain of this letter, and they are not, as it is defined in the full-split case, distributed to both split-up letters. An entire analogous treatment of single-split and full-split letters would in the single-split case give nodes that were associated to ghost-letters. Hence in the single-split cases we cannot, as in the standard case, get the corresponding estimate by a direct application of Proposition 3.12 in 3.5 for the entire length of a word. The 3.12-estimate is correct after the last single-split letter at the beginning and before the last single-split letter at the end of a word is passed, since then in the counting of our nodes the effects of the nodes that are placed differently due to the exceptional rule for single-split letters are undone. Hence in the following we need in particular to take a look at these initial and terminal segments of paths where within the corresponding word sequence we have not yet or already passed the single-split letters whose split-up partners are ghost-letters at the end or at the beginning of our word, respectively.
In any of the corresponding cases we want to look at the possible intersection patters of the corridors in the split-up diagrams, to associate the attributes "horizontal" or "vertical" to them (since intersecting corridors can only be associated with different of these attributes, we will never have more choice than to interchange them completely) and then to translate the corresponding pattern into the behaviour of those paths in the X_i-plane that are as defined in 3.2. It will be possible to provide pictures (cf. Fig.i) that up to trivial symmetries and easy substitutions of substructures in principle show all the possible phenomena. In this context we should mention, that usually, when we draw two corridors in a parallel way, then there could have been even more than two, and that there is no warranty that a system of parallel corridors in the split-up diagram actually stands for acts of crossing in the same direction (as we will always translate it into our X_i-picture). Hence the corresponding pictures must be understood in such a way that in the corresponding segment the two paths of consideration (the one which reflects

ω_i and the one that reflects ω_{i+1}) also could perform simultaneously non-monotonous movements; but in any of the cases as discussed below it is not hard to see that even if such non-monotonous behaviour is added in the relevant segments to our pictures, then any two correspondingly parametrized segments of our paths are never further apart then three times the diameter of an $\mathcal{N}_i$-region. And this is the result that will enable us to use in these cases the same arguments as given in 3.5 for those cases where 3.12 could be applied. We should also recall that our picture does not show the possible impact that the inessential segmentation points could have on the path u_{i+1} (cf. 3.2(i)–(iii)), hence one has, similarly as in 3.5) to add this possible impact to the given pictures. However, in this case we can only see much less of this effect as in the general case: For, as long as the behaviour of our path is restricted by some $\mathcal{N}_i$-letter which does not have its full set of split-up letters, the corresponding path is said to have started inside the $\mathcal{N}_i$-corridor by performing some act of crossing that is not yet completed. In such a situation the path cannot actually have left the domain of the corresponding corridor: This would either contradict to the direction of the partial act of crossing that our path is said to perform, or it would have to be acknowledged by an $\mathcal{N}_i$-letter and an essential $\mathcal{N}_{i+1}$-letter, also. On the other hand the corresponding effect under discussion rests in the possibility that a path that by the essential segmentation points is said to be inside a lattice square, might by the inessential lattice points actually be forced to go into the adjacent corridor, also. This explains, why this effect cannot badly mess up the pictures given below. However, to a certain extend it must be taken into account.
After these general remarks we shall now look at the various possibilities where some ω_i-letters do not have the full set of split-up letters, separately.

3.14 Discussion of the first of the semi-split cases:
There are two letters with zero split-up letters around in ω_i. Since according to 1.7(iii) their braces cover the entire region, any additional letter of ω_i would have to come with a brace that intersects them both, and this would contradict to the fact that we have only two basic directions, and only braces associated with different basic direction may intersect each other. Hence ω_i can only have the above two letters, that belong to corridors that intersect in a crossing region. According to the principles explained above, ω_{i+1} is forced to remain in the corresponding crossing region. Since any attempt to leave this region would result in additional $\overline{\mathcal{A}_i}$-letters, ω_{i+1} must be empty. Consequently, u_{i+1} is the constant path remaining at the mid-point of the crossing region. And u_i is a path which performs a kind of semicircle around the crossing region. However, this implies that u_i is never further apart from u_{i+1} then permitted.

3.15 Discussion of the second of the semi-split cases (cf. 3.13(Fig.i$_1$)):
Now consider the case that there is only one letter with zero split-up letters around (this case has briefly been described in 3.11, already). Denote it by L. The brace-condition 1.7(i) then implies that all other letters of ω_i have the same basic direction, opposed to the basic direction of L. Hence with one exception u_i performs only moves into one basic direction (which implies that it can only move parallel to one line of corridors), and the exception that is imposed by the zero-split letter makes our path just change the side of the corresponding line of corridors. The essential nodes of u_{i+1} force the same

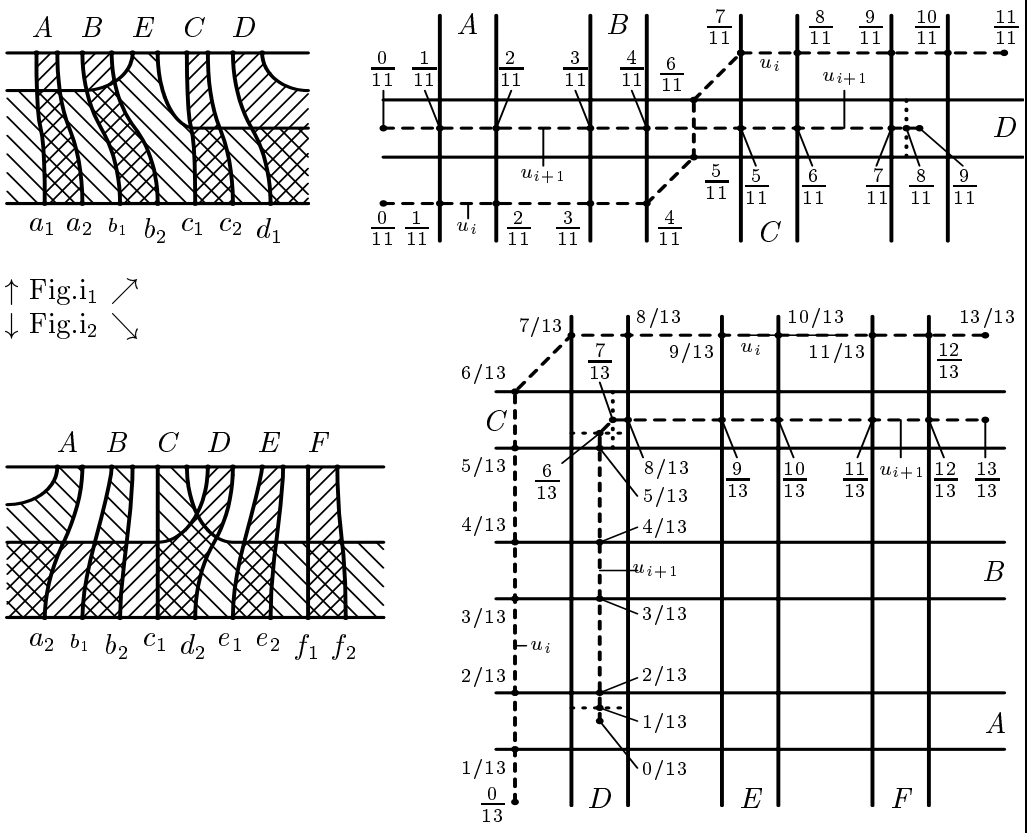


Fig.i$_1$ and Fig.i$_2$ (belonging to 3.13):
The series of Fig.i-figures contains six split-up diagrams that show either the entire or just initial subsegments of the ith and the $(i+1)^{\text{st}}$ word of a word sequence. For each of these split-up diagrams we also give, up to trivial symmetries and reversion effects as described in 3.13, the corresponding pictures of the approximating paths as constructed according to the scheme of 3.2. For that purpose the corridors of the ith level have been drawn by solid lines, the corresponding approximating paths in dashed lines. We did not go for the entire symmetric picture of B.1.1(Fig.g), i.e. we have drawn the width of the corridors smaller than the width of the gap-lines so that one gets a clear picture of what are the corridors and what are the gap-lines. Corridor boundaries of the $(i+1)^{\text{st}}$ level are only partially put into this picture, where they are needed to hold the segmentation points of semi-split letters. Where they are pictured, they are drawn by dotted lines. The nodes for the path on the ith level and the essential nodes for a path on the $(i+1)^{\text{st}}$ level are also marked and are annotated with what might be the corresponding parameter values.

movements as for u_i, but in the interior of the corresponding line of the $\mathcal{N}_i$-corridor, and the inessential nodes may be able to vary these movements a bit, but with the same arguments as in 3.14 our path u_{i+1} cannot leave this corridor. By 3.2 the nodes for u_i crossing our line of corridors are disregarded for parametrization, and hence we get from this geometrical picture that u_i can never be further apart from u_{i+1} than having gone into an diagonally adjacent region. Note that this result can not be overturned if the first and/or the last letter of ω_i is a single-split letter, as also pictured in 3.13(Fig.i$_1$).

3.16 Discussion of the third of the semi-split cases (cf. 3.13(Fig.i$_2$)):

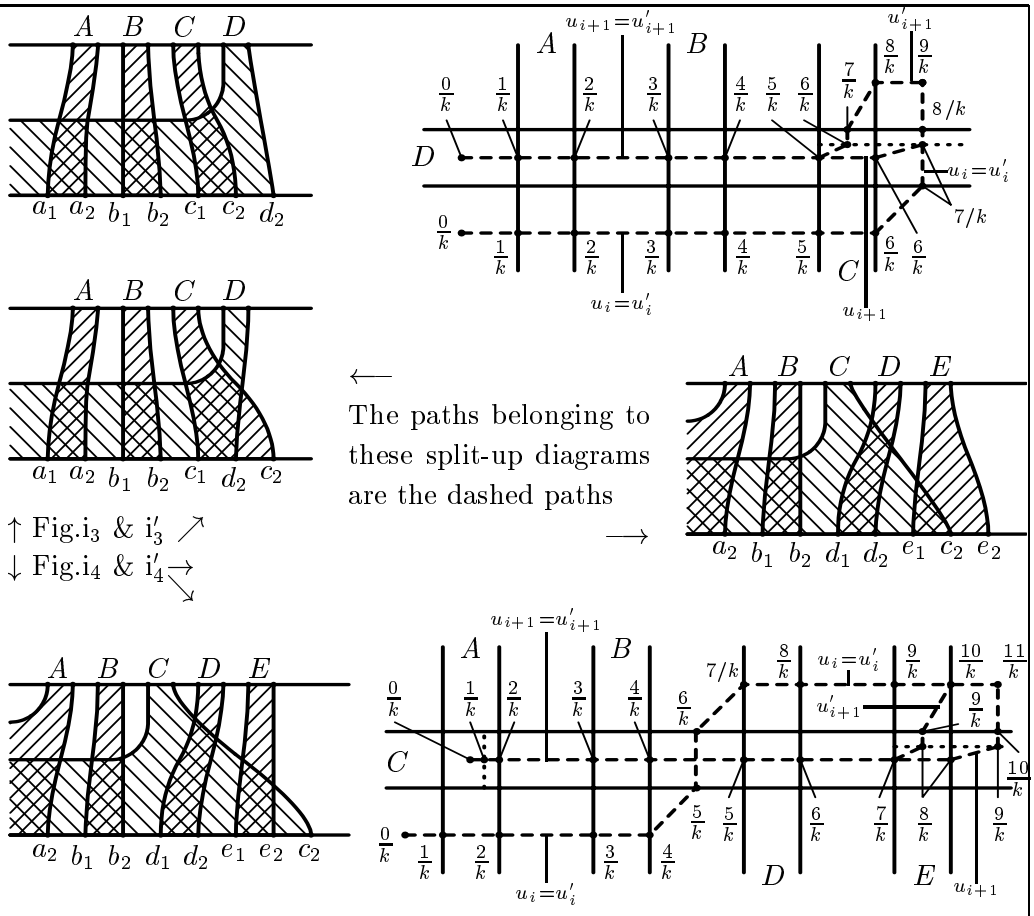


Fig.i$_3$ and Fig.i$_4$ (belonging to 3.13):
With the same convention as for Fig.i$_1$–Fig.i$_2$ this figure gives four more split-up diagrams and the corresponding paths. This time two paths are drawn in each of the corresponding subfigures simultaneously. In order to avoid further use of variables the parameter values have for the ith level been marked in all figures of this series by using the assumption that they are all equidistant. This assumption is unrealistic, since the 3.2-scheme requires to adjust on each level the new nodes by interpolating the ones that have been given on the previous level, and we are not allowed to assume that there is the same number of new nodes in each of these intervals. However, our unrealistic annotation eases our pictures in such a way, that we just need one more variable for the parameter values of the nodes, namely "k" which we use for the total number of node-intervals in the subpictures, where we only see the initial segment.
In drawing these pictures, we imposed another simplification: In order to clearly identify those letters which end at ghost-letter markings, we have drawn them not as required in 3.4 so as to end at the corresponding markings at the bottom lines, but so as to end at the side bars.

Now assume that ω_i has no zero-split letters, but that it does have two single-split letters L_1, L_2 in reversed order. I.e. L_2 comes after L_1, but it is the first split-up letter of L_2, and it is the second split-up letter of L_1, that are missing. If we draw the corresponding split-up diagram according to 3.4, we see that ω_i cannot have any letter between L_1 and L_2, since the corridor that in the split-up diagram would be associated to that letter would have to intersect both the L_1- and the L_2-corridor, and it could only do such a thing if

it could have a colour that is different from white and from black. Since there is no such colour in a split-up diagram, L_1 and L_2 must come in an adjacent position. Because the corridors that are associated with L_1 and L_2 together cover the entire lower halfspace of the split-up diagram, the letters that in ω_i could come before L_1 or after L_2 have to be of the same basic direction as L_1 or L_2, respectively. Hence the picture that we get for this case in 3.13(Fig.i$_2$) is somehow similar to the the preceding one with just one 90°-turn added: u_{i+1} can again only move inside $\mathcal{N}_i$-lines of corridors, and when it passes the two single-split letters, it turns to another basic direction and continues in another line of corridors, while u_i moves the whole time parallel to the corresponding line(s) of corridor, passing the crossing region with the turn on the outside. As in the preceding case, the first and the last letter could be additional (non-reverted) single-split letters. Their appearance would not overturn the present picture, they would just cause that the start- (or the end)point of u_i and u_{i+1} do not have the same orthogonal projection to corresponding line of corridors, but they differ by one grid-unit. However, as in the standard situation 3.17 our method of parametrization undoes this effect already in the start phase.

3.17 Discussion of the last of the semi-split cases: Now we assume that we have only single-split letters and that the order is not reversed. This case might be understood as the standard case. As pointed out in the introductory paragraph of 3.13 already, the parameter-distortion effect is undone after all single-split letters have been passed, and hence we need not consider the middle segment of our words or paths, since for those segments the estimates of 3.5 are valid. Hence we can restrict our consideration to the start and terminal segment of our words, and for symmetry reasons we need only consider the first. I.e. the general situation is already given by considering "$L_1L_2L_3\dots L_{k-1}L_k$" as the initial segment of ω_i when we have that $BD(L_1) = BD(L_2) = \dots = BD(L_{k-1}) \neq BD(L_k)$ and where L_k is a single-split letter (cf. B.1.2(i)). In addition L_1 may or may not be a single-split letter. However, in this case there is more freedom in what order the split-up letters of the corresponding letters could appear than in 3.14–3.16: The second split-up letter of L_{k-1} might come after the second split-up letter of L_k, (3.13(Fig.i$'_3$)) or conversely might L_{k+1} have one (or even two) of its split-letter before the second split-up letter of L_k (3.13(Fig.i$'_4$), 3.13(Fig.i$_4$)). We might even see the latter phenomenon for the split-up letters of more than just one of the successor-letters of L_k. For that reason we have drawn four figures for that case, in principle just two figures, but each of them showing two routes of a path. Three of the four path-routes show the behaviour if one of the the latter reversion phenomena occurs. The fourth one is devoted to showing the case that is free of such additional reversion phenomena (3.13(Fig.i$_3$)).
Note that in 3.13(Fig.i$_4$) we see a phenomenon that we have not seen before, namely two paths that in some corresponding segments are not in diagonally adjacent squares, but further apart. The estimate is still within the tolerance of Formula 3.5(1)), but if we add the effect as described in 3.5, we see that in these exceptional cases we only obtain a worse estimate instead of 3.3/3.5(1), namely that

$$\|u_i(t) - u_{i-1}(t)\| \leq 4 \cdot \sqrt{2} \cdot (\frac{1}{3})^i. \tag{1}$$

For, since "4" is now the ultimate global bound, the essential conclusion of 3.5 is sufficient to give uniform convergence in this case, also. Having achieved this ultimate bound, the proof of Lemma 3.1 is now definitely complete. □

§4. Some observations on the Reading- and the Realization process

4.1 Lemma:

(i) *The same triple sequences can be read from paths that just differ by their parametrization.*

(ii) *Let u_1, u_2 be two paths both of which permit to read same triple sequence (u, ω, v) from them. Then $u_1 = u_2 \circ \varphi$, where φ is a monotonous parameter transformation, i.e. u_1 and u_2 must describe the same abstract path (cf. A.5.1).*

Skeleton of the Proof — Ad (i): This is an immediate observation, since none of the demands of how to read a word sequence from a path did depend on how the path was parametrized. In principle this observation had implicitly been used in B.1.6(1) where we had used the system of abstract paths as domain of definition for our mapping Ψ.

Ad (ii): This is the non-trivial direction of this lemma whose proof will be lengthy. I.e. the following paragraph will only give the skeleton and basic definitions, while Propositions 4.2–4.7 will be needed in order to complete the proof:

For the sake of simplicity we will in (1)–4.6 first assume that neither u_1 nor u_2 has a segment on which it is constant, and we will restrict ourselves to a brief explanation for how to handle the more general case in 4.7. The basic observation that will allow us to prove the existence of a parameter transformation $\varphi : [0,1] \to [0,1]$ with $u_2(\varphi(t)) = u_1(t)$, is now the following:

By B.1.4 such a reading implies that each letter of the corresponding word sequence ω (which by assumption is the same for u_1 and for u_2), is in the one hand associated with a certain subinterval $[t_1, t_2]$ of $\mathrm{Dom}(u_1)$ and on the other hand with $[s_1, s_2]$ of $\mathrm{Dom}(u_2)$ ("Dom" = domain of definition). This allows us to obtain a partial definition of φ, namely

$$(1) \qquad \varphi(t_1) := s_1 \qquad \text{and} \qquad \varphi(t_2) := s_2.$$

In the following we want to prove that

(1.) This partial definition of φ is well-defined, i.e. if two different assignments of φ-values are obtained for the same parameter value t since with respect to u_1 this parameter value is an endpoint of a reading domain for more the one interval, then the φ-values assigned to this parameter value must coincide, too.

(2.) This partial definition of φ is strictly monotonous. i.e. for $t_1 < t_2$ we get $\varphi(t_1) < \varphi(t_2)$.

(3.) The parameter values for which a partial definition can be obtained that way, form a dense set D.

(4.) These partial definitions of φ permit the extensions to a continuous and strictly monotonous function $\varphi : \mathrm{Dom}(u_1) \to \mathrm{Dom}(u_2)$ with $\varphi(0) = 0$ and $\varphi(1) = 1$.
(5.) Even for φ extended that way we get: $u_1(t) = u_2(\varphi(t))\ \forall t \in \mathrm{Dom}(u_1)$.
We will only be able to prove (1.)–(5.) if, in addition, we assume that the reading domains of u_1 and u_2 have been adjusted in a minimal way in the sense of 2.7. Since by 2.7 this can be done without affecting the word sequences we do not loose generality by this assumption.

4.2 Proposition: *φ as defined in 4.1(1) satisfies 4.1(1.).*

Proof: If initial points and endpoints of reading domains coincide, for the corresponding parameter values the path u_1 must be simultaneously on the boundaries of different corridors. In the following we will discuss the cases separately where the corresponding corridors have the same or different basic directions:
CASE ONE: Both corridors have the same basic direction, say vertical. Now our network is defined that way that the boundaries of two vertical corridors do not coincide unless one corridor is a split-up corridor (maybe by more than one split-up steps) of the other. In this case the reading condition B.1.4(vii) enforces us to adjust the corresponding reading domains, also, and hence this property holds the same for u_1 and for u_2. Thus we cannot get a contradiction to the partial definition of φ from such a coincidence of corridor boundaries.
CASE TWO: Now assume that u_1 goes through a crossing point between a horizontal corridor boundary and a vertical corridor boundary, and that according to the way ω is read from u_1, this point is a coincidence point for the corresponding endpoints of the reading domains. In the following we want to argue that such a coincidence of endpoints of reading domains is reflected within the structure of a word sequence, such that knowing that same word sequence ω could be read from u_1 and u_2 gives that we find the same coincidence for u_2. Hence we have to ask how the endpoints of the reading domains are represented by the corresponding word sequence. This will be answered by the following definition and the subsequent Proposition 4.5: By using the statement of that proposition which implies that it is characterized by the word sequence ω whether two start- and endpoints of reading domains do or do not coincide, we are able to rule out that our partial definition of φ can be contradictory. Hence modulo 4.5 the proof of 4.2 is complete. □

4.3 Definition of a combinatorial description of a start- or endpoint of a reading domain:
The natural definition for a domain of the ith level is apparently associating the place before or after the corresponding letter l in the ith word of our word sequence with the start- or endpoint of our reading domain, respectively. It further seems appropriate to extend these places throughout the entire word sequence, and the natural definition for that is apparently to also associate for each level $j \geq i$ the place before the corresponding iterated left split-up letter of l or after the corresponding right split-up letter of l with the start-point or with the end-point of the corresponding reading domain that corresponds

to l, respectively. By this definition each start- and end-point of the reading domain gives rise to an infinite combinatorial system of places within the corresponding word sequence.

4.4 Remark *that coinciding parameter values do not necessarily result in coinciding combinatorial descriptions in the sense of Definition 4.3:*
In order to see that this simple relation cannot hold, consider the following situation: Think of a path which approximately diagonally goes through a crossing region of corridors, and that way ends the acts of crossing a vertical and a horizontal corridor at the same time by going through the corner point of a crossing region. The corresponding places within the associated word sequence that describe the endpoints of these passages are on the one hand places to the right of horizontal letters, and on the other hand they are given by the places to the right of vertical letters. Such places can never coincide. However, the relation between coinciding endpoints of reading domains and the associated system of word sequences can be combinatorially understood, and the following proposition is devoted to establish that relation:

4.5 Proposition: *Coinciding endpoints of reading domains correspond by the Combinatorial Description 4.3 to systems of places which are just a bounded distance apart.*

Proof: We have to show that if and only if the sequence given by the number of letters that stand on each level index between the corresponding places is a bounded sequence, then the corresponding endpoints of reading domains coincide. We are going to show the contraposition of this "if and only if"-claim:

"$\neg(\ ) \Longrightarrow \neg(\ )$" Assume that a and b are distinct parameter values each of which is the start- or the endpoint of a reading domain. Without loss of generality, $a < b$ so that we can ask for the behaviour of our path $u_{|[a,b]}$. Since at the beginning of the proof of 4.1 we agreed not to discuss paths with constant segments, $u_{|[a,b]}$ is a non-constant path and hence on a sufficiently fine level i it has to perform a completed act of crossing a corridor. If a' and b' are the bounding parameter values of this act of crossing, we may assume that $a < a' < b' < b$. If a should have been the endpoint of some act of crossing a corridor and b should have been the start point, it is clear that on the corresponding level of the associated word sequence the letters of the $[a', b']$-crossing have to appear between the places that correspond to a and that correspond to b. Since the same holds for all the iterated split-up letters and since $[a', b']$ is an interior segment of our path u, so that the corresponding letter has to show up with its full set of split-up letters (that makes 2^{j-i} for the level j), we have proved the existence of an unbounded number of letters between the places that correspond to a and to b in that case. In the other cases, where a is the start-point of some reading domain and/or b is the end-point of some reading domain, we first have to find a sufficiently fine level where reading domains for the corresponding iterated left split-up letters of the $[a, ?]$-reading and the $[?, b]$-reading have finished till a' or have not started before b', respectively. Note that 2.7 gives precisely the situation where this argument does not work. In this context we have a path which, after having passed the start point of its reading domain, remains on the boundary before it enters the interior, and

hence a situation where even on a sufficiently high index level we will not get that the length of the $[a,?]$- and $[?,b]$-reading domains tend to zero. But by assuming minimal reading domains some subcases of 2.6 have been ruled out; and for those which are not ruled out we get the information that during its segment on the boundary of our corridor our path has to intersect other corridors whose reading domains of its split-up letters get in the usual way arbitrary short. In addition, we get from the minimality assumption that the iterated left split-up letters of our $[a,?]$-readings are trapped between the split-up letters of these other corridors (cf. D.10.8). Based on considering auxiliary reading domains such as $[a',b']$ or the ones that can be found in the special case above, then essentially the same arguments as in first part of the proof of this direction hold.

"$\neg(\ ) \Longleftarrow \neg(\ )$" Now assume that some path u and its associated word sequence ω are given and that for two reading domains of ω the places within ω that correspond to the start- and endpoints have been marked in such a way that the number of letters between the corresponding sequence of places is an unbounded sequence. Let a and b be the parameter values that correspond to these start- and end-points, and, without loss of generality let $a < b$. Can we conclude that within ω between the places that correspond to a and to b an act of crossing a corridor is found which u entirely has to perform between its parameter values a and b? – Well, there might be exceptions due to the rule 1.3(ii) that the orders of letters is arbitrary if acts of crossing corridors overlap. However, this exception can affect at most four letters for each level index, namely at most one horizontal an one vertical corridor in which $u(a)$ could be contained, and another two corridors in which $u(b)$ could be contained. Hence, if more than five letters show up between the places corresponding to a and corresponding to b, they can only refer to acts of corridor crossings which u definitely performs after having completed all acts in which it might be involved at a and before starting those acts of crossing corridors in which it might be involved at b. Since we assume that the number of places between the places corresponding to a and corresponding to b is unbounded, we can now conclude that there are such letters, hence that u does perform such acts of crossing corridors, and this rules out that a and b could have been the same parameter value. □

4.6 Proposition: *φ as defined in 4.1(1) satisfies 4.1(2.)–(5.).*

Proof – Ad 4.1(2.): The arguments for monotony have in principle been already given in the course of the discussion of 4.5: If two endpoints t_1, t_2 of reading domains are situated on different parameter values of u_1, the path u_1 has to perform on the intermediary segment some corridor crossings which have to be acknowledged by letters of the words of ω with sufficiently high index. Consequently the letters corresponding to t_1 and t_2 are such letters of ω, which are either not in adjacent positions or there are higher indexed letters which are inserted in between them. Hence, this rules out that the reading domains that give $\varphi(u_1)$ or $\varphi(u_2)$ (if we associate them with sufficiently high split-up corridors) could come on another order as the reading domains that gave t_1 or t_2.

Ad 4.1(3.): Since our path was supposed not to have any segments at which it is constant and since our corridor network is dense, our path has at any interval to cross some of the sufficiently high indexed corridors. This makes sure that above in "Ad 4.1(2.)" we get definitions of φ on an dense set.

Ad 4.1(4.): Note that the monotony 4.1(2.) implies an alternative method of extending φ : To each value t of $\mathrm{Dom}(u_1)$ the interval $[\sup_{(t\leq t_i)}(\varphi(t_i)), \inf_{(t\geq t_i)}(\varphi(t_i))]$ can be associated. With luck, this interval only contains a single point, which can then be defined as $\varphi(t)$. However, if we obtained a non-constant interval that way, this would imply that $\varphi_{|D}$ could not take on any value inside this interval, and this yields a contradiction since the same arguments that gave in 4.1(3.) that D is dense can also be used to conclude that $\varphi(D)$ is dense. The resulting φ is continuous, since the last argument can rule out that it performs jumps. Since the same argument could be applied to the inverse of φ, we know that φ cannot have subintervals on which it is constant, and hence that the entire φ is strictly monotonous.
The fact φ does not perform jumps does also rule out that $\varphi(0)$ could be bigger than zero or that $\varphi(1)$ could be smaller than one.

Ad 4.1(5.): Note that for each $t \in D$ either the x- or the y-coordinate of $u_1(t)$ or of $u_2(\varphi(t))$ have to coincide, since this is the corresponding constant coordinate of the boundary line of the corridor to be crossed. However, the other coordinate also has to coincide, since by the same mechanism as in 1.6 this coordinate is determined by $\overleftarrow{\omega}_i$, $\overrightarrow{\omega}_i$ and the same prescription for how to compute for each place between adjacent letters the corresponding lattice square. Since this can be done for all sufficiently high level indices i, and since our corridor network is dense, this uniquely determines the u_1- and the u_2-value. From that mechanism we get that our definition for $\varphi_{|D}$ satisfies the condition 4.1(5.). Since D is dense and since u_1, u_2 and φ are continuous, this condition holds for all $t \in [0, 1]$. □

4.7 Remark on how to prove Lemma 4.1 if we do not want to assume that u_1 and u_2 do not have intervals on which they are constant:
Observe that in this case we may get a φ which could be constant for some interval as well (since u_1 is constant on the corresponding interval), or we could get a φ which has places at which it is not continuous any more, and where there exists only an upper limit and a lower limit such that the upper limit is smaller than the lower limit (here φ jumps over regions where u_2 is constant). However we will consider φ even in such a case as a parameter transformation, as long as $u_2 \circ \varphi$ still is continuous. It seems not worth to explain in detail how to adapt the above definitions of φ to that situation, also (cf. E.1.7 as a lemma giving technical hints for how to get rid of constant intervals by some changes of the parametrization). By that it should be clear that the arguments of 4.2–4.6 can be adapted to this situation, also, and hence we consider Lemma 4.1 now as proved. □

4.8 Lemma:

(i) Any bounded word sequence ω (i.e. $(\ell(\omega_i))_{i\in\mathbb{N}}$ is a bounded sequence of non-negative integers) can only describe a constant path (ℓ = length of the word = number of the letters of the word).

(ii) Any reading from a constant path gives a bounded word sequence with at most four letters on each level.

(iii) Any non-bounded word sequence and any word sequence that has at least five letters on one of its levels can only describe a non-constant path.

(iv) Any reading from a non-constant path can only give a non-bounded word sequence.

Proof: Only (i) and (ii) have to be proved, (iii) and (iv) then follow from contraposition.

Ad (i): Let u be the path associated with ω, and assume it would not be constant, i.e. there exist parameter values t_1 and t_2 with $u(t_1) = Q_1 \neq Q_2 = u(t_2)$. Since $Q_1 \neq Q_2$, there exists either a horizontal or a vertical line of corridors such that Q_1 and Q_2 lie on different sides of this line of corridors. Hence any path running from Q_1 to Q_2 has to cross this line of corridor, its two split-up lines of corridors, the four split-up lines of the two split-up lines, and so on. Hence any path running from Q_1 to Q_2 can only be described by an unbounded word sequence, which contradicts to our assumption.

Ad (ii): Note that, even if our path is assumed to be constant, the freedom described in 1.3(i)–(iii)/B.1.4 makes it possible nevertheless to associate (partial) acts of crossing corridors to such a path. The rule 1.3(iii) that a partial act of crossing a corridor, when performed at the beginning or at the end of the path, may yet be acknowledged by a letter in the word in a similar way as a completed crossing, can also be applied to a constant path that has its value inside the corridor: Provided the first and the last entry of the corresponding triple sequence are accordingly adjusted, such path can also be regarded as having performed an (uncompleted) act of crossing. This also applies to paths that take on their constant value on the boundary of a corridor. In addition, such paths can be treated as if they perform two uncompleted acts of crossing; they can be treated as if they start an act of crossing in the interior of the corridor, then finish this act by reaching the boundary, then turn around and start a new (uncompleted) act of crossing in the opposite direction. This interpretation would give two letters for the corresponding word of ω. And, in addition, if a constant path takes on its values at crossing point of the boundaries of a horizontal corridor and of a vertical one, what has been explained in the previous sentence can be applied to both of those corridors, and in this case a constant path can even contribute four letters to the corresponding word sequence. □

4.9 Definition and Remark: ω^{-1}, the *"inverse of the word sequence ω"*, is defined as $(\omega_i^{-1})_{i\in\mathbb{N}}$, i.e. as the word sequence in which the ith word is obtained from ω_i by writing the letters in the inverse order and inverting the exponent of each letter. Note that this procedure is compatible with all demands of 1.5, i.e. what results by inverting a valid word sequence gives again a valid one. If such an inversion process is to be extended to the entire triple sequence, the sequences of the first and third entries have to be interchanged. Further, it is immediate from the reading conditions B.1.4 that if

ω can be read from some path u, then ω^{-1} can be read from the inverted path $\overline{u}$ (cf. A.5.1).

4.10 Remark: This paper uses techniques of two earlier works of the author which have have not yet appeared or brought into its final shape yet, respectively. Firstly, the concept of word sequences was developed in [Zas2].§2 for a description of the fundamental group of the Hawaiian Earrings, and it is also used in [Zas4] for that purpose. For further remarks see A.4.4. The other paper [Zas3] is about generalizing the concept of universal coverings. The results are that each subspace of $\mathbb{R}^2$ should have a universal covering, and that each subspace without inner points can be regarded as a generalized graph in the sense of A.4.13. The universal cover of such a space is then an $\mathbb{R}$-tree permitting a natural contraction of any embedded sphere. Hence it was natural to guess that even bringing open sets into play should not mess up this construction. However the results of this paper are not as general as the first guess since the proof definitely needs the Steps of Sections §A.2 and §A.3, i.e. the replacement of an arbitrary subset by a nicer one.
However, even if this piece of work was motivated by constructing contractions for those generalized universal covering spaces coming from subspaces of $\mathbb{R}^2$, for understanding this paper it is not necessary to have studied the author's definition in [Zas3] of a generalized universal covering, since here a concept was chosen that makes references to generalized universal covering spaces redundant. However, at this place it should once be mentioned that instead of using the concepts about "homotopy combings" and "homotopical nearby paths" in A.5.5(ii) and A.5.2(2), respectively, one could have also worked with combings on generalized universal covering spaces and with a metric on those spaces. And this was the author's original way of thinking of these concepts.

§5. Splitting word sequences at position sequences

5.1 Motivating Remark: One of the most essential operations that can be applied to a path $u : [0,1] \to X$ is to fix some parameter value t and, by doing so, to split this path up into a product of two subpaths $u = u_{|[0,t]} \cdot u_{|[t,1]}$. Since within X all paths can be described by word sequences, we now want to discuss the corresponding procedure for word sequences, i.e. how all words of a word sequence can synchronously be split up such that the sequence of the initial parts describes $u_{|[0,t]}$ and that the sequence of the end parts describes $u_{|[t,1]}$ as a valid word sequence. In principle, a finite word with $k+j$ letters can be uniquely split up by naming an integer between 0 and $k+j$ (e.g. k in our case) in order to define that the first k letters should go into the first part and that the remaining j letters should go into the second part. Similarly, it will take a sequence of such integer position numbers for describing how to split up each word of a word sequence. However, not every sequence of non-positive integers all of which are smaller than the length of the corresponding word will give that the two resulting parts of a word sequence again satisfy all compatibility properties 1.5(i)–1.7(iii) of word sequence. In brief, we will call those sequences of position numbers which do give valid

split-ups of a particular word sequence *"position sequences"*. The following is about to understand on a combinatorial level, based on the inserting and split-up structure of a word sequence, which sequences of position numbers can be called "position sequences".

5.2 How to enumerate positions and letters of a word:
Note that for purposes such as described in the preceding remark we have to carefully differentiate between counting the letters of a word on the one hand, and counting the positions between the letters on the other hand. Within this paper we will have to work with both concepts, and the purpose of the formula displayed below is to show the conventions of both concepts by considering the Latin alphabet as a word:

(1)
$$\begin{array}{ccccccccccccccccccccccccc} & a & & b & & c & & d & & e & & f & & g & \dots & & x & & y & & z & \\ \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & & \uparrow & & \uparrow & & \uparrow & \\ 0 & | & 1 & | & 2 & | & 3 & | & 4 & | & 5 & | & 6 & | & 7 & & 23 & | & 24 & | & 25 & | & 26 \\ & 1 & & 2 & & 3 & & 4 & & 5 & & 6 & & 7 & & & 24 & & 25 & & 26 & \end{array}$$

5.3 Definition of the set of position sequences of a word sequence:
(i) Recall that by 1.5, with some exceptions at the beginning and at the end of a word, usually a letter of the ith word of a word sequence is associated to two "split-up-letters" of the $(i+1)^{\text{st}}$ word. This makes us now define for each word sequence ω two infinite systems of functions $(\nu^{\text{r}}_{\omega,i})_{i\in\mathbb{N}}$ and $(\nu^{\text{l}}_{\omega,i})_{i\in\mathbb{N}}$ which relate the corresponding two position numbers of letters to each other. If ω_i denotes the ith word of our word sequence ω and ℓ denotes the length of a word, then $\nu^{\text{l}}_{\omega,i} : \{0,1,2,\dots,\ell(\omega_i)\} \to \{0,1,2,\dots,\ell(\omega_{i+1})\}$ and $\nu^{\text{r}}_{\omega,i} : \{1,2,3\ ,\dots,\ \ell(\omega_i)+1\} \to \{1,2,3,\dots,\ell(\omega_{i+1})+1\}$ are defined as follows:

$$\nu^{\text{l}}_{\omega,i}(k) := \begin{cases} 0 & \text{if } k = 0, \\ 0 & \text{if according to the exceptional rule for letters near the beginning of a word the first split-up letter of the } k^{\text{th}} \text{ letter of } \omega_i \text{ does not show up in } \omega_{i+1}, \\ \text{position number of the left split-up letter of the } k^{\text{th}} \text{ letter of } \omega_i & \text{in any other case,} \end{cases}$$

$$\nu^{\text{r}}_{\omega,i}(k) := \begin{cases} \ell(\omega_{i+1})+1 & \text{if according to the exceptional rule for letters near the end of a word the last split-up letter of the } k^{\text{th}} \text{ letter of } \omega_i \text{ does not show up in } \omega_{i+1}, \\ \ell(\omega_{i+1})+1 & \text{if } k = \ell(\omega_i)+1, \\ \text{position number of the right split-up letter of the } k^{\text{th}} \text{ letter of } \omega_i & \text{in any other case.} \end{cases}$$

Note that in principle we will be omitting the two lower indices of the ν-functions in most formulae, if they follow from the context.
(ii) Let ω be a word sequence. Then $\mathbb{P}(\omega)$ is the set of all sequences $\kappa = (k_i)_{i\in\mathbb{N}}$ of non-negative integers which satisfy the following two conditions
(1.) $k_i \le \ell(\omega_i)$

(2.) If ω is split up along κ in the sense of 5.1, then the two resulting sequences of words are both word sequences in the sense of 1.5/1.9.

The elements of $\mathbb{P}(\omega)$ are called "position sequences".

(iii) Note that $(0,0,0,\ldots) =: 0^{\mathbb{N}}$ and $(\ell(\omega_1),\ell(\omega_2),\ell(\omega_3),\ldots) =: \ell^{\mathbb{N}}$ are always to be regarded as well-defined position sequences that induce those trivial split-ups of a word sequence where the the first or the second of the resulting sequences are empty, respectively.

We hereby agree to use a similar notation as it is standard for identifying subsegments of a path in order to describe those word sequences which arise from such trivial or a non-trivial split-ups of a given word sequence. If ω is split up along $\kappa \in \mathbb{P}(\omega)$ into two halves, then the resulting word sequences will be denoted as $\omega_{\lfloor[0^{\mathbb{N}},\kappa]}$ or as $\omega_{\lfloor[\kappa,\ell^{\mathbb{N}}]}$. In case such processes are iteratively performed, also word sequences such as $\omega_{\lfloor[\kappa_1,\kappa_2]}$ with $\kappa_1,\kappa_2 \in \mathbb{P}(\omega)$ may be constructed.

(iv) Those letters whose position numbers can be obtained by $\nu^{\mathrm{l}}_{\omega,i+j}\circ\ldots\circ\nu^{\mathrm{l}}_{\omega,i+1}\circ\nu^{\mathrm{l}}_{\omega,i}(k)$ and by $\nu^{\mathrm{r}}_{\omega,i+j}\circ\ldots\circ\nu^{\mathrm{r}}_{\omega,i+1}\circ\nu^{\mathrm{r}}_{\omega,i}(k)$ will be called the *"iterated left"* and the *"iterated right split-up letters"* of that letters that on the ith level appears at the kth position.

As an alternative in particular to condition (2.), position sequences could have also be defined by purely combinatorial means based on the $\nu_{\omega,i}$-functions. It is the purpose of the following lemma to establish the corresponding combinatorial description of position sequences:

5.4 Lemma:

(i) Let $\omega = (\omega_i)_{i\in\mathbb{N}}$ be a word sequence and $\kappa = (k_i)_{i\in\mathbb{N}}$ a sequence of non-negative integers satisfying 5.3(1.) and, in addition, that

$$\max\{\nu^{\mathrm{l}}_{\omega,i}(j) \mid j \le k_i\} \le k_{i+1} < \min\{\nu^{\mathrm{r}}_{\omega,i}(j) \mid j > k_i\}. \tag{1}$$

Then κ is a position sequence of ω.

(ii) Let $k_1,\ldots,k_n$ be a finite sequence of integers satisfying the conditions of (i). Then there is at least one possibility to extend this finite sequence to a position sequence of ω.

Proof — Ad (i): Observe that, since ω is assumed to be a word sequence and hence satisfies all compatibility demands 1.5(i)–1.7(iii) already, even if ω' and ω'' result from an unfortunate split-up of $\omega = \omega'\omega''$, they have to satisfy most of the compatibility demands already. In principle there remains only one possibility of how ω' or ω'' could fail to satisfy the conditions of a word sequence: namely if both split-up letters a_1 and a_2 of some letter A of ω'_i have gone into ω''_{i+1}, i.e. into the other split-up sequence. However, Formula (1) is precisely created so as to prevent this: It makes sure that at least one of the split-up letters has always to be in the same part. In view of 1.7(ii), i.e. of the exceptional rule for letters near the beginning or near the end of a word sequence (this rule may be applied to the end of ω' or the beginning of ω'' also), this is sufficient to make sure that ω' and ω'' satisfy all facilities of a word sequence again.

Ad (ii): A geometric picture in order to prove this item is developed:

Think of the words of ω as written consecutively in different lines above each other, ω_1

in the topmost line. Then for all $i \in \mathbb{N}$ visualize the functions $\nu^{\mathrm{l}}_{\omega,i}$ and $\nu^{\mathrm{r}}_{\omega,i}$ by putting arrows between the ith and $(i+1)^{\mathrm{st}}$ line that connect each letter of ω_i with its left and its right split-up letter. This gets a graph-like structure into this picture. From each letter of ω_i there are two arrows emanating which end up at different letters of ω_{i+1}, thus we may call them "left" and "right" arrows. The condition (1) can precisely be fulfilled if for any two different letters L_1, $L_2 \in \omega_i$ with L_1 being placed to the left of L_1 the left arrow emanating from L_1 ends to the right of some right arrow emanating from L_2. Note that it would be precisely the sixth case of 1.8(ii) if this conditions was not satisfied, and we have already pointed out in 1.8(ii) that this sixth case is not consistent with Definition 1.5. □

5.5 A geometric picture of the split-up structure and the corresponding set of position sequences of a word sequence: (cf. D.10.4(Fig.o))
By extending the concept of 5.4(Ad (i)) in the following a picture of how to construct two new graphs or graph-like structures is developed that are associated with a word sequence ω. Usually we will think of both graphs as being drawn into the same picture, where they will appear as being dual to each other. According to that duality all elements of our drawings are attached with one of the following attributes, white or yellow:
We regard the area that shall take up our picture as being subdivided in countably many horizontal lines (cf. 5.10), where the ith line will be associated with the ith level of our word sequence ω. The lines shall be enumerated by $\mathbb{N}$ with "1" being associated with the topmost line. We write (the letters of) ω_i into the ith line, and by doing this we mark $\ell(\omega_i)$ linearly ordered white places within that line that correspond to the letters of ω_i within the ith line. At the end, $\ell(\omega_i)+2$ white places within each line will be named, and we achieve this by additionally marking one extra place to the left of each word and one extra place to the right of each word; these extra places may be associated with "ghost letters" (cf. 1.7). We then achieve a white graph-like structure, by drawing those white arrows in this diagram the reflect the (pre)image-relations as induced by the ν^{r}- and the ν^{l}-functions as defined in 5.3(i). Since these functions are not defined on the sets $\{1,2,\ldots,\ell(\omega_i)\}$ but on $\{0,1,2,\ldots,\ell(\omega_i)\}$ or on $\{1,2,3,\ldots,\ell(\omega_i)+1\}$, we need the ghost letters to have a complete picture of our functions in the figure. Hence from each letter in one of our lines there are arrows emanating and pointing to some of the letters in the line below, one arrow from each of the ghost letters and two arrows from each of the real letters. For each real letter these two arrows can never point at the same image letter by 1.7, and the condition 1.8(ii) translates into the fact that both of the two arrows cannot intersect the two arrows emanating from some other letter. By associating all letters and ghost letters with vertices, and all arrows with edges, the white entries of our picture as described so far may be interpreted as a generalization of a finitely branching infinite union of trees as described in 5.10 (note that the non-split-up letters do not have an upward connection). We shall now describe how a yellow graph shall be drawn into our figure in such a way that it is dual to this white graph-like structure.

In the same lines as our letters of ω the vertices for the yellow graph can be found; they are placed between the letters, and in the gaps between the letters and ghost-letters of ω. I.e. the ith line contains $\ell(\omega_i)+1$ yellow vertices; the leftmost and the rightmost entry of each line being a white ghost letter, so that the first three entries of each line that contains a non-empty ω_i are a white ghost-letter, followed by the leftmost yellow vertex, and then comes the first (real) letter of ω_i. We get a similar picture for the end of each such line. An empty ω_i gives rise to three entries in a line only, two ghost letters and one yellow vertex in between. Having obtained the vertex structure for the yellow graph that way, we now can complete the drawing of this graph by inserting its edges according to the following rules: Two yellow vertices shall be connected by a yellow edge, if they are placed in adjacent lines and if this connection can be drawn without intersecting both of the two white arrows emanating from one of the letters. Note that the latter condition is equivalent to 5.4(1). *Hence we get that the infinite monotonously downleading paths starting at the topmost line (we will call them "infinite branches", even if the yellow graph is not necessarily a tree) correspond to the elements of* $\mathbb{P}(\omega)$.
In this context it seems convenient to alter the present picture slightly by introducing an additional line above the line with index 1, that had been the topmost so far. The indices of our lines are not affected by this step; we will regard the new topmost line as the zeroth one. The vertices entered in this line are the same as for an empty word: Two white ghost letters and one yellow vertex in between. There are only two white arrows between the zeroth and the first line, namely the obligatory arrows connecting the left ghost letters and the right ghost letters with each other. Hence each yellow vertex of the first line is to be connected with the single yellow vertex of the zeroth line and hence that yellow vertex can be considered as the "root vertex" for the entire yellow graph. The yellow and the white graph together are called the *"dual split-up picture"*. Sometimes, the white or yellow graph, only, will also have to be considered. Then it will be called the *"split-up graph (of a word sequence)"*, or the *"graph of position sequences"*. In the following we proceed by remarking some properties of this picture:

5.6 Remark: Note that pairs of white arrows intersecting each other in pairs could give the only obstruction for finding from an arbitrary yellow vertex at least one upward leading or one downward leading yellow edge. Since according to 1.8(ii) this obstruction can not be given, there *will never be an obstruction for finding an infinite branch through an arbitrary yellow vertex* that is placed in the interior of our graph of position sequence. For yellow vertices that can be found near to the left or right side of our picture one could think of another obstruction, namely of a pair of white arrows starting at the boundary vertices and both going to the interior of our picture. However, the white boundary vertices belong to ghost-letters, and according to 5.3(i) such pairs of arrows are not defined. Hence there is not any need to exempt any yellow vertices from the statement that *there always exist infinite yellow branches through such a vertex.*

5.7 Proposition: *Let p_1 and p_2 be two monotonous yellow edge-paths in the dual split-up picture that intersect each other within the graph of position sequences. Then*

the part of p_1 that lies above the intersection point can be plugged together with the part of p_2 that lies below the intersection point to give again a monotonous yellow edge-path.

Proof: If the intersections point is at a vertex of the graph of position sequences, this is evident. If not, then we have to take a slightly closer look at the situation of two yellow edges that both are spanned between vertices of the ith and $(i+1)^{\text{st}}$ line, and that intersect each other. We have to argue that the corresponding left yellow vertices in the lines with indices i and $(i+1)$ and the corresponding right yellow vertices are also connected by a yellow edge, so that the infinite yellow edge-paths can be composed as required. However, this follows from the geometry of our pictures if we recall that our construction principle required to draw any yellow edge unless it would have been intersected by a pair of white arrows. So, let us think of two crossing yellow edges and let us think of all possible positions of a pair of white arrows (recalling that in this context "pair" means that the two white arrows start at the same point). When discussing all these possibilities it straightforwardly follows that such a pair, if it is to be placed so that each of the crossing yellow lines is at most intersected by one of the white lines, then cannot obstruct the connection between the left endpoints and the right endpoints of the yellow cross. □

5.8 Remark: Recall that in 4.3 we tried to define a combinatorial description of the start- or endpoint of a reading domain by associating the system of places to the left or right of the corresponding iterated split-up letters to the corresponding parameter value. In the the dual split-up graph as defined in 5.5 such system of places corresponds to a system of yellow vertices, one in each line with a sufficiently high index. However note that we cannot guarantee that yellow vertices constructed that way can necessarily be connected by a monotonous yellow edge-path. For, in case of the third or fourth example for the permitted order of split-up letters as listed in 1.8(ii), we have some white arrow that is intersected by an entire other pair of white arrows. And in that case a parallel yellow edge would also be intersected by the same pair of white arrows, and hence according to the above construction scheme such a yellow edge is not inserted into our graph of position sequences. In D.10.5 we will associate position sequences to systems of split-up letters, but the corresponding definition will be less straightforward than our auxiliary construction 4.3 in the course of proof of Lemma 4.1.
We should keep in mind that the positions of iterated split-up letters do not straightforwardly determine position sequences.

5.9 Remark: Let l_1 and l_2 be two letters that on some level of a word sequence ω appear in a neighbouring position. Then any position sequence κ that points at the place between these two letters can only point at places to the right of the iterated left split-up letter of l_1 and to the left of the iterated right split-up letters of l_2. This is an immediate consequence of how yellow and white edges in our dual split-up picture are allowed to cross each other. Note that for commuting letters l_1 and l_2 this is the best result that could be stated in this context, since according to an iterated application of 1.8(ii) indeed all iterated left split-up letters of l_1 and all iterated right split-up letters of l_2 may show up in an adjacent position. For non-commuting letters l_1 and l_2 we

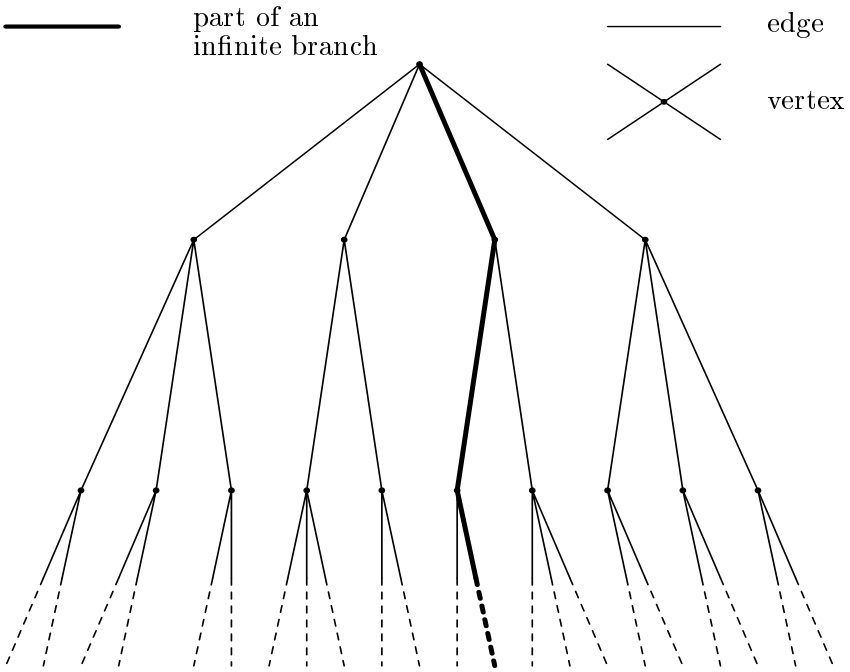


Fig. j$_1$ (belonging to 5.10):
This figure shows the upper part of a finitely branching infinite tree as an example for a graph to which the classical Königslemma can be applied.

will be able to obtain a better result, namely that there actually exists some position sequence $\widehat{\kappa}$ which separates the split-up letters of l_1 and l_2 from each other in the sense that all iterated split-up letters of l_1 and of l_2 appear to the left or to the right of $\widehat{\kappa}$, respectively. In case where the reading domains of l_1 and of l_2 are disjoint, this will come out as an immediate corollary of the reading conditions B.1.4 and 5.18(iii), since $\widehat{\kappa}$ can be defined as some position sequence that points to some parameter value between the reading domains. If the reading domains should not be disjoint (this can only happen if $l_1 = l_2^{-1}$) and the associated path performs a tangency to one of the boundary lines of l_1, the proof-techniques of 5.18(iii) have to be recycled to give this result: Since the split-up letters of l_1 and l_2 cannot mix in this case, our dual split-up picture permits to define a corridor between these split-up letters, and then from the arguments 5.22(i)–(iii) it follows that this corridor cannot be obstructed by white arrows coming from letters of the other basic direction.

5.10 Remark: *("Königslemma-type of Conclusion")*
Figure j$_1$ shows a part of a finitely branching infinite tree for which the classical Königslemma states that this tree must have an infinite branch ([Kö1].§1.A, [Kö2].§3, [Kö3].VI.§2(Satz 6)), Figure j$_2$ shows the more generalized type of graph to which we also can apply this classical Königslemma-type of conclusions. For both graphs as pictured

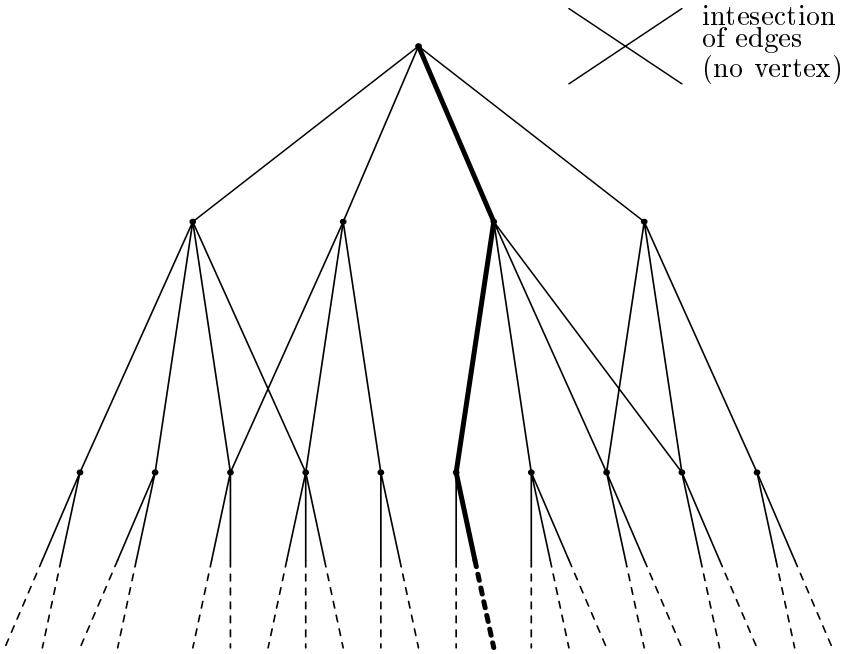


Fig. j_2 (belonging to 5.10):
This picture shows a more general type of graph than that as seen in Fig.j_1 to which also the Königslemma-type of conclusions can be applied.

in Figure j consider a subset of vertices that have been drawn on the same horizontal level as a *"line"* in this graph. The characterizing property of the Fig.j_2-graphs is given by having subdivided the set of vertices into a disjoint union of countably many lines which are denumerated by $\mathbb{N}$, so that each line only contains finitely many vertices and that each edge connects only vertices of adjacent lines. For both types of graphs these line-structures can in principle be defined by identifying the root-vertex and then measuring the length of an edge-path that connects the given vertex to the root-vertex. However, in particular for verifying that a graph that comes from a certain problem fits into the framework of Fig.j_2-graphs it is usually more convenient to use a property for defining the lines and their enumeration which is related to the underlying problem, and to verify from this defining property that all edges only connect vertices in adjacent lines.

It is an observation which is apparantly due to Denés König that if a graph as pictured in Fig.j_1 has infinitely many vertices, it has also an infinite branch: If all monotonously down-leading edge-paths emanating from the root-vertex that may end up anywhere in the graph are considered, then at least one of the edges that start at the root-vertex has to be passed over by infinitely many of those paths. One can use this edge e_1 as the first edge for our infinite branch, and then iterate this conclusion by asking which

of the finitely many downleading edges that start at the endpoint of e_1 has to be passed over infinitely many times by our infinite set of edge-paths. That way we find e_2 and can continue our construction. Obviously the same scheme of conclusion can be used to find an infinite monotonous edge-path in a Fig.j$_2$-graph.
Graphs that fit into the framework of Fig.j will be considered at many places in this paper (5.5, D.2.12, D.7.1,...). Apart from finding infinite branches in these graphs by the argument as explained above, another point of our interest should be mentioned here: In some cases (e.g. D.1.18) in this paper we will consider these graphs with a concrete embedding into the Euclidean plane, and then discuss properties whether an infinite branch can be found on all (or on an infinite subset of) lines to the left or to the right of some other infinite branch.

5.11 Remark: Recall that the concept of word sequences as discussed in [Zas2].§2 was based on an insertion structure, i.e. each letter reappeared on each higher level. Consequently any two letters appeared on almost all levels, and that way is was naturally defined in which order the two letters appeared.
For the word sequences as used in this paper, this can, of course, also be naturally defined if the letters appear on the same level of the word sequence. However, the purpose of this paragraph is to point out that there is no such natural definition if the letters L, l occur on different levels. Or, more precisely, there are several non-matching ways of how one could define such relations. One way would be given by looking at position sequences and asking, whether there exist some which pass on the upper level on a certain side of the corresponding letter L, that do pass on the lower level on the other side with respect to the letter l there. This cannot define "left/right" straightforwardly, since various combinations of these properties can occur. Another sensible way of defining left/right-relations of letters which occur on different levels of word sequences would be given by looking in what order on the lower level (some or all of) the iterated split-up letters of L occur with respect to l. Since, as had been pointed out in 5.8 already, in our dual split-up picture (where "white" stands for split-up relations and "yellow" for position sequences, roughly speaking) yellow edges cannot always be found parallel to each of the white edges, any left/right-relation that could be defined with respect to the latter philosophy would probably not match whatever could be defined according to the first one.
The situation, as exposed in the preceding paragraph makes us agree not to introduce a global concept of defining "left" or "right" when it comes to comparing the positions of letters that are placed on different levels of word sequences. If we need to compare them in certain situation, we will not be able to get around to state the corresponding properties in such difficult way as "l occurs to the left of the iterated left split-up letter of L" or "there exist a position sequence κ such that L is placed to the left of κ and l stands to the right of κ and (or: but not) the same for all position sequences that are equivalent (in the sense of 5.18(iii)) to κ".

5.12 Lemma: *Let ω and η be word sequences which result as the first and as the second part from having split up a word sequence that here will be merely denoted by*

$\omega\eta$. Assume that ω and η are valid readings from two paths u and v, respectively. Then $\omega\eta$ is a valid reading of $u \cdot v$.

Proof: By 3.1 each of the two split-up word sequences ω and η can be associated with a path u and v, respectively. From the internal compatibility 1.6) of the given word sequence $\omega\eta$, we get that $\overrightarrow{\omega} = \overleftarrow{\eta}$, and hence that the endpoint of u coincides with the beginning of v so that u and v are compatible concatenable paths. Hence they can be plugged together to give one path $\widetilde{u}$, and ω and η are valid readings for a segmentation of $\widetilde{u}$. However, in general this method of reading a word sequence (i.e. reading according to some segmentation the word sequences separately and concatenating them afterwards) may in general fail to produce a valid reading, since it could have happened that the freedom of 1.3(iii) (to ignore a partial act of crossing or to acknowledge it by a letter) might have been used for the end phase of the first segment and the beginning of the second segment in such an inconsistent way that the concatenated word sequence fails to describe $\widetilde{u}$ properly. Namely, it could have happened that an act of $\widetilde{u}$ crossing a corridor is acknowledged twice by the corresponding letter (since the corresponding corridor has been partially crossed by each of the segments) or is not acknowledged at all (for similar reasons) or that, if $\widetilde{u}$ takes a peek inside a corridor, this is acknowledged by one letter or by a cancelling pair of letters. Cf. the reading conditions B.1.4 in order to see that these are actually the only possibilities in how far the reading conditions might fail if working with the "segmentation technique". Now observe that in any of these cases what is obtained by concatenating the segmentwise read word sequences fails to satisfy the conditions for a word sequence. (Apart from the last of the above cases the internal compatibility 1.6 cannot be given, and in the last case our cancelling pair of letters cannot have its full set of split-up letters, and hence 1.7(i) cannot be fulfilled.) Since in our case we assumed that $\omega\eta$ was a valid word sequence, none of the above cases can actually happen to us and $\omega\eta$ is a valid reading for $\widetilde{u}$. □

5.13 Lemma: *Let $\kappa = (k_i)_{i\in\mathbb{N}}$ and $\kappa' = (k'_i)_{i\in\mathbb{N}}$ be two position sequences of a word sequence ω such that $(|k_i - k'_i|)_{i\in\mathbb{N}}$ gives an unbounded sequence. Then for almost all i either $k_i < k'_i$ or $k_i > k'_i$.*

Proof: This is an immediate consequence of Proposition 5.14, since, if $k_i - k'_i \geq 5$, i.e. there are at least five letters in $\omega_{i|[k'_i,k_i]}$, then at least three of them have to be of the same basic direction. □

5.14 Proposition: Let κ, κ' be as in 5.13. In addition, assume that $k'_i < k_i$ for some i and that $\omega_{i|[k'_i,k_i]}$ contains at least three letters a, b and c from the same basic direction. Then $k_{i+j} - k'_{i+j} \geq 2 + 2^j$ for all $j \in \mathbb{N}_0$.

Proof: Let j_1, j_2, j_3 be the position numbers of a, b and c respectively. Then we have

$$k'_i < j_1 < j_2 < j_3 \leq k_i.$$

By 5.4(1) we get that $k'_{i+1} < \nu^{\mathrm{r}}(j_1)$ and that $\nu^{\mathrm{l}}(j_3) \leq k_{i+1}$. On the other hand, by 5.3(i) and the fact that the letters a, b, c are non-commuting and hence only the first

case of 1.8(ii) can be given, we get that $\nu^{\mathrm{r}}(j_1) < \nu^{\mathrm{l}}(j_2) < \nu^{\mathrm{r}}(j_2) < \nu^{\mathrm{l}}(j_3)$. If we put these estimates in one line and iterate them, we finally end up with

$$k'_{i+j} < (\nu^{\mathrm{r}})^j(j_1) < (\nu^{\mathrm{l}})^j(j_2) < (\nu^{\mathrm{r}})^j(j_2) \leq (\nu^{\mathrm{l}})^j(j_3) \leq k_{i+j}.$$

The desired claim follows from this inequality and an additionally noting that the differences $(\nu^{\mathrm{r}})^j(j_2) - (\nu^{\mathrm{l}})^j(j_2)$ are growing at least as fast as the powers of two. The latter observation is due to the fact, that b, by assumption, is neither the first nor the last letter of its basic direction, and hence the exceptional rules of 1.7/1.8(i) do not apply to b. It has to have two split-up letters, and so has each of its iterated split-up letters. Hence, just due to the iterated split-up letters, we see at least the double number of letters when we pass to the next level of our word sequence and this gives the desired estimate. □

5.15 Definition: With ω as in 5.13 we establish the following partial order on $\mathbb{P}(\omega)$:
(i) If two position sequences κ and κ' satisfy that $|k_i - k'_i|$ is bounded, they are defined to be incomparable.
(ii) In case their sequence of differences is unbounded, then 5.13 provides a natural way to define whether "$\kappa < \kappa'$" or whether "$\kappa > \kappa'$".

5.16 Proposition: *For any two position sequences $(k_i)_{i\in\mathbb{N}}$ and $(k'_i)_{i\in\mathbb{N}} \in \mathbb{P}(\omega)$, $(\min\{k_i, k'_i\})_{i\in\mathbb{N}}$ and $(\max\{k_i, k'_i\})_{i\in\mathbb{N}}$ define also valid position sequences in $\mathbb{P}(\omega)$.*

Proof: It is an immediate consequence from κ and κ' satisfying 5.4(1) that this relation also holds for the min- and max-sequences. □
Remark: Note that even max- and min-constructions of arbitrary (even infinite) sets of position sequences again give valid position sequences. The case of arbitrary finite sets can be reduced by induction to the case of cardinality 2 as described above, and the case of infinite sets can be reduced to the finite sets, if one notices that the range of values for any entry of a $\mathbb{P}(\omega)$-sequence at a fixed level index or at two adjacent level indices is finite. Hence even infinite sets of position sequences can only produce finite sets of pairs (k_i, k_{i+1}), and since on the other hand the characterizing condition for a position sequence is a local one, the infinite case follows straightforwardly from the finite case. *As a convention for our notation we establish, that we will write expressions like* $\max\{\kappa_1, \kappa_2\}$ *for the above defined level-wise constructed maxima.* Note that this convention is inconsistent with the classical maximum-operator, which just picks the biggest element out of a given set; we can have $\kappa \neq \max\{\kappa, \kappa'\} \neq \kappa'$. In case of infinite sets we will usually just write $\sup\{\kappa_1, \kappa_2, \ldots\}$, since due to phenomena as pictured in D.10.4(Fig.o) we might have sets where the supremum-position sequence is bigger then any of the entries. Please note that for position sequences also suprema and infima are to be understood as level-wise defined. □

5.17 Remark: *Note that each position sequence κ of a word sequence ω uniquely describes a point in the plane which lies on the path that is associated with ω. In addition, if this path comes with a parametrization without constant segments, κ even uniquely*

corresponds to a parameter value. This is an immediate consequence of the principles discussed in 1.6 and in 5.12, in how far the internal compatibility of a word sequence determines certain other entries of the word sequence, or even of the corresponding position sequences. Hence any position sequence can be associated with a sequence of lattice squares giving a nesting of two-dimensional intervals, or, more technically speaking: By 5.1/5.4(Ad (ii)) the position sequence $\kappa \in \mathbb{P}(\omega)$ induces the split-up of ω into $\omega_{|[0^{\mathbb{N}},\kappa]}$ and into $\omega_{|[\kappa,\ell^{\mathbb{N}}]}$. Now by 5.12 the endpoint of the paths that is associated with $\omega_{|[0^{\mathbb{N}},\kappa]}$ is the same as the initial point of the path associated with $\omega_{|[\kappa,\ell^{\mathbb{N}}]}$, and hence this point can be regarded as a point that is defined by ω and by κ.

5.18 Lemma: *Let ω be a word sequence, let $u : [0,1] \to X$ be a path realizing this word sequence without constant segments and let $\kappa = (k_i)_{i\in\mathbb{N}}$ and $\kappa' = (k'_i)_{i\in\mathbb{N}}$ be position sequences of $\mathbb{P}(\omega)$.*

(i) If $\kappa < \kappa'$, then in the sense of 5.17 κ corresponds to a parameter value which is smaller than the parameter value to which κ' corresponds.

(ii) If κ and κ' are incomparable, they correspond to the same parameter value.

(iii) If we establish an equivalence relation on the set of position sequences such that incomparable position sequences are equivalent, the partial order of position sequences induces a total order on the equivalence classes, and this set bijectively corresponds to $[0,1]$, the parameter domain of u.

Proof — Ad (i): By the natural correspondence between splitting up word sequences and splitting up paths according to 5.12, from the splitting up of u according to $\omega = \omega_{|[0^{\mathbb{N}},\kappa]}\omega_{|[\kappa,\kappa']}\omega_{|[\kappa',\ell^{\mathbb{N}}]}$ into $u_{|[0,t]} \cdot u_{|[t,t']} \cdot u_{|[t',1]}$ we get that the parameter values t and t' that correspond to κ and κ', respectively, satisfy that $t \leq t'$. Then by applying 4.8(iii), i.e. the fact that $\omega_{|[\kappa,\kappa']}$ is an unbounded word sequence, we in addition get that $t \neq t'$.
Ad (ii): In case that $k_i \leq k'_i$ for all i, we can use the same line of arguments as in (i), applying 4.8(ii) instead of 4.8(iii). Since this gives that the path corresponding to $\omega_{|[\kappa,\kappa']}$ is constant, but u is assumed to be parametrized without constant segment, we get that $t = t'$. In the more difficult case where no such property like "$k_i \leq k'_i$ for all i" holds, we have to apply 5.16 for defining an auxiliary position sequence $\kappa'' := \min\{\kappa, \kappa'\}$ and to apply the above scheme of conclusion twice.
Ad (iii): "One-to-one" is now an immediate corollary from (i) and (ii); "onto" requires some graph-theoretical arguments that are given below, that come out to some kind of an inversion of what has been proven in 5.12: We have to think of a particular path u and we are given a particular way for how to read the word sequence ω from $u : [0,1] \to X$. In addition we are given the parameter value $t \in [0,1]$ and we have to ask whether, while keeping all principle decisions for how to read ω from u unchanged, just by imposing new decisions which of the reading domains should go into the first half and which into the second half, the corresponding reading of ω from u can be adjusted in such a way that ω is split up such that the first/second half of ω describe $u_{|[0,t]}$ and $u_{|[t,1]}$ respectively. The corresponding splitting up can hardly be done algorithmically; in particular in the case where one has to cope with lots of commuting letters and the hardly predictable order of their split-up letters (cf. 1.8(ii)), we always have look in

which part the corresponding split-up letters of the finer levels have to go to before we can decide how in a compatible way their root-letters can be put into the first or into the second half. In the following we will mainly use graph-theoretical conclusions for finding a position sequence κ that gives the desired split-up of ω and corresponds to the given parameter value t. These graph-theoretical conclusions are spread over various propositions and definitions in 5.19–5.23:

5.19 Definition: *Let ω be a word sequence that can be read from a path u and let t be a parameter value of u. Then with the pair (ω, t) a subgraph of marked yellow vertices can be associated in the dual split-up picture of ω according to the following rules:*
Let $\widehat{\omega}$ be the sequence of words that is derived from ω by eliminating all those letters of ω that have reading domains that contain the parameter value t (even if only as an endpoint). The fact, that $\widehat{\omega}$ need not be a word sequence any more, does not matter. Since it is derived from ω by eliminating letters, only, the yellow vertices of the dual split-up graph of ω (cf. 5.5) can clearly be projected to the yellow vertices of a similar graph-construction for $\widehat{\omega}$, and this projection will not be one-to-one in the neighbourhood of those ω-letters that do not show up in $\widehat{\omega}$ anymore. Note that we do not discuss the edge-structure of the $\widehat{\omega}$-graph, so we do not worry what happens to the edges in the course of the projection. Our interest lies in the following feature of the $\widehat{\omega}$-graph: it has on each line clearly one yellow vertex that corresponds to the parameter value t. Namely, since all $\widehat{\omega}$-letters clearly correspond to reading domains that have terminated before t or started after t, there is precisely one place between the corresponding letters that is related to t. In the following we will call these places the "exceptional yellow vertices of the $\widehat{\omega}$-graph". We then put marks on those yellow ω-vertices, that by the above map are mapped to one of the exceptional $\widehat{\omega}$-vertices. This gives the subgraph of marked yellow vertices which was to be defined.

5.20 Remark: In order proceed with the proof of Lemma 5.18(iii), we want to show that the system of the marked yellow vertices together with those yellow edges of the graph of position sequences (cf. 5.5) of ω that connect two of those marked vertices, is a graph to which the Königslemma-type of conclusions as described in 5.10 can be applied. Since this graph as a whole fits into the framework described in 5.10, and since by construction there are marked vertices on each level, only one item remains to be shown for that purpose: Namely, that for each marked vertex there is at least one upward leading edge that again ends at a marked yellow vertex. In the following we want to assume that for some two adjacent level indices i and $i+1$ this may be not given, and then prove that this assumption leads to a contradiction:
For that purpose we proceed by recalling some of the geometrical properties of the dual split-up picture of ω: As above, in each line some of the yellow vertices are marked. In addition, there need not but may exist white vertices which are to be deleted when passing to the $\widehat{\omega}$-graph. Please observe that, if there should be more, they need not necessarily be in an adjacent position.

(We could easily think of a situation where t falls into a huge reading domain for one of the basic directions which contains a lot of tinier reading domains of the other basic direction, so that t lies in the middle of one of those tinier reading domains. Then two letters are two be deleted, one corresponding to the tinier reading domain, and the other corresponding to the huge reading domain. But the letter corresponding to the huge one may have been inserted before or after those letters the correspond to the other tinier reading domains.)

However, other than for the white vertices as just described, we are able to show:

5.21 Proposition: *For the markings as defined in 5.19 we have that, if there should be more than one marked yellow vertex, they are always in an adjacent position.*

Proof: If a yellow vertex is not marked and a marked yellow vertex comes later, then by definition of our marking, a white vertex has to lie between them which corresponds to a reading domain that comes before t. Conversely, if an unmarked yellow vertex comes before a marked one, then in between can be found a white vertex that corresponds to a reading domain with a start-parameter that is bigger than t. Hence an unmarked yellow vertex cannot simultaneously satisfy the condition of coming after and before marked ones, and hence in each line the marked yellow vertices are in an adjacent position. □

The result 5.21 makes us speak of the "corridor of marked yellow vertices", and we may think that this corridor is marked by some kind of hatching in our dual graph that is associated with ω. However, such a marked subdomain might not be a connected graph, and hence we still need to prove:

5.22 Proposition: *For a marking as defined in 5.19 there does not exist any level index i where all downward directed yellow edges that start in the marked corridor end outside, and all yellow edges that end on the $(i+1)^{\text{st}}$ level inside the marked corridor come from outside.*

Proof: By 5.5, if the marked yellow vertices of the levels i and $(i+1)$ are not to be connected by yellow edges, then there must exist some pair of white arrows in such a unfortunate position, that it prevents the corresponding yellow vertices from being connected. Such a pair of white arrows connects by definition one letter of ω_i with both of its split-up letters, and this gives several possibilities where one could find t with respect to the corresponding reading domains. In the following we will discuss these possibilities separately, and we will show in each of these cases that the corresponding pair of white arrows can never give an obstruction for finding a yellow edge that remains inside the marked corridor:

(i) Assume that t is outside the reading domain of the root-letter. Hence it either comes before or after the corresponding reading domain, and since some initial/terminal subinterval of the reading domain of the root-letter must be the reading domain for its first/second split-up letter, respectively, the analogous relation holds for the reading domains of both split-up letters. Hence we know that all three of the corresponding white vertices can be found on the same side of the marked yellow corridor, and thus

these pairs of white arrows cannot give an obstruction for placing a yellow edge inside the marked corridor.

(ii) Now assume that t falls inside the reading domain of our root-letter, and between the reading domain of the two split-up letters. Hence one of these split-up letters can be found to the left and the other to the right of our marked corridor. This implies, that the arrows pointing at the corresponding white vertices can never be an obstruction for placing a yellow edge inside the marked corridor, since, regardless whether the white vertex that corresponds to the root letter is placed to the left, to the right or in the interior of the marked corridor, any yellow line that ends on the $(i+1)^{\text{st}}$ level inside the corridor cannot have intersected both of the corresponding white arrows.

(iii) Now consider the situation that t falls inside the reading domain of one of the split-up letters, (without loss of generality: inside the reading domain of the first). Hence the white vertex that corresponds to the second split-up letter has to be placed to the right of the marked corridor. As indicated above, we can hardly predict the precise place of the left split-up letter and the root letter in this case. Hence at first sight in this case there could be a situation where the corresponding pair of white arrows gives an obstruction for finding the desired yellow edge-path: Namely, if the corresponding yellow marked edge on the $(i+1)^{\text{st}}$ level is placed left of the left split-up letter, and it shall be connected to some marked vertex that can be found to the right of the root-letter. In this case we have to hope that on the ith level there also exist marked yellow vertices to the left of the root letter, since in this case the desired upward connection of our marked yellow edge would exist. Hence we now assume that such marked yellow vertices do not exist, and show, that this will give us a contradiction:
We recall our convention for marking yellow vertices and ask: "What has to be given if a marked yellow vertex cannot be found before some letter L whose reading domain contains t?" – According to our marking condition this can only be given if the reading domain for L has an overlap with the reading domain for some other letter l and t comes after the reading domain of l, and if the ith word of the corresponding word sequence contains "L" before "l". As a consequence of this set-up the white vertices that correspond to l or to any of its split-letters come before the marked corridor. By 1.8(ii) the first split-up letter of L has to be placed before the second split-up letter of l, and hence this letter must also be placed to the left of the marked yellow corridor. But this contradicts to our set-up, since we have assumed that there exist marked yellow vertices to the left of the left split-up letter of L, and we now see that there are none.

As a result of the discussion of the preceding three items, we see there cannot be pairs of white arrows which prevent us from connecting any marked yellow vertex to some marked yellow vertex in the preceding level. This completes the proof of this proposition. □

5.23 Completion of the Proof of Lemma 5.18:
By 5.20–5.22 the subgraph of the graph of position sequences that in the sense of Definition 5.19 is given by the marked yellow vertices and the edges that connect two of these vertices satisfies all conditions for applying the Königslemma type of conclusions

as described in 5.10. Hence this subgraph has an infinite branch which describes a position sequence κ.
Now we can end the proof of 5.18 by showing that this position sequence κ corresponds in the sense of 5.17 to t. For that purpose we have to show that $\omega_{|[0^{\mathbb{N}},\kappa]}$ and $\omega_{|[\kappa,\ell^{\mathbb{N}}]}$ are valid readings of $u_{|[0,t]}$ or $u_{|[t,1]}$, respectively. However, this is straightforward from our construction: Recall that at the beginning of this lengthy proof we had pointed out that all it takes to show the desired surjectivity is to find a criterion whether an essential domain that contains t is to be treated as an essential domain in the reading of the second half and as an inessential domain in the first half, or vice versa. By constructing κ we finally found this criterion: We treat our reading domain in that part as essential that contains the corresponding letter when ω is split up at κ. Since κ was by graph-theoretical means constructed as a well-defined element of $\mathbb{P}(\omega)$, we know that $\omega_{|[0^{\mathbb{N}},\kappa]}$ and $\omega_{|[\kappa,\ell^{\mathbb{N}}]}$ satisfy all conditions for word sequences. By using that information on the one hand and, apart from that, the same reading conventions for $u_{|[0,t]}$ and $u_{|[t,1]}$ as when reading ω, as a third, the above guideline for how to treat those reading domains that contain t, it is straightforward to verify that indeed $\omega_{|[0^{\mathbb{N}},\kappa]}$ and $\omega_{|[\kappa,\ell^{\mathbb{N}}]}$ are valid readings from $u_{|[0,t]}$ and $u_{|[t,1]}$, respectively. Hence κ is a preimage of t, and this completes the proof the desired surjectivity. □

5.24 Definition: The *"inverse of a position sequence* $\kappa = (k_i)_{i\in\mathbb{N}} \in \mathbb{P}(\omega)$" is defined as $\kappa^{-1} := (\ell(\omega_i) - k_i)_{i\in\mathbb{N}} \in \mathbb{P}(\omega^{-1})$. This definition gives that the places in ω and in ω^{-1} to which κ and κ^{-1} point, respectively, correspond to each other with respect to the inversion process (cf. 4.9).

Abschnitt D: Proofs of Section §B.2

Recall that the main purpose of this section is proving Theorems B.2.10 and B.2.12, i.e. to prove that the equivalence relation that via $\Phi = \Psi^{-1}$ is induced on the set of all word sequences from the relative homotopy of paths permits to define almost unique reduced representatives within each homotopy class. By B.3.1 this enables us to select an abstract homotopy combing. Part E will then be devoted to showing that this homotopy combing can be lifted to a continuous parametrized homotopy combing and this will prove Theorem A.5.6 and hence by A.6.5 complete the proof of our Main Theorem.

§1. Addenda to the combinatorial definitions of our equivalence relations

The first step in order to follow the above program rests on combinatorially describing

the equivalence relations that are induced on $\mathcal{A}^{**}$ by the relative homotopy on the one hand and by the non-injectivity of Φ on the other. This will be done by stating the combinatorial definitions of the corresponding relations and by later proving that these combinatorial definitions indeed describe the desired relations. For the sake of being able to precisely explain the structure of the paper at an early state already, the essential parts of these definitions have already been given in Section §B.2. Hence the following paragraphs mainly consist of giving those comments that for the sake of brevity in the early parts of this paper have not been included there.

1.1 Geometric motivation for the elementary processes of our equivalence relations:

(i) For *elementary cancellations and insertions:*

By B.2.2(i), if two inverse letters follow each other in a word sequence, i.e. two letters indicating acts of crossing the same corridor with inverse direction, they may be cancelled. Conversely, such a pair of two inverse letters may be inserted into a word, provided it is compatible with the demands C.1.6. The corresponding geometric picture is that of a path which does cross a corridor, then turns around, and crosses it again in the inverse direction. And in such a case it might seem possible to pull the corresponding segment of a path into such a position where it does not touch the corresponding corridor any more, or at least where it does not perform a completed act of crossing any more. On the other hand, if we are given such a picture, some of the holes cropping up later might one prevent from pulling the path into its more straight position (cf. A.3.4).

(ii) For *elementary transpositions:*

By B.2.2(ii), if two letters follow each other within a word and if these letters had been assigned the attribute "commuting" by B.1.8, this process interchanges these letters. The corresponding geometric picture is the following: If the direct line $\overline{P\ Q}$ would diagonally go through a crossing region of two corridors A and B and the corresponding path joining P and Q passes along this region $A \cap B$ on its one side, it may be possible to move our path so as to pass along $A \cap B$ on the other side. This makes our path intersect the corridors A and B in reversed order. However, even we we know $A \cap B$ to be free of punctures, the domain between the two traces of our path and $\partial(A \cap B)$ may have holes cropping up later, and these holes then would prevent us from performing a corresponding homotopy. See 3.10(Fig.k) for a figure containing such path moves.

(iii) For *elementary truncations or appendances:*

By B.2.2(iii), if we perform one of these processes to a given word, then we either remove the first or the last letter from this word, or we append an additional letter at the beginning or at the end of this word, provided it is one of the letters that could be added at this place by rule C.1.6. Of course a careless use of this operation would result in performing such a (non-relative) homotopy which moves the endpoint of a path and hence cuts off the initial segment or the end segment of path, or it enlarges a path by drawing an additional segment. Note that a description of this operation for triple sequences (cf. C.1.4/C.1.9) has to include a readjusting of the first or of the third entries of the corresponding triples. However, by C.1.3(iii) we cannot

completely forget about this operation, but we will have to restrict it to those cases where it is only used to readjust the given word of a word sequence to the situation where we decide to alter the description for its initial point or for its endpoint by using a different sequence of square lattices. We cannot give a geometric picture of this process, because we will essentially only use it to skip between various permitted readings of the same path.

1.2 Addendum to B.2.1(ii): Recall that it was the purpose of the *compatibility demand* B.1.2(ii) to make sure that, as explained in 1.1(i)&(ii), the processes B.2.2(i) and B.2.2(ii) cannot be applied to a word sequence, if the corresponding move of a path is obstructed by some of the holes cropping up later in the corresponding region. For this purpose we have defined in B.2.9 that two word sequences are only equivalent if such processes can be *"synchronously"* applied to transform the corresponding words of any index into each other. Recall (C.1.7) that any word of a word sequence is more or less determined by its successors. Analogously, we will prove in 1.9 that any of the processes B.2.2(i), B.2.2(ii) and B.2.2(iii) induces more or less for the corresponding preceding words of a word sequences either the identity or a similar process. Accordingly we will define that a system of elementary transformations that for each $i \in \mathbb{N}$ transforms the ith word of a word sequence ω into the ith word of a word sequence η is only *"synchronizable"*, if the sequence of transformations for the index i is induced by the sequence of transformations that is prescribed for the $(i+1)^{\text{st}}$ word. Conversely, such sequences of transformations for words with a fixed index i are called *"non-extendible"*, if for some higher index j a system of transformations cannot be found that transforms the corresponding word and induces the transformations for the index i. The two geometric pictures of 1.1(i) and 1.1(ii), where holes cropping up later obstruct a certain homotopy correspond to those non-extendible processes: If a path, when crossing a certain corridor and coming back later runs around some hole cropping up later within this corridor, it will intersect essentially different split-up corridors on its way back than on the outward direction, and these split-up corridors correspond to different variables which cannot be cancelled against each other. That way the cancelling of variables that correspond to the original corridor could not be extended. Similarly, if we want to move a path to the other side of a crossing region of corridors, and, if in the course of this process we had to move it to the other side of a small hole cropping up later outside the crossing region, the four corridors emanating from this hole belong to four different variables, and hence the corresponding move on that level would imply a complete exchange of variables, not just an interchange between two commuting ones. Since these parallel corridors are on fixed prescribed geometric position ("left of" our "right of", "above" or "below") with respect to the crossing ones, the interchange of the variables belonging to the crossing pair of corridors could be not extended throughout the word sequences.

The next paragraph is devoted to give a more precise description of transformation diagrams as it was possible in B.2.5:

1.3 Addendum to Definition B.2.5: Transformation diagrams have been introduced in B.2.5 in order to visualize transformation processes and to ease the definition of some of their properties:

Each transformation diagram consists of a frame of two vertical and of two horizontal bars where the upper and the lower bar is annotated by the two words to be transformed into each other. This annotation by words defines a certain number of fixed "positions" on the upper and on the lower bar, each letter of the word corresponds to one position. The basic idea of this diagram is that the position of each letter of a word shall be connected with the position of the corresponding "partner letter", where apart from truncations and appendances there always is some natural partner. B.2.5(Fig.h_1)–Fig.h_7 show the corresponding diagrams for the following elementary transformations: elementary cancellation, elementary insertion, elementary transposition, truncation of the first letter, appendance of the first letter, truncation of the last letter, appendance of the last letter. Note that the lines corresponding to appendances and to truncations, i.e. those processes for which there is not any partner-letter, end at some suitable place of the corresponding vertical bar. This type of diagrams will also be used to visualize a sequence of such transformations, such a diagram can then be obtained by putting the corresponding elementary diagrams over each other (similarly as when multiplying braid-diagrams), while removing the separating vertical bars and the corresponding annotations of the intermediary steps. B.2.5(Fig.h_8) gives an example for such a diagram.

1.4 Remark to the Definition B.2.6(i) of the nesting depth:
Note that in particular two properties of a transformation diagram are able to bring us fairly big nesting depths. This is on the one hand, if there should be lines going from the top bar to the bottom bar which intersect with all their neighbouring lines that are associated with a variable of the other direction, since such intersections cause that a lot of different edges have to be used to make the way (even in the dual picture); and on the other hand it helps to get big nesting depths, if a lot of cancellation lines or truncation lines are in a classical way nested, so that it takes a lot of edges to emanate from this nest if one starts at one of the innermost points. This explains the origin of the name "Nesting Depth".

1.5 Remark to the Definition B.2.6(ii) of the truncation depth:
While the nesting depth is a concept which includes all the phenomena coming from truncations, cancellations and from transpositions, we also needed a concept which gives the "nesting depth with respect to the truncation lines only". For that purpose in B.2.6(ii) we defined the *"truncation depth"* by counting for each of the truncation lines how many truncation lines are (more precisely: have segments that are) contained in the interior of the cell that in the transformation diagram is bounded by the corresponding truncation line and the corresponding segment of the outer frame, and by taking the maximum of these numbers over all truncation lines in our diagram we get the corresponding quantity. In that context observe that by combining our elementary pictures from B.2.5(Fig.h_1)–B.2.5(Fig.h_7) we can also get truncation lines that connect the two vertical bars of the with each other. These lines correspond to letters which are appended at the beginning of our word and are later truncated to the end. These lines are also to be taken into account when truncation lines are counted, in this case the corresponding associated cell is that one of the two cells into which the truncation line splits our diagram that contains fewer truncation lines.

1.6 Addendum to the Boundedness Conditions B.2.1(iii):
Note that if our path performs a certain act of crossing a corridor entirely, it also crosses the two split-up corridors, and the 2·2 split-up corridors of those corridors, and the same for the 2^i split-up corridors of the ith level below. Hence the number of corridor-crossings performed to the overall network is unbounded. For similar reasons the moving of a path to the other side of a crossing region of corridors $\mathcal{N}_i$ implies that an exponentially increasing number of interchanges of adjacent variables for the higher indexed lattices has to be performed. We will use the phenomenon that these numbers tend to be unbounded for the overall network to characterize the difference between our weak equivalence relation and the strong one.
Consequently our definition for when two word sequences ω and η can be considered as strongly equivalent will in particular require $(n_i)_{i\in\mathbb{N}}$ is a bounded sequence, where n_i is the nesting depth of the transformation processes that brings the ith word of ω into the ith word of η.
Apart from that, since by truncations and appendances we are not allowed to perform real movements (cf. 1.1(iii)), but these processes only come in by the necessity to include all phenomena of the strong equivalence relation into the weak one, we in addition have to require that the sequence of the truncation depths is bounded, even if ω and η shall be only weakly equivalent.

1.7 Addendum to Definition B.2.2:
The purpose of this addendum is to point out that Definition B.2.2/1.1 is to be understood in such a way that (far-distance) insertions and (far-distance) appendances are only allowed as long as they do not destroy the internal compatibility. For the remaining operations (transpositions, (far-distance) cancellations and -truncations we did not explicitly require this condition, since evidently none of these processes can mess up the internal compatibility (cf. C.1.6) of a word.

1.8 Addendum to Definition B.2.7:
Recall our Definition of compatible transformations as given in B.2.7. Note that the corresponding conditions B.2.7(i)–B.2.7(ii) can never be satisfied if ω_i differs from η_i by an elementary cancellation or insertion and if $\omega_{i+1} = \eta_{i+1}$; this situation cannot be given for a compatible pair of transformations, since the exceptional rules C.1.7(ii)/C.1.7(iii) cannot apply to the middle two of the four split-up letters of a cancelling pair, and hence cancelling of letters on the ith level needs a corresponding adjustment on the $(i+1)^{\text{st}}$ level, also.

Roughly speaking the following lemma states that each sequence of transformations that transforms the ith words of two word sequences into each other can be "upward extended" to the less fine levels. This is the first of three similar observations (3.13, 6.2) which become essential when Königslemma-type arguments (cf. C.5.10) have to be applied to this situation.
1.9 Lemma: Let ω be a word sequence and for $j \in \{1, 2, 3, \ldots, i\}$ let η_j be a word in $\overline{\mathcal{A}_j}$ such that η_i differs from ω_i just by one of the elementary operations B.2.2(i)–B.2.2(v). The remaining η_j may be any words such that the finite sequence $(\eta_1, \eta_2, \eta_3, \ldots, \eta_i)$

satisfies all compatibility conditions C.1.6–C.1.7(iii) of an initial substring of a word sequence. Then there exist $k \in \mathbb{N}$ and words $(\eta_{j,m})_{1\leq j\leq i, 1\leq m<k}$ (in addition let $\eta_{j,0} := \omega_j$ and $\eta_{j,k} := \eta_k$) such that for all $j \in \{2,3,4,\ldots,i\}$ and for $m \in \{1,2,\ldots,k-1\}$ each (2×2)-submatrix of adjacent elements of the matrix

$$\widehat{\Omega} = \begin{pmatrix} \eta_{1,0} & \cdots & \eta_{1,k} \\ \vdots & & \vdots \\ \eta_{i,0} & \cdots & \eta_{i,k} \end{pmatrix} \tag{1}$$

describes compatible elementary transformations in the sense of B.2.7.

The proof of this lemma is lengthy and hence split into the paragraphs 1.10–1.13.

1.10 Basic reduction Steps for the proof of Lemma 1.9:
Note that Lemma 1.9 will be applied in many situations, where either $(\eta_1,\ldots,\eta_{i-1})$ or $(\omega_1,\ldots,\omega_i)$ or both of these tuples are not given, but are to be constructed. It is an immediate consequence of the possibility to read the lower indexed words from the higher indexed ones (cf. C.1.7(i)) that in any case a corresponding tuple of words can be constructed and by this a situation to which the statement of the lemma can be applied has been constructed. However note that, if we had decided to state this lemma without giving $(\eta_1,\ldots,\eta_{i-1})$ but by claiming their existence instead, then we would have got a weaker statement. For, in that case the $\eta_1,\ldots,\eta_{i-1}$ could have be constructed by imposing some particular relation to $\omega_1,\ldots,\omega_{i-1}$, and this situation might be easier to handle than that where one has to cope with whatever is given as $\eta_1,\ldots,\eta_{i-1}$.
On the other hand, the proof of Lemma 1.9 below is written up in precisely such a way, as if just this "weaker statement" is to be proved. This has been done, since it considerably eases the language of the inductive process which is at the heart of the proof. The purpose of this paragraph is now to point out that and why actually the even stronger statement is implied by the arguments as they are given below:
The proof of 1.9 will essentially be based on filling up the top right corner in a (2×2)-transformation matrix where the other entries are given, and this missing entry is constructed as a reading from a certain path. This will be essentially the only condition for that reading, we do not take any further care to what the neighbouring entries are. Now by C.1.5 the entries of such a tuple as $(\eta_1,\ldots,\eta_i)$ also satisfy, that each of them has be a possible reading of any path realizing any of the successors, and hence none of the arguments as given in the proof of 1.9 below is overturned if in the crucial situation we just assume that the given η_j are to be chosen as our readings. Thus the "hard" statements follows from essentially the same arguments as the weak one, to which the language of the proof below is adapted:

1.11 On a possible combinatorial scheme for a proof of Lemma 1.9:
In principle what has to be proved for Lemma 1.9 is that if three of the four entries of a (2×2)-submatrix Ω of $\widehat{\Omega}$ describing a compatible transformation pattern are given, and if the missing entry is in the top row of this matrix (without loss of generality let us assume that it is the top right entry) then there exists at least one word which could be placed there and either gives a valid (2×2)-matrix or gives such a (2×2)-matrix that by

the corresponding doubling of the left or right bottom entry and by filling up the first row correspondingly with other suitable entries gives a $(2 \times k')$-matrix such that each (2×2)-submatrix describes a compatible transformation in the sense of B.2.7. When iteratively applying this principle, all entries $\eta_{j,m}$ for the matrix $\widehat{\Omega}$ can be constructed by corresponding doubling processes and as first-line entries of such $(2 \times k')$-matrices. In the course of such a process we usually proceed in each line from the left to the right, and when the corresponding line is filled up, we continue by filling up the line above correspondingly from the left to the right.

(Note that such a result could not have been achieved if the corresponding missing entry was in the bottom row of the (2×2)-transformation matrix, due to the phenomenon of non-extendible transformations as they have already been described in 1.1(i)–(ii)).

Hence in the remainder of the proof of Lemma 1.9 we only have to point out that in any case where one the one hand the three given entries of a given (2×2)-matrix Ω satisfy the internal compatibility C.1.6, and we in addition know that the two entries of the bottom row just differ by one elementary operation and the two entries of the left column satisfy the compatibility condition C.1.7, then some entry for the top right corner satisfying C.1.6 can be constructed out of the given data so that C.1.7 also holds for the right column and that the two entries in the top line either

(1.) also differ by at most one elementary process from the list B.2.2(i)–B.2.2(v), and such a process may only involve the root-letter of the process of the bottom line (cf. B.2.7(iii)). In this case we would be done.

(2.) However note that Lemma 1.9 is stated in such a way that we are also done when only proving a weaker condition, namely that the two top-entries can be transformed into each other by a sequence of k transformations that can in the sense of B.2.7 be synchronized with $(k-1)$ identical ones and with the given transformation of the bottom line.

Such a proof might be performed in the following by essentially looking at each possibilities for such a non-identical process on the bottom line separately. However, in this case we want to avoid the nasty discussion of lots of cases; instead of this we will give a quicker proof for 1.9 modulo 1.10 by geometrical arguments:

1.12 Analysis of the geometric situation of two paths that correspond to *words that just differ by one elementary transformation:*

We denote the four entries of Ω (cf. 1.11) by

$$\Omega = \begin{pmatrix} \eta_j & \eta'_j \\ \eta_{j+1} & \eta'_{j+1} \end{pmatrix}, \tag{1}$$

and for our two entries η_{i+1} and η'_{i+1} of the bottom line we construct by the same scheme as in C.3.2 realizing paths u_j and u'_j in X_{j+1}.

According to the final remark of C.1.7 the conditions C.1.7(i)–C.1.7(iii) can be understood as an encoding of what might be the possible reading of the ith word of a path that has given the $(i+1)^{\text{st}}$ reading (up to the precise order of letters), hence η_j must be a possible reading of u_j. We then define η'_j as a possible reading of u'_j that is adjusted in

such a way that $\overleftarrow{\omega}_j := \overleftarrow{\eta}_j$ and that $\overrightarrow{\omega}_j := \overrightarrow{\eta}_j$, unless the adjustment of the corresponding start-point or endpoint of our path has been moved by some truncations or appendances in the preceding string of transformations. Observe that unlike the general case, the problem of mismatching orders are irrelevant here, since our paths u_j and u'_j are particular nice paths that have been constructed according to the scheme of C.3.2. Since those paths can never go through a crossing region of $\mathcal{N}_{i+1}$, and hence they can only through a crossing region of $\mathcal{N}_i$ if it is clear from the order of $\mathcal{N}_{i+1}$-letters that they have actually entered that crossing region. Hence in any case where u_j or u'_j is in an $\mathcal{N}_j$-crossing region of corridors and performs two acts of crossing $\mathcal{N}_j$-corridors simultaneously, the corresponding $\overline{\mathcal{A}_{j+1}}$-letters come in such an order that the order of their root-letters is free according to C.1.3(ii). As already noted in C.2.3(Ad C.1.5(ii)), the internal compatibility is a similar natural encoding of geometrical into combinatorial demands. Hence all the conditions C.1.6–C.1.7(iii) are satisfied for each of the η'_j and each pair (η'_j, η'_{j+1}).

We now look at the geometrical pictures of u_j and u'_j. Since they have been constructed as realizations of words that just differ by one elementary process, the paths have to coincide up to one place, where we see one the following phenomena:

(i) One of our path has a thorn at which it crosses a corridor of $\mathcal{N}_{i+1}$ twice in inverse directions, while the other paths skips that thorn (this picture corresponds to an elementary cancellation).

(ii) The two paths fall at some segment apart so as to pass a crossing region of $\mathcal{N}_{i+1}$-corridors on different sides (this corresponds to a transposition of variables).

(iii) At the end or at the beginning one of our paths performs some act of crossing an $\mathcal{N}_i$-corridor, while the other path terminates before reaching the corresponding corridor or starts at the other side, respectively (this corresponds to truncations).

(iv) Apart from the above, our two paths might also fall apart for an entire segment, where one of them performs only movements into one of our basic directions and hence only moves parallel to a certain line of $\mathcal{N}_{i+1}$-corridors in the corresponding segment, while the other path first crosses the corresponding line of corridors, then performs the same movements as the first path on the other side of it, and finally joins the first path again by crossing the corresponding line in the opposite direction. If the enclosed segment of the corresponding line of corridors does not have holes that are integrated into $\mathcal{N}_{i+1}$, this is the picture of a far-distance cancellation.

(v) Finally, we might also see a similar picture as in the preceding item, just differing by the phenomenon that the second path ends before its final act of crossing the line of corridor under consideration. In this case that first path has to end at the corresponding place on the other side of our line of corridors, and in this case we have the picture of a far-distance truncation to the end. Similarly, our paths might also start at corresponding places of opposite sides of a line of corridors, then only perform analogous movements parallel to this line, before one of our paths crosses this line and henceforth coincides with the other. This gives the picture of a far-distance truncation to the beginning.

1.13 Completion of the Proof of Lemma 1.9:
Having developed the pictures 1.12(i)–(v) showing how u_j and u'_j might differ and having in 1.10–1.11 reduced the task of proving 1.9 essentially to one of those situations, we can now complete the proof of 1.9 by asking in how far this difference might affect the reading that these paths provide with respect to any corridor lattice with an index j that is smaller than or equal to i. Note that in many cases they need not differ at all, i.e. if in the situation of 1.12(i) the thorn does not intersect a corridor boundary of $\mathcal{N}_j$ or does only provide a peek into an $\mathcal{N}_j$-corridor without causing that the corridor is crossed, then the withdrawal of the thorn does not affect the reading of our path. However, it could also happen that the segment provided by the thorn is the crucial segment that completes an act of crossing of an $\mathcal{N}_j$-corridor, and hence the withdrawal of the thorn undoes this act of crossing. In this case the $\mathcal{N}_j$-readings differ by a corresponding cancellation also. Similarly, in 1.12(ii) the moving of our path to the other side of a crossing region of corridors might take place outside of all crossing regions of $\mathcal{N}_j$-corridors, or in such a way that, even if u_j is moved into the u'_j-position, both paths still pass through the same $\mathcal{N}_j$-crossing region of corridors. In these cases an adaptation of the $\mathcal{N}_j$-readings need not take place. However, if u_j is moved to the u'_j-position, it might just be moved out of an $\mathcal{N}_j$-crossing region of corridors, and if our paths then should both cross the corresponding $\mathcal{N}_j$-corridors, the order of the associated letters of $\overline{\mathcal{A}_j}$ is free for u_j, but not free for u'_j. In the unlucky case of the latter situation we hence have to perform a transposition, in order to turn the jth u_j-reading into an u'_j-reading. The remaining cases 1.12(iii)–1.12(v) give similar phenomena. In each of those cases it is not hard to see that on the jth level fortunate readings of u_j and of u'_j never differ (as required in B.2.7(iii)) by more than one elementary process that is of the same type as the one that turns ω_{i+1} into η_{i+1} and and only affects the corresponding root-letter.
However, we cannot assume fortunate readings, since that words $\omega_1,\ldots,\omega_i$ had been given and the words $\eta_1,\ldots,\eta_i$ had been constructed without paying attention to $\omega_1,\ldots$ $\ldots,\omega_i$. Hence we might need more than one elementary transformation to turn η_j into η'_j, because we cannot assume (cf. 1.14) that η_j is a reading that is fortunate with respect to the behaviour of u'_j.
However, note that in any case such an adjustment of the readings will only take transpositions, never any of the other operations. For, the approximating paths u_j and u'_j have been constructed in such a way that they never perform tangencies, and hence we cannot see phenomena as in C.1.3(i). On the other hand, as already stressed in C.2.2, our reading mechanism is strong enough so as to cope with any valid adjustment of the first or third entries of our triple sequence, and hence it is not necessary to adjust our readings in such a way. Now, if it should be necessary (as in Example 1.14) to perform additional transpositions in order to turn η_j into some valid value of η'_j, then either u_j or u'_j has to go through the corresponding crossing region of corridors. Consequently, either for η_{j+1} or for η'_{j+1} the corresponding split-up letters have to come in such an order, that they indicate the entering of our path into the crossing region of corridors. In that situation the corresponding $\overline{\mathcal{A}_j}$-letters may be exchanged without having to similarly transpose the corresponding $\overline{\mathcal{A}_{j+1}}$-letters also. Hence we can meet the list of

demands 1.11(1.)–1.11(2.) as stated at the beginning of the proof of Lemma 1.9. This completes the proof of this lemma. □

1.14 Example for a situation of non-fortunate readings as it is indicated in 1.13: Note that, even if we would have constructed $\eta_1, \ldots, \eta_i$ with some "paying attention" to $\omega_1, \ldots, \omega_i$, such an according adaptation of our readings could not have been completely avoided, either: Assume that ω_i starts with "$LL' \ldots$", where L and L' are commuting variables. Use l_1, l_2, l_1' and l_2' to denote their split-up letters, and further assume that all these split-up letters commute with each other. Then ω_{i+1} could start with "$l_1'\, l_2'\, l_2 \ldots$". This makes l_1 a letter which is appendable at the beginning of ω_{i+1}. In the course of a far-distance appendance it could even be inserted after the two l_ν'-letters, and hence η_{i+1} could read "$l_1'\, l_2'\, l_1\, l_2 \ldots$". In this case the rule C.1.6 implies that η_i has to start with "$L'\, L \ldots$", i.e. we have to reverse the order of these two letters with respect to ω_i. And this is an operation, which can only be understood as an adaptation of the reading, not as a process that is induced by an appendance to the beginning.

1.15 Remark: The purpose of this remark is to explicitly state a fact which, since the method of the above proof 1.9 was geometric, was only implicitly used there, namely:

"When two split-up letters commute, so do the corresponding root letters"

(cf. B.1.8). This observation is an immediate consequence of the fact that by B.1.2(v) the corridors that are associated with the root letters either meet each other in a crossing region of corridors, or only cover regions that are disjoint up to at most one corner-point; and in the latter case any corresponding pair of split-up letters can only cover (such almost) disjoint regions, also.

By now, the definition of our weak and strong equivalence relations as given B.2.9 are fully motivated. However, there is one additional point which has to be stressed:

1.16 Addendum to the definition B.2.9:

The purpose of this addendum is to clearly point out that the precise information which of the letters of a word sequence are affected by a transformation process as defined in the Definitions B.2.8/B.2.9 must be regarded as part of the corresponding transformation process. This has to be explicitly stated, since the notation as chosen in Definition B.2.8 and as it will be used in the remainder of this paper, does not always provide this information. In many cases it does, since by comparing the adjacent entries of a tuple of words one is usually able to identify the letters that have been inserted, deleted or moved. However, there are exceptions, e.g. as follows:

If a word reads "$abb^{-1}bc$" and if the neighbouring word in the corresponding tuple is just "abc", both of the letters "b" could have been been the one that is cancelled with respect to b^{-1}. Observe that if in such case one translates the combinatorial data into a homotopy (cf. 1.1(i)), one gets a different picture according to which of letters was the cancelling partner and which survived. For that purpose it is essential that we regard such information as included wherever we use the expression "transformation pattern", even if we will continue to work with such tuples of words as our notation that do not always provide this information. We could have avoided this ambiguity of notation by using parenthesized tuples of words instead of ordinary tuples in B.2.8,

so that the parenthesizing then could be used to clearly mark which letters are to be effected by the corresponding elementary process. This would have made several other formulae in this paper more verbose, and so, for the sake of keeping our notation simple we decided to avoid this. Please note, that the ambiguity is only in the notation, not in the mathematics!

1.17 Remark: In the following we want to give an idea of how one could try to compute from B.2.8 an overall picture of how the elementary processes that transform our word sequence ω into η can by synchronized. One way how one could obtain such a picture is to give a sequence $(i \times l''_{i-1})$ matrices of words that can be inductively build up as follows: The first of these matrices is

$$\begin{pmatrix} \omega'_{1,1} \ldots\ldots \omega'_{1,l'_1} \\ \omega''_{2,1} \ldots\ldots \omega''_{2,l'_1} \end{pmatrix} \tag{1}$$

i.e. what is obtained from synchronizing the transformation processes for the indices 1 and 2 as explained above in B.2.8 (note that $l'_1 = l''_1$). The next matrix is a $(3 \times l''_2)$-matrix which is obtained by applying the synchronization scheme to $(\omega_{3,1}, \ldots, \omega_{3,l_3})$ on the one hand and to the columns of the matrix (1) on the other hand. The corresponding doubling processes to be performed for that purpose are in principle the same as when synchronizing $(\omega_{2,1}, \ldots, \omega_{2,l_2})$ to $(\omega_{3,1}, \ldots, \omega_{3,l_3})$, but with taking into account that instead of $(\omega_{2,1}, \ldots, \omega_{2,l_2})$ we now have to look at $(\omega''_{2,1}, \ldots, \omega''_{2,l'_1})$ since this is the bottom line of the matrix to be synchronized. Recall that $(\omega''_{2,1}, \ldots, \omega''_{2,l'_1})$ arose by doubling some of the entries of $(\omega_{2,1}, \ldots, \omega_{2,l_2})$. More precisely, in order to obtain the desired $(3 \times l''_2)$-matrix, we have to double some of the columns of the matrix (1) on the one hand, and some of the columns of

$$\begin{pmatrix} \omega'_{2,1}, \ldots, \omega'_{2,l'_2} \\ \omega''_{3,1}, \ldots, \omega''_{3,l'_2} \end{pmatrix} \tag{2}$$

on the other hand till the second line of the first matrix coincides with the first line of the second matrix, and then we have to gather together these three rows in a $(3 \times l''_2)$-matrix. By iterating this principle an $(i+1 \times l''_i)$-matrix can be build up by doubling some of the entries of $(\omega_{i+1,1}, \ldots, \omega_{i+1,l_{i+1}})$ and some of the columns of the corresponding $(i \times l''_{i-1})$-matrix, as when synchronizing them as in B.2.8. In any case the resulting matrix satisfies that each (2×2)-submatrix of adjacent elements describes a compatible transformation process in the sense of B.2.7.
We have developed this picture here because such a sequence of matrices with the corresponding condition on each (2×2)-submatrix and on how the ith matrix is embedded into the $(i+1)^{\text{st}}$ could be used as a description for an overall information on the order of steps that are to be performed if one wants to transform ω into η. Since this concept of an infinite sequence of matrices appears fairly hard to work with, we develop in the following an alternative graph-theoretical concept for describing the overall information of the steps that it takes to transform to equivalent word sequences into each other:

1.18 Definition of the transformation graph:
Let ω and ω' be two equivalent word sequences as in B.2.9, and assume that we are given

tuples for how to transform ω_i into ω'_i that satisfy all properties as listed in B.2.8. Recall that B.2.8 requires that the transformation steps of adjacent indices are synchronizable; however, there is not any requirement that for the index i the synchronization of the tuples associated with the indices $i-1$ and i behaves compatible with the respect to the synchronization of the tuples that are associated with the indices i and $i+1$. In 1.17 we already described some method how this phenomenon could be appropriately treated, and by using graph-theoretical methods we develop below an alternative picture for describing what happens if ω is transformed into ω' that will be called *"transformation graph"*; do not mix up this graph-construction with the "transformation diagram" which has been introduced in B.2.5/1.3 (or with the split-up diagrams or the dual split-up picture as introduced in C.3.4 and C.5.5, respectively). Given ω and ω' and all the basic transformation tuples as described in B.2.8, the corresponding transformation graph is constructed by performing the following steps:

(i) Subdivide that area into which the graph shall be painted into countable many horizontal lines that are enumerated by the naturals, the line number "1" being associated with the topmost line (cf. C.5.10). Write the elements of the tuples $(\omega_{i,1},\dots,\omega_{i,l_i})$ in their natural order into the ith line. At a later state of this construction we will associate each of these entries with a correspondingly labelled vertex of our graph. However, it will take the subsequent steps of essentially doubling the entries in order to reach a situation where we can describe what are the edges of the graph to be constructed. Note that the doubling of our entries will be undone later.

(ii) Apart from the first line each line is doubled, i.e. the entries that have been written before into the ith line are now written as well into the $(2\cdot(i-1))^{\text{st}}$ line $(i>1)$, and into the $(2\cdot(i-1)+1)^{\text{st}}$ line $(i\geq 1)$.

(iii) Now synchronize each tuple in a line with an odd index with the corresponding tuple in the next line whose index must be even. Hence according to the notation of B.2.8 for odd i the tuple $(\omega_{i,1},\dots,\omega_{i,l_i})$ is replaced by $(\omega'_{i,1},\dots,\omega'_{i,l'_i})$ and for even i the same tuple is replaced by $(\omega''_{i,1},\dots,\omega''_{i,l'_i})$.

(iv) Note that according to (ii) we now have that the number of entries in each line with an odd line number coincides with the according number of entries in the line below. We now insert edges into our picture so as to connect the jth entry in the ith line with the jth entry in the $(i+1)^{\text{st}}$ line for all j and for all odd i. Hence at the present situation all edges appear vertical, and they only connect entries in adjacent lines. In addition, it depends on whether the corresponding line number is even ore odd, whether all edges emanating from that line lead to the line above or to the line below, respectively.

(v) Now we undo Step (iii), i.e. we again replace the tuples obtained by having synchronized the original tuple by those ones as adjusted in the first step. Undoing Step (iii) implies replacing in each line identical neighbouring entries by the same entry. All edges emanating from one of the entries are to be adjusted so as to emanate from the new entry.

(vi) Note that none of the edges as drawn in Step (iv) could have intersected any other one, or could have met another one at some vertex, since they all have been drawn

in a vertical position in Step (iv). However, due to the above identification process most of the edges cannot be drawn entirely vertical anymore, and we may also find more than one edge emanating from the same vertex. However, due to the definition of synchronizability we cannot have two different edges that connect the same pair or vertices. In addition, edges can still not intersect each other. Hence the edges connecting two adjacent lines can be the same way regarded as naturally ordered as the vertices in each line are.

(vii) Observe that the above process gives that the entries in each line with an even line number and in the corresponding line below coincide. This makes us identifying the two lines with each other and hence undoing Step (ii), so that now the entries of the ith transformation tuple will be again be placed in the ith line. Note that we have not drawn any edges between the corresponding pair of lines to be identified. Those edges which had been emanating from the upper line of such a pair and had been directed upwards and those edges which had been emanating from the lower line and had been directed downwards are now redrawn so as to emanate from the newly created entry. As a consequence of this identification process, apart from the first line, each vertex can now have such edges emanating to the upper line and such emanating to the lower line. In addition, it follows that in any of these directions there has to exist at least one edge. This completes the definition of the transformation graph.

1.19 Analysis of the behavior of the infinite branches that can be found within a transformation graph:

The entries $\omega_{?,?}$ play the role of vertices of this graph, and we will refer to them accordingly henceforth. Observe that our graph is regarded as drawn into a plane with its vertices arranged in lines, its edges only connecting vertices of adjacent lines, and with the vertices and the edges of an according level naturally ordered from the right to the left. Since in the general case the number of vertices of a line should averagely grow in a similar way as the sequence $(\ell(\omega_i))_{i\in\mathbb{N}}$ for a word sequence ω, we expect in the general situation to see at each vertex more than one downleading edge, but only one coming from above. However, thanks to the situation of (2×2)-transformation matrices with identical entries in their bottom lines (cf. 1.11(2.)), we may also have situations with more than one upward leading edges. Hence the resulting graph need not be a tree, but it fits into the pattern described in C.5.10. Since the conditions C.1.6–C.1.7(iii) had been explicitly required in B.2.7, all infinite branches represent valid word sequences. These word sequences may be regarded as the intermediate states in the course of transforming ω into ω'. The leftmost of those paths represents ω, and the rightmost ω'. If two of these infinite branches b_1, b_2 satisfy that on each level b_1 uses some edge that is placed left of the edge that is used by b_2 or coincides with the b_2-edge, then we say that "$b_1 \leq b_2$". This defines a partial order on the set of infinite branches. However, since our graph is not a tree, we may have incomparable infinite branches and hence only a partial order. Thus we are left with a similar problem as when discussing in C.5.9/C.5.15 the graph of position sequences, namely how to obtain a total order out of this partial order. Other than in C.5.18(iii) we will not solve this problem by passing to equivalence classes, but by skipping to a maximal totally ordered subset. It is a straightforward application of

Zorn's Lemma that such a maximal linearly ordered subset exists (consider the set of all linearly ordered subsets and use the inclusion on this set of subsets for the application of Zorn's Lemma). Such a maximal linearly ordered subset can be geometrically interpreted as such a set of infinite branches where any two of them may coincide for some segment but must not cross each other. That latter property that they do not cross each other makes it evident that each edge of our graph has to be at least part of one infinite branch of our maximal linearly ordered subset.

(Assume the contrary, i.e. that there is an edge without branch passing over it. By running upward or downward till we meet an existing branch, and then following the existing one, we could define a new branch which could be used to extend our set that had been assumed to be maximal already.)

1.20 Definition of a transformations pattern of two equivalent word sequences:

Defining such a maximal set of non-intersecting infinite branches as described in 1.19 will be regarded as putting an order on a somehow complete subset of word sequences that arise as intermediary steps in the course of transforming ω into ω'. Hence we call such a subset a "*transformation pattern* between ω and ω'". The above arguments give that for any two equivalent ω and ω' such a transformation pattern can be chosen. However, note that finding such a pattern may imply an arbitrary choice. The fact that we did not get a tree as a transformation graph and hence needed an arbitrary choice to obtain a maximal linearly ordered set of infinite branches means that there may be elementary processes on distinct levels for which one cannot tell from comparing word sequences what is the natural order in which these process shall be performed. It is a part of the definition of the transformation pattern to define such an order in such a case.

1.21 On the scheme of the Proof of Thm.B.2.10(i):

Recall (Thm.B.2.10(i) and B.2.0(ii)) that it was the main purpose of having introduced weak equivalence in B.2.9(i) and in the addenda of this section, to show that weak equivalence precisely reflects the relative homotopy of paths. The proof of this statement will be fairly lengthy and indeed involve most of the parts of the entire Part D. This paragraph is now devoted to explain the basic principles of this proof:

The scheme for proving that weak equivalence reflects relative homotopy is somehow analogous to C.3.1 where we had to construct a path out of the combinatorial data given by a word sequence. In this case we have to construct a relative homotopy out of the combinatorial data that are given by the transformation pattern for word sequences. As in C.3.1 we will construct the desired homotopy H as a limit of a sequence of functions H_i, and this time the H_i will be homotopies that inside X_i homotope u and v into each other.

More precisely, it will take the following steps that are given by several propositions below to prove this lemma:

(i) We must prove that $u(0) = v(0)$ and that $u(1) = v(1)$, i.e. that the initial points and the endpoints of u and of v coincide (2.1).

(ii) The approximating functions H_i have to be constructed (8.11–8.17).

(iii) It has to be proven that these functions H_i uniformly converge (8.20)

(iv) It has to be proven that the corresponding limit function H_i only contains values in X (this will follow from the same arguments as in C.3.6, cf. 8.20)

(v) It has to be proven that limit function H takes on the boundary of its domain of definition the appropriate values such that it can be interpreted as a homotopy between u and v (this will follow, since in 8.11–8.17 we will not use the approximating paths of C.3.2 for the construction of the H_i, but all the H_i will be extensions of the functions u and v directly, so that we see on each point of the boundary domain a constant sequence, either).

As again indicated by the above references it will take a while to perform this programme. However, it should be pointed out at this place already that the transformation diagram as it had been introduced in B.2.5/1.3 can be interpreted as a preliminary state of the homotopies H_i, since the lines of these diagrams are a fairly good approximation of the level lines of the x- and y-coordinate of the function H_i as they will be constructed in 8.11–8.17.

The analogous statement that strong equivalence reflects the identity of paths (cf. B.2.10(ii)) is comparatively easy to prove, since several key statements can be reduced to discussing the non-deterministic aspects of the reading process B.1.4/C.2.2–C.2.7. It will merely take the next section to complete this proof:

§2. Verifying that the strong equivalence reflects the geometric identity

This section is devoted to prove Theorem B.2.10(ii) (also cf. B.2.0(i)). However, the first observations that need to be proved for that purpose do also hold for weak equivalence.

2.1 Lemma: *Let ω and η be two weakly equivalent word sequences and let u and v be paths that realize these word sequences. Then $u(0) = v(0)$ and $u(1) = v(1)$.*

Proof: Assume that $u(0) \neq v(0)$. Use that our corridor network $\mathcal{N}$ is arbitrarily fine, and hence that there exists a level index j such that $u(0)$ and $v(0)$ are not contained in adjacent or diagonally adjacent lattice squares of $\mathcal{N}_j$. Let $\zeta_j := \overleftarrow{\omega}_j$, $\xi_j := \overrightarrow{\omega}_j$, i.e. let ζ_j and ξ_j denote the lattice squares containing $u(0)$ and $v(0)$, respectively. Hence at least for the lattice index $j+1$ there exists a line A of corridors which is disjoint from the extended lattice squares ζ_j and ξ_j and which separates them. Note that the only ones of the list of five elementary operations B.2.2(i)–B.2.2(v) that can affect the first entry of the corresponding triple sequences are (far-distance) truncations and appendances, i.e. such an operation must be applied if in the course of the transformation process that brings ω into η and simultaneously has to transform ζ_j into ξ_j. Hence for the overall word sequence such an transformation process requires an unbounded number of truncations or appendances, since in the course of such a transformation process each of the iterated split-up lines of A-corridors has to be removed or appended.

We now have to ask in what nesting situation the corresponding lines could appear in a transformation diagram that is associated to the process of bringing ω into η:

There could be, of course, two lines centred around the top left corner or the bottom left corner, and they correspond to letters which are truncated to the beginning of our words at an early state of the process, or are at the same place appended at a late state of the process, respectively. Note that none of the lines can intersect, since we are here discussing only corridors belonging to the same basic direction as A and ignoring all processes that may have to be simultaneously performed to variables of the other basic direction. There could, of course, be further truncation lines which emanate and end at end at the left bar of our frame, but such lines reflect an appendance which is undone by a corresponding truncation at a later stage of the process. Since these processes cannot help to transform ζ_j into ξ_j, they are irrelevant. We hence conclude that the unbounded number of truncation have to be essentially grouped in at most two nests of truncation lines, and the corresponding nesting depth of at least one of these nests of lines has to be unbounded also. This implies that bringing ω into η can only be done by a process which has an unbounded truncation depth, and hence by B.2.9(i) ω could not have been even weakly equivalent to η which is a contradiction to our assumption that $u(0) = v(0)$.
$u(1) = v(1)$ follows analogously. □

2.2 Addendum to the Proof of 2.1:
At this place we should recall the phenomenon described in C.1.9: The same word sequence might describe essentially different paths if it contains letters from at most one of the basic directions. We should ask whether this does not contradict the statement apparently just proven, i.e. that any weak equivalence can never affect the start- and the endpoint. – Well, it does not if one also recalls that in C.1.9 we also agreed to use word sequences only as an abbreviation for triple sequences, for the sake of keeping our notation simple as long as this can be done without loss of essential information. However, this is one of the few cases where we have to consider the full triple sequences. When taking them into account, we see that essentially the same arguments as in the main part of this proof hold in this case, also:
By restricting our language to the case of a constant path, let us look at the path which takes on all its values at the point P and is described by a triple sequence whose first entries give some description of P by an interval nesting of lattice squares that accumulate at P, whose sequence of middle entries is empty, and each of whose third entries coincide with the corresponding first entry. If this triple sequence should be transformed to the corresponding description of the constant path that takes on the point $Q \neq P$, then, as explained in the main part of this proof, it takes truncations and appendances to impose the necessary changes to the first and third entries of our triple sequences. In that particular context the most promising strategy for such adjustment is probably the following: First move some first entry of our triple sequence to some neighbouring lattice square by appending the corresponding letter at the beginning of our word, and then to get rid of the corresponding letter again by truncating it to the end. The latter process will then affect the corresponding third entry in the same way as the first entry before. By the scheme we can "go" for each level from the P-square to the Q-square, and by applying the same Königslemma-strategy as in 2.5–2.8 we could

perfectly put together such transformation processes to a compatible transformation pattern as described in 1.17. However, by looking at the number of split-up letters and then using the same arguments as in the main part of the proof, we see that the number of truncations and appendances that we need on the ith level is unbounded when compared for $i \to \infty$. Since each appendance, if it is followed by a consecutive truncation as above, gives in our transformation diagram a line from the left bar to the right bar, we see that the unbounded number of truncations also implies an unbounded truncation depth in this case. Consequently, even if we might have got a compatible transformation pattern, it does not induce a weak equivalence, since it does not satisfy the condition B.2.9(i) of having a bounded truncation depth.
That way we saw that essentially the same arguments as in the main part of the proof of 2.1 had been applicable to that special case, also.

2.3 Corollary: *Let ω and η be two (weakly) equivalent word sequences. Then in the course of the transformation process that brings ω into η at most two horizontal variables and at most two vertical variables (one of each of them at the beginning and the other at the end of a word) can be affected by truncations and appendances. These variables correspond to the corridors that contain the start point or the endpoint of any path realizing ω or η* (the corresponding points have to coincide by 2.1). Observe that there is not any bound for the number of times at which in the course of such a process the letter may be truncated and reappended again, (and that the corresponding number of times could even be infinite if maximized over all lattice indices of our word sequence).

Proof: If more that the claimed number of letters are involved in truncation- or appendance-processes, then, at least at some intermediary state of transforming ω into η the start point of a realization of this word sequence would have to be moved to a lattice square which does not intersect any more with the corresponding lattice square that contains the original start point. Then the same arguments as given to prove the preceding Lemma 2.1 can be applied to rule our this possibility, cf. 1.19 for techniques for how to get hands on "intermediary states" and and how to see that they also represent equivalent word sequences. □

2.4 Lemma: *Let ω and ω' be two word sequences which are (weakly) equivalent to each other in such a way that it takes only truncations and appendances to transform ω into ω'.* (This implies, that ω is even strongly equivalent to ω'.) *Let η_i be obtained from ω_i and ω_i' by taking the word of those letters only which have not been removed at any intermediary step in the course of transforming ω_i into ω_i', and let $\eta := (\eta_i)_{i\in\mathbb{N}}$. Then η is strongly equivalent to ω and to ω'. In addition, there exists a continuous path $w : [0,1] \to X$ which has ω, ω' and η as valid readings.*

Proof: By 2.1–2.3 we know that the number of letters that might be truncated to be appended is small indeed; however this information turns out to be of little relevance for this proof to be given here:
In order to see that η is actually a word sequence we have to check the compatibility properties of the words of η. Note that C.1.6, the internal compatibility cannot be

affected by truncations or any by appendances that respect this internal compatibility, and hence it is definitely given for the words of η. Thus let us now consider the compatibility of two adjacent words of η denoted by η_i and η_{i+1}. Since these words are obtained from truncating the corresponding words of the word sequences ω and ω', there is not much chance for η for fail this property. It might appear that there is not any chance at all, since not only do ω and ω' satisfy the compatibility properties of word sequences, but also each intermediary step in the course of transforming ω into ω', i.e. each infinite branch in the corresponding transformation graph 1.18. However, η has not been defined by taking an infinite branch from this diagram, but by taking minimal entries in each line. Hence, in principle it has to be shown in the next paragraph of this proof that these minimal entries can be found along an infinite branch. I.e., we have to ask: "Could it happen that some letter A is not contained at the beginning of η_i, but we find letters a_1 and a_2 at the beginning of η_{i+1} so that they are split-up companions that cannot be related to a root-letter?" — Observe that this and the analogous symmetric situations (consider the ends of our words instead reverse the roles of A and of $a_1\&a_2$, i.e. so that A is present and a_1 and a_2 are missing) are the only possibilities how our manipulation of word sequences that was based on appendances and truncations only could give word sequences that fail to satisfy C.1.7. However, when viewed as follows, we definitely see that the above question is always to be answered by "No!":

> If η_{i+1} starts with "$a_1\,a_2\,\ldots$", then by definition of η these letters had to occur in all words written into the $(i+1)^{\text{st}}$ line of the corresponding transformation diagram Since all these words satisfy C.1.6–C.1.7(iii), the letter A must occur in all entries of the ith line as well, and hence it must have occurred in η_i.

Hence we proved that η is a word sequence, and thus we can now proceed in our proof by showing that η, ω and ω' are readings of the same path. Since each word of η_j is a subword of ω_j, and since we proved that η satisfies all conditions of a word sequence, it also follows that the bounding positions of each word together form a position sequence of ω. They can be used to split up ω in three part η', η and η''. Since the middle part of ω, i.e. η, had been under consideration already, we now want to focus on η' and η''. By C.5.12/C.4.1(ii) any path realizing ω is split up into three subpaths that correspond to the split-up of ω into η', η and η''. By C.4.1(ii) the second of these paths has to coincide with any path directly constructed for realizing η, and hence it just remains to be proved that the paths that realize η' and η'' are constant. However, since this is a straightforward consequence of 2.3 (that implies that η' and η'' are bounded word sequences) and of C.4.8(i), we get that any path realizing η is also realizing ω and vice versa. A similar result is obtained when comparing the paths that realize ω' and η. □

2.5 Lemma: *Let u be a path and let ω and η be two word sequences that can be read from it. Then ω is strongly equivalent to η.*

Proof: First observe that skipping from one valid reading to another gives for each level index a valid transformation: A mere readjusting of the boundaries of reading domains does not affect the reading of the ith level, yet: the decision to disregard or, conversely, to take into account two inverse acts of crossing that just end at a tangency point gives

an elementary cancellation or an insertion, and a similar revision of the way how to report a partial act of crossing at the beginning or at the end of our path corresponds to a truncation or an appendance, and finally, the decision to put letters of intersecting reading domains in different orders corresponds to a transposition. Hence we get at least some tuple of transformations on each of the levels of our word sequence. When putting down these tuples we avoid unnecessary repetition of transformation patterns as one gets them if one consecutively performs a certain operation and its inverse: Wherever there has to be some decision a different treatment of tangencies, we just perform once the corresponding process, and if these processes do not suffice to turn the words of different readings into each other, we perform transpositions in such a way that each variable is moved monotonously to its new place.
These tuples have now to be synchronized in order to meet the demand B.2.8, and the scheme of synchronization shall be given by combining an upward extension as constructed in 1.9 with the Königslemma type of conclusion. Of course, in this context the upward extension must be constructed in such a way that each of the correspondingly constructed transformation tuples does just correspond to a readjusting of the corresponding readings. In this context recall that our method of constructing the upward extension in 1.9 was based on reading the words of less finer indices from a path, and hence by using the same techniques as in the proof of 1.9 we can find an upward extension with the desired properties. Observe that the proof of 1.9 was based on constructing auxiliary paths according to the scheme given by C.3.2, and sometimes the properties of these paths had been explicitly used. However, when now working with explicitly given paths instead, all these references can be accordingly replaced. Instead of referring to the property that our path avoids to enter $\mathcal{N}_{i+1}$-crossing regions and its reading hence provides a correspondingly clear order of letters if ever our path should have entered an $\mathcal{N}_i$-crossing region, we can now refer to C.2.4(3) and to the fact that we only attempt to interchange variables if we know our path to go through a crossing region of corridors; instead of referring to the fact that the C.3.2-paths do not perform tangencies, we can now refer that we only read our word sequences in such a way that we decide consistently how to treat tangencies. That way a sequence of $(i \times l_i'')$-matrices as in 1.17 is build up. These matrices do not yet satisfy all demands of 1.17, since there is not any warranty that the upward extensions from different index levels cause identical entries on the levels with small indices. However, by now constructing an appropriate "graph of reading adjustments", and by applying the scheme of conclusions of C.5.10 to it we will prove that amongst the readings that are obtained by upward extension there are enough alternatives to construct matrices that satisfy all conditions of 1.17. That way we will be able to complete the proof of 2.5 in 2.7/2.8.

2.6 Definition of the *"graph of reading adjustments":*
Assume that we are in the same situation as in 2.5: From some path u two different word sequences ω can be read and similar as in the proof of 2.5 the necessary reading adjustments are recorded in 1.17-style matrices. For the purpose of constructing our graph of reading adjustments we regard each line of our present $(i \times l_i')$-matrices (with doubled entries removed again) as a vertex of a C.5.10-graph. Lines coming from the

i^{th} row of our matrix are put into the ith line of the graph to be constructed. Vertices of adjacent lines that in one of our matrices had appeared as adjacent lines are to be connected by an edge. In addition, we see that each line of our graph contains only finitely many vertices, since we decided to undo the doubling of entries before we turned the lines of our matrices into vertices and since there are only finitely many different words that can be read on the fixed level from a path. Apart from that, in the above conclusions we can refer to the facts that when constructing these tuples we avoided to construct senselessly long transformation tuples and that even the upward extension cannot produce them.

2.7 Completion of the proof of 2.5:
Note that the corresponding graph of reading adjustments as defined in 2.6 fits into the framework as described in C.5.10 and hence it has an infinite branch. Such an infinite branch then gives a synchronizable tuple of transformations as required in B.2.8. Hence it just takes to prove that the nesting depth is bounded in order to get see that we actually managed to satisfy all combinatorial criteria of strong equivalence:
It is clear that never nested cancellation could have happened: If a path performs a tangency, then the two letters might be cancelled, but (apart from what might happen in the other basic direction) there cannot be letters around which in addition are cancelled. Similarly just those two letters which belong to a crossing region of corridors may be interchanged. This could imply the possibility that one reading domain contains a lot of reading domains of the other basic direction, then our letter might be interchanged with all of them. However, the lines in the transformation diagrams of these other letters can be intersected at most once, and this cannot cause a huge nesting depth (for a general picture see the proof of 6.2 in 6.3–6.6 and cf. B.2.5/1.3). However it appears as if a pair of two neighbouring horizontal letters could never be interchanged with a pair of two neighbouring vertical letters. Even if the latter statement is not correct since there are some exceptional situations (as indicated by having used the conjunctive, cf. 2.8), for the sake of having a simpler language we will proceed arguing as if it was and will postpone the discussion of the exceptional cases to the postscript 2.8 of this proof. Finally, truncations are also restricted to those at most four letters which belong to corridors that either contain the start point or the endpoint of our path. These effects together, with the fact that we did not perform senseless transformations give that on each level any transformation diagram looks at follows:
If we look at the lines coming from one of the basic directions only, we may see near the top bar and near the bottom bar cups or caps, but never in a nesting position. In two of the four corners there may be at most one quarter-circle around this corner, or we see one horizontal line running from the left bar to the right bar (cf. 3.7(iii) and 2.2). Apart from that there might be lines running between the top bar and the bottom bar which do not intersect each other. This is the picture for one of the basic directions, the picture for the other basic direction looks similar. Both pictures have now to be superposed in one frame. When superposing them, lines coming from the different basic directions (the best way is to think of them as couloured in different coulours) may intersect each other, but never may a line a of one colour intersect two lines c and d of

the other colour, if some other line b that belongs to the first colour also intersects b and c. From that picture it is a straightforward combinatorial exercise to compute the highest nesting depth that could arise if all effects come together. If seems not worth doing this exercise here, since the above description should have made clear that it is a global bound, which limits the number of lines that one has to cross if one travels from any start-point at the top-bar to a suitable target on the bottom bar. And that finally gives that by now we have fulfilled all combinatorial demands of strong equivalence.

2.8 Addendum to the Proof 2.5/2.7:
The exceptional situations as indicated by the conjunctive-sentences in the above proof are more thoroughly described in 2.10, even though there they arise under different assumptions. Here they are due to the effect, since a tangency-point may belong to two reading domains (coming from the two inverse acts of crossing), and, if this tangency-point should be lie on a corner point, we might face a point which actually belongs to four reading domains. Apart from that, due to the effect as described in C.2.7, i.e. that reading domains may be readjusted without affecting the associated word sequence, such segments where our path runs along a boundary line of a corridor, must be regarded as segments which potentially belong to two reading domains. However, even in this case the two reading domains have to belong to inverse acts of crossing the same corridor, since as long as our path has to remain on a boundary line it has not the opportunity to cross any other line of corridors of the same basic direction. In addition, since due to the effects as described in C.2.4 any non-uniqueness of how to adjust a reading domain ceases at those places where our path passes along the outside boundary of a hole that crops up in our corridor of consideration, it it also clear that the two inverse acts of crossing that our path performs at the end of its boundary-line segments have actually to belong to the same corridor, we cannot have an exceptional situation when they just belong to the same line of corridors.
Due to these effects we have to admit that actually we might see more (and in particular deeper nested) transpositions as admitted in the main part of the proof of this lemma. On the other hand, wherever one of the exceptional effects should have occurred, then cancelling pairs of letters are amongst the letters to be transposed. Hence in principle within the discussion of this proof the exceptional cases can be easily avoided: all it takes is to decide to perform the corresponding moves of letters not by actually transposing these letters, but by cancelling them at one place and reinserting them at another place. This is all what it takes to complete the proof in this case.
However, since not just the statement of this lemma, but also its proof-techniques will have to be referred, we are now going to analyze that even if we do not avoid the corresponding transpositions of cancelling pairs in the exceptional situations but perform them as transpositions instead, we still get a transformation pattern with the properties of strong equivalence. As in the main part of the proof of this lemma, only one aspect may be doubted, namely that the corresponding sequence of transformation diagrams has bounded nesting depths:
Use "diamond" to denote the pattern of lines in a transformation diagram that one sees if two lines that are parallel to each other (at least at the crucial segment) intersect another

pair of apparently parallel lines. In the main part of the above proof we argued that "no diamond" implies "bounded nesting depth". Having had a look at the exceptional cases we have to admit that it might be possible to readjust a reading in such a way that even diamonds might occur. However, the discussion of the exceptional cases gave that the corresponding transformation diagrams might contain "lines of diamonds" (i.e. two parallel lines which are intersected by more than two other lines, so that a number of diamonds is lined up between the first to parallel lines), but we never can see a (2×2)-pattern of diamonds. And with a "no (2×2)-diamonds" assumption the existence of a global bound for each transformation diagram can be concluded in a similar way as in the main part of the proof in 2.5/2.7, even if in this case we get, of course, a larger bound. This observation finally completes the proof of 2.5. □

2.9 Lemma: *Any strong equivalence transformation pattern can never contain*
(i) nested cancellations on any level,
(ii) more truncations as permitted by 2.3,
(iii) the interchange of six letters "$A \dots B \dots C \dots D \dots E \dots F$" into "$D \dots E \dots F \dots \dots A \dots B \dots C$" by nine transpositions ("triple interchanges").
(iv) Similarly as in (iii) "pair interchanges", i.e. the interchange of four letter "$A \dots B \dots \dots C \dots D$" into "$C \dots D \dots A \dots B$" by four transpositions, cannot occur except if one of the pairs (A, B) or (C, D) is a cancelling pair (i.e. two inverse letters that are in a cancelling position).

Proof: Note that similar properties did already occur in the proof of 2.5, but unlike as in that case (where they had been derived from a geometric origin: how could different readings of the same path look like?), they have now to be derived from the combinatorial demands of strong equivalence. I.e. essentially we have to prove in the following, that if any of these transformations should occur in a compatible transformation pattern, then the nesting depth of this pattern must be unbounded so that it could only be weak, but never strong equivalence.
Ad (i): For the situation of nested cancellations this can be seen as follows: As stressed in 1.1(i) already, a cancellation of letters of the ith word can only be extended to higher index levels if at least the innermost split-up letters of the cancelling letters are also cancelled. And if we should manage to get into the situation that the corresponding processes can be performed to the for the split-up letters of our outer pair, then on all higher levels all split-up letters of the inner pair have to be gone before. Since they are neither the first nor the last letters of their basic direction, the number of these letters has to grow at least as fast as the powers of two, if going to higher index levels. Consequently the corresponding transformation diagrams will contain huger and huger nests, and by having observed this we see that the nesting depth cannot be bounded.
Ad (iv): By C.1.6 our assumption gives that A and B must be from parallel neighbouring lines of corridors, and hence on the next index level we see there will be a non-split-up letter a' between the split-up letters of A and of B, and similar we will see some c' between the C- and D-offspring. What can be said about the order of split-up letters of any of our letters for the higher index levels:

- ▷ In the word sequence where we see "$A \dots B \dots C \dots D$" the iterated first split-up letter of B has always to come before the iterated right split-up letter of C.
- ▷ Conversely, in the word sequence where we see "$C \dots D \dots A \dots B$" the iterated first split-up letter of D has always to come before the iterated right split-up letter of A.
- ▷ Since letters of the same basic direction cannot have split-up letters with reversed orders, the first property implies that in the corresponding words all iterated split-up letters of a' come before all iterated split-up letters of c'.
- ▷ Analogously it follows that for words as described in the second item all iterated split-up letters of c' can be found before all iterated split-up letters of a'.

Hence all the iterated split-up letters that are described in the preceding two items would have to be interchanged in the course of a compatible transformation pattern that turns "$A \dots B \dots C \dots D$" into "$C \dots D \dots A \dots B$". This would give that systems of essentially parallel (at least: non-intersecting) lines would have to cross each other, and in 1.6 it had been described that this is sufficient to get an unbounded nesting depth.

Ad (iii): This essentially can be proved by the same line of arguments as (iv). In this context observe that the additional assumption on the exponents in (iv) was only needed to conclude that there have to exist additional letters on the appropriate index levels between the split-up letters of pairs that have to be interchanged. Since in cases where triples have to be interchanged, the outer entries of the triples can be regarded as pairs and the middle entry as some additional entry in between, the same line of arguments that gave (iv) also implies (iii).

Ad (ii): No addendum to the arguments as given in 2.3 is required. □

2.10 Examples: Observe that the additional condition in 2.9(iv) on the exponents was not only essentially needed in the proof, in addition there exist counter examples if this condition is not fulfilled:

(1.) Assume that a path goes diagonally over a crossing region of corridors till it touches the opposite corner point. There it turns around and goes back. I.e. at the turning point it performs two acts of tangency simultaneously. If correspondingly read, this behaviour of our path could by as well described as "$AA^{-1}BB^{-1}$" or as "$BB^{-1}AA^{-1}$", if A and B denote the two corridors that meet at the crossing region. In addition, the skipping between these two readings may not just be performed by cancellations and reinsertion at some other place, but also by mere transpositions. Since there are not any letters that show up between A and A^{-1} or between B or B^{-1} and hence a mere adjusting of the way to read our word (i.e. to put the four reading domain that reach the tangent point into a different order) could give this transposition, even for higher levels these permutations can be performed in a bounded way according to 2.8.

(2.) Think of a path that behaves as follows: It crosses one horizontal corridor A till it reaches the opposite boundary. There it continues on the opposite boundary, while, following that boundary line, it consecutively crosses the vertical corridors B, C and D. The it crosses A again in reversed direction and disappears from our picture. The natural reading from that segment of our path is apparently "$ABCDA^{-1}$". However, if we are not in the case of fine holes that are tangent to the boundary (cf. C.2.5), the

segment on the boundary could be the same declared as part of the A-act of crossing, or as a part of the A^{-1}-act of crossing. Hence "$AA^{-1}BCD$" or "$BCDAA^{-1}$" also appear as valid readings. In addition by what has been proven in 2.5 (cf. in particular the addendum 2.8), the skipping between these readings can be done by matters of strong equivalence only, namely the corresponding readjustment of reading domains and the use of C.1.3(ii), i.e. the interchanging of letters that belong to overlapping reading domains.

2.11 Lemma: *Let u be a path and ω be one of its associated word sequences. Assume that ω is strongly equivalent to some word sequence η. Then η is a possible reading of u, also.*

The Proof of this Lemma is lengthy and hence split up into 2.12–2.16. It rests on applying graph-theoretical conclusions to a "graph of all possible readings" which will be defined below. Once having set up this graph the lemma actually will follow by proving the following three assertions in 2.14–2.16:

(i) If each word of the given word sequence η occurs in one of the vertex labels of the corresponding line, η can be read from u.

(ii) If this does not happen for at least one line, the η cannot have been strongly equivalent to ω or to any other word sequence that could be read from u.

(iii) For each marked vertex in the graph of reading possibilities there exists one finite monotonous edge-path that connects this vertex to the root-vertex and which only runs over marked vertices.

Actually, the lemma will be implied by (i) and (ii) alone; however it will turn out that (i) cannot be proven without (iii).

2.12 Definition of the *graph of all possible readings* for a given path $u : [0,1] \to X$.

(i): Given a path u only. Then recall that a "reading" (each of the classes of essentially different readings will give rise to a vertex of that graph) is an adjustment of reading domains that is compatible with the rules explained in B.1.4. In that context readings are regarded as "essentially the same" if the same number of reading domains that correspond to the same letters have been adjusted, so that the possible "minimal reading domain" as it is used in the course of our algorithm C.2.2 is just a subset of the corresponding reading domains. This needs to be explained a bit more precisely: In C.2.2 reading domains had been in such a way adjusted, that at the start point of the reading domain our path was on one side of the corridor and it was not allowed to touch the corresponding boundary line till reaching the opposite side, and the first point where the path reached that side was already to be considered as the endpoint of the reading domain. This is the "minimal adjustment". However, now we want to use the "maximal adjustment" as annotations for our vertices, and that is defined as follows:
If a path has reached the endpoint of the minimal adjustment, and it remains a while on the corridor boundary, this segment on the corridor boundary is still regarded as a part of the maximal adjustment, at least as long as our path does not pass the outside boundaries of finer holes that are tangent to the boundary. If the paths at the end of its minimal adjustment just performs a tangency to the corridor boundary and goes a bit

back into the corridor (without performing a complete act of crossing again, i.e. it finally has to leave the corridor on that side that contained the end of the minimal reading domain or at least to reach another tangency point on that side), then the corresponding segment is to be considered a part of the maximal reading domain. If our path after the end of the minimal reading domain and before its final exit from the corridor should have touched the opposite boundary, then even a maximal reading domain could have ended at the last tangent point on the exit side that comes before some tangent point on the opposite side; the same holds for the last tangent point that a path takes on the boundary if it ends inside the corridor. The various possibilities for such adjustments of reading domains will be considered as essentially different "maximal readings". We put these maximal readings as annotations of our vertices each of which represents a class of readings that do not differ essentially. The associated words that correspond to that reading (note, it can only be a class of words which differs just by permutations of variables the corresponds to overlapping domains) is to be attached as additional information to this annotation. Note that these "maximal readings" may violate one of the demands of B.1.4 (namely B.1.4(iii)) and hence they are not necessarily "valid readings":

> In case a path performs an act of crossing a corridor, then remains a while on the opposite boundary and then crosses the corridor again in the opposite direction and in case we decide to acknowledge both acts of crossing by a letter, then according to the conditions of a maximal reading the segment on the opposite boundary (the same might happen if the corridor has just gone a tiny bit back into the corridor between its both acts of crossings) will be a part of both reading domains, i.e. reading domains of non-commuting letters will overlap. We will have to keep an eye on that effect.

However, these "maximal readings" (valid or not) define the vertices of our graph. They are drawn into the plane such that they admit a line structure in the sense of C.5.10, the ith line being defined as those readings that arise with respect to the corridor lattice $\mathcal{N}_i$.

Now we still need to insert edges into our graph: For that purpose we try to find the corresponding reading with respect to the preceding less fine lattice in the sense of B.1.4(vi). According to the principles of C.1.7(i) it follows that if a pair of split-up corridors is crossed, then the root-corridor must have been crossed, also, each vertex can be connected to at least one vertex of the line above.

(ii): Given a pair (u, η) where u is a path as in (i) and η is one of its possible readings. Then we put additional marks on all those vertices where one of the associated words is the corresponding word of η.

2.13 Remarks on Definition 2.12:

(i) Note that drawing an edge in a graph of reading possibilities does not imply that all reading conditions of B.1.4 are satisfied between the maximal readings that are correspondingly connected: The conditions B.1.4(vi)–(vii) of matching endpoints of reading domains need not be fulfilled. However, since all our readings used to be maximal, the start- and end-points of the reading domains of the ith line have just to be moved into the corresponding reading domains to match with the start- and

the endpoint of the corresponding split-up letters. That way an infinite branch of our graph will give rise to a valid reading; we are just not allowed to use the start- and endpoints of the annotations as start- and endpoints of our reading domains, we have to set up the sequence of the start-points of all the corresponding split-up domains, and then to use the limit of that sequence (it is a monotonous bounded sequence which hence has a limit) for the valid reading that corresponds to the infinite branch.

(ii) A graph as defined in 2.12 will in general-position situations be fairly slim (with at most 16 vertices on each line), for a general position situation and a path that has its start- and endpoint outside of all corridors it will even be just a segmented ray (i.e. just one vertex per line). The "16" as mentioned in the preceding sentence refer to the following phenomena: If the path ends inside a corridor, this gives by C.1.3(iii)/B.1.4(3.) essentially two possible readings. If a path should end inside a crossing region of corridors there are four possibilities how to read this situation, and since that could happen at the beginning and at the end of a path there are $4^2 = 16$ overall possibilities. However we can get even more vertices in some line, if we have to deal with a path that performs a lot of tangencies. One of the worst cases appears to be if a segment of our path looks like a a part of sinus-curve with the lines given $y = 1$ and $y = -1$ being the boundaries of our corridor. In that case we experience really a lot of freedom which of the corresponding acts of crossing this corridor should be acknowledged by a letter and which should not. However, the freedom is finite, since by continuity our path can only run a finite number of times between different sides of a corridor. This guarantees that each line in our graph is finite and hence the Königslemma applicable in the following. Note that we will set up our graph in such a way that it has a "zeroth" line that contains one vertex ("root vertex") that corresponds to the empty reading from the empty corridor network, (cf. the zero-step of A.3.6, i.e. A.3.9(2)).

Having set up the graph of all possible readings we are now ready to start the proof of our Lemma 2.11:

2.14 Proof of 2.11(i): In principle, again we want to give a Königslemma-type of argument: an infinite branch in the corresponding graph of reading possibilities shall define the desired compatible reading. The Königslemma (cf. C.5.10) gives that our graph must have such an infinite branch, however in our case this information alone does not suffice. For, when using markings as defined in 2.12(ii) with respect to our word sequence η, we see that what can be concluded from the assumption of 2.11(i) is just that our graph has at least one marked vertex on each level, but that they are placed in such a way that an infinite branch exists that only runs over marked vertices cannot be concluded from that set-up alone. This needs additional combinatorial information, e.g. a statement such as 2.11(iii). However, observe that 2.11(iii) together with the properties of our reading graph as explained above and with the classical infinite-branch construction mechanism of the Königslemma imply the assertion of our lemma in Case 2.11(i). □

2.15 Proof of 2.11(iii): In principle 2.11(iii) follows from carefully comparing the compatibility demands C.1.5 of a word-sequence (which are satisfied by the given η) with that compatibility demand of the "maximal reading" as explained in the first half of the proof of this lemma. According to the idea, that the C.1.5-conditions are nothing but an encoding of geometrical demands into combinatorial ones (cf. the final remark of C.1.7 or the proof of C.2.1), this is not that surprising. However, some fine points, in principle centred around the freedom that the reading process provides, still deserve some attention:

- ▷ The freedom that often comes along with tangencies cannot give an obstruction in our case, since our tree had been constructed in such a way that it in principle acknowledged any possibility for how to read at such a tangency. Hence this degree of freedom cannot be the reason that we might not find marks on a graph-theoretical connection of a given marked vertex and the root vertex.
- ▷ The order condition can also not give such an obstruction, and this is mainly due to the fact that the combinatorial conditions are more restrictive than the geometrical ones: If the order of the four split-up letters a_1, a_2, b_1 and b_2 of two letters A and B is not just "$a_1a_2b_1b_2$" but one of the other four permitted orders of C.1.8(ii), then our path must have gone through the crossing region of A and of B, and hence there must exist a valid reading for both of the corresponding orders of A and of B in our graph. However, if the order of our letters should actually be "$a_1a_2b_1b_2$", then the order of A and B in the word sequence is not free, but our path could still have gone through the crossing region so that still both possibilities for a reading might be found in our graph. However, our graph contains all possibilities and connects them appropriately.
- ▷ Now consider the degree of freedom we see when we find a letter with a missing split-up partner. By compatibility demands of our word sequence this can only happen for the first or the last letter of the corresponding basic direction, let us restrict our language to the case where it was such a last letter. Hence any path that realizes the corresponding word of a word sequence must end either between the corresponding split-up corridors (the one showing up and the one missing), or inside one of these corridors. And hence we know that our path on the preceding level also ends inside the area covered by the root-corridor. Hence, with respect to the root-corridor our path performs (what can be considered as) an uncompleted act of crossing. Thus we enjoy in the reading process the full freedom of C.1.3(iii), and hence, whatever was the given word of the word sequence η with respect to this aspect, it is a possible reading. Hence this aspect can also not give an obstruction for finding a way over marked vertices.
- ▷ The same argument in principle also holds in the zero-split case, i.e. in the case where the preceding word of η contains a letter which does not have any of its split-up letters. In that case, similarly as in C.3.14–C.3.15 $\overleftarrow{\eta}$ and $\overrightarrow{\eta}$ provide the information that our path actually lies inside the corresponding corridor, and this suffices to conclude that the prescribed version, i.e. reading the corresponding word so as if this corridor is crossed in an uncompleted way, is a possible reading in any such case. □

2.16 Proof of 2.11(ii): Assume that η_i is not amongst the possibilities to read the ith word from our path u, and recall that our graph had been set up so as to contain all of them. In that case it suffices to consider the picture of a plane where finitely many holes are cut out and connected by corridors. Since our graph has taken all possibilities into account for (dis)regarding a start- or end-segment that lies into a corridor and for (dis)regarding acts of crossing corridors that just end up with a tangency (or to put an arbitrary order on the corresponding letters if a path goes through a crossing region of corridors), we conclude that either u must perform some acts of crossing more or less than are reported by η_i or that it must pass aside a crossing region of corridors which according to η it should have passed along the other side or should have entered. Since in any case the missing acts of crossing corridors cannot be just a matter of reinterpreting tangencies, u would have to be moved into a really different position to permit the desired reading η_i. But such a move of u would always change a word sequence by what might still be a weak equivalence relation, but never could be strong one! – E.g. if u just approaches a corridor boundary and then turns around without having touched it, but η_i reports a path that intersects the boundary line or at least touches it, then any path which permits such a reading as η_i would have on some sufficiently fine level also crossed corridors (and their full set of split-up corridors) that u has not seen (at least, in the corresponding segment). Bringing all the letters into the word sequence η is an unbounded process, and hence it is not a matter of strong equivalence (cf. 2.9(i)). The analogous argument holds for all the geometrical situation when a path within the fixed corridor lattice $\mathcal{N}_i$ should perform any different corridor crossings as reported by some given word sequence, and if all degrees of freedom of the reading have already been taken into account. In that case the given word sequence cannot be strongly equivalent to any word sequence that could be read from the given path. And hence this Case 2.11(ii) contradicts to the assumptions of Lemma 2.11. Since in the other case, 2.11(i), the assertion of our lemma could be shown, the proof of 2.11 is now complete. □

2.17 Corollary: *The reading process of B.1.4 and the construction process of realizing paths in C.3.1 both induce bijections between the set of abstract paths and the set of strong equivalence relations.*
Proof: This follows directly from C.4.1(ii), C.2.6 and 2.11. □

With the above Corollary we have completed the proof of Theorem B.2.10(ii), that had already stated in a preliminary version as "Theorem D" in the Introduction. This makes us now proceed with the proof-programme for Theorem B.2.10(i) as outlined in 1.21. At first this requires some preparatory steps to which our next section is devoted:

§3. Essential definitions for and on the structure of the next subsections

3.1 Remark: Observe that there are several possibilities for how one might understand terms as "truncatable" or "cancellable", the most natural two of which are the

following:

(i) whether it is possible to perform a truncation/cancellation to some given letters of a word sequence and to extend this process in such a way through the entire word sequence that it gives a two-column matrix of compatible transformations (i.e. even for the higher levels it is not necessary to perform any other operations than at most the one which in the sense of B.2.7(iii) might be needed in order to extend the given truncations/cancellations compatibly to the corresponding level) — or

(ii) whether in the course of a compatible reduction (in B.2.3 we have subdivided transformations in reducing and non-reducing ones) of our word sequences it might happen that the corresponding letter(s) are at some state truncated/cancelled.

In the course of this paper we will have to discuss criteria for both types of truncatability or cancellability. However, the following paragraphs 3.2–3.5 are only devoted for discussing criteria of the first of these types, the other type will be discussed in the course of the proof of Lemma 11.6. In addition, we hereby agree that our use of the terms "truncatable" or "cancellable" is to be understood in the sense of (i), and this will not change in or during the proof of 11.6 in 11.6–11.9.

3.2 Proposition: *Let L be a letter which is contained in one of the words of a word sequence ω. Then L is truncatable to the beginning of ω if and only if one of the following two conditions holds:*

(i) One of the iterated left split-up letters of L does not show up in the corresponding word — or:

(ii) All of the iterated left split-up letters of L exist and each of them is the first letter of its basic direction in the corresponding word. In addition, each of these iterated split-up letters does commute with all its predecessors on the corresponding level.

Proof: Observe that in cases (i) and (ii) L and each of its existing iterated left split-up letters has to occur as the first of its basic direction in the corresponding words, since, if one of them is preceded by some other letter L' (and hence (ii) is not satisfied), then at least the iterated right split-up letter of L' would have to occur on each higher level. Then L would have to have the full set of its iterated left split-up letters and could not satisfy (i), either. Having made this observation we see in Cases (i) and (ii) a method for truncating L: we remove L and all of its higher (if existing) left split-up letters, and this gives all changes on the higher indexed words. In order to obtain how the truncation of L can be compatibly extended to the lower indexed word, we ask whether L is a left split-up letter, a right split-up letter or not a split-up letter at all. The words above the one containing L are only affected, if L is a right split-up letter, in this case the split-up root is also truncated. This process is iterated, i.e. we remove the split-up root of each truncated letter, if the truncated letter had been the right split-up letter of its root. The process stops in case the topmost level has been reached, or if some truncated letter has been reached, which has not been a right split-up letter. It is a straightforward exercise to verify that the given procedure generates a valid word sequence again, and hence to verify the "if"-claim of parts (i) and (ii). It is also not hard to verify the "only-if" claim, since the only alternative left by (i) and (ii) is given by the situation that L has the full

set of its iterated left split-up letters, but at least one of them is not the first of its basic direction or is preceded by non-commuting letters. And this assumption gives a perfect obstruction for extending the truncation of L to the entire word sequence. □

3.3 Remark: The analogous proof as for 3.2 gives that *L is truncatable to the end of ω if and only if one of the following two conditions holds:*
(i) One of the iterated right split-up letters of L does not show up in the corresponding word — or:
(ii) All of the iterated right split-up letters of L exist and each of them is the last letter of its basic direction in the corresponding word. In addition, each of these iterated split-up letters does commute with all its successors on the corresponding level.

3.4 Proposition: *Let L be a letter which is contained in one of the words of a word sequence ω. Then L is cancellable against some letter L^{-1} if and only if either the first two or the last two of the following three conditions hold:*
(i) L comes before L^{-1}, and for the level that contains L and L^{-1} and for all subsequent levels we have that the subword that is bounded by the iterated right split-up letter of L and by the iterated left split-up letter of L^{-1} only contains letters the are commuting with these split-up letters.
(ii) On each level the corresponding iterated leftmost and rightmost split-up letters as considered in (i) and (iii) are inverse to each other.
(iii) L comes after L^{-1}, and on each level the subword that is bounded by the iterated left split-up letter of L and by the iterated right split-up letter of L^{-1} satisfies a similar condition as in (i).

The Proof follows a similar strategy as the proof in 3.2 did for truncations: One has to observe by elementary combinatorics that if the given conditions are satisfied, then there exists at least one combinatorial possibility the extend the given operation to the finer levels, and, if not, then a sufficient obstruction crops up on some level. Most of this is immediate, hence we restrict ourselves to mentioning one or the other point below: In the cancellation case all of the relevant split-up letters actually have to appear, since the presence of the split-up partner makes it impossible that any of these letters might appear as the first or last of its basic direction. In this case any compatible extension of our cancellation requires that the relevant split-up letters do disappear by cancellations also, since otherwise the corresponding level would still report a completed act of crossing the L-corridor, and then L could according to C.1.5 not be taken out, either. □

3.5 Proposition: *Let L be a letter which is contained in one of the words of a word sequence ω. Then L is transposable with respect to some other letter L' (without loss of generality, assume that L comes before L'), if all of the following three conditions are fulfilled:*
(i) L commutes with L', and these two letter are in an adjacent position.
(ii) For any finer level of L and L' that contains both, the iterated right split-up letter of L and the iterated left split-up letter of L', either these two letters are in an adjacent

position or the corresponding L'-split-up letter comes actually before the L-split-up letter.

(iii) On any level that sees the iterated right split-up letter of L before the iterated left split-up letter of L', these two letters have to commute with each other.

Proof: This follows from the same scheme of proof as 3.2/3.4. However, note that in our case now the order of split-up letters is much less restricted than in the cancellation case 3.4. We might see the corresponding reversion of the order of our split-up letters on some fine level already, and if this happens, then one need not impose any changes to the yet finer levels if one wants to extend the transposition throughout the word sequence. A similar situation is given if some of the corresponding split-up letters does not show up on some fine level, since according to C.1.7 this case is treated as if the corresponding split-up letter was some ghost letter beyond the end of our word, and hence this case is to be analogously treated as the reversion-case.
From the geometrical arguments of 1.9 in 1.12–1.13 it should be clear that obstructions for compatible extensions of transformation processes can never come from less fine levels. However, for other reasons the corresponding case will have to be discussed within this paper from the combinatorial point of view, also, see 3.13, so that we will see again that such an obstruction does not exist. □

3.6 Definition of weak and strong equivalence relations:
In B.2.9–B.2.12 we mentioned three equivalence relations for word sequences that are to be discussed in this paper. The purpose this paragraph is to raise this number to nine. The reason that makes us now look at our equivalence relations in a different way as before is mainly centred around the question, to what extend truncations and appendances should be regarded as a legitimate process:
Recall that the main idea of introducing these equivalence relations is to prove the existence of almost unique representatives in the weak equivalence class that can then be used to define the combing paths of our homotopy combing (cf. A.5.5(ii)). For that purpose it seems desirable to permit truncations as legitimate processes since otherwise, if the desired endpoint of a combing path falls into the region of a corridor, (cf. C.2.2(i)) there would be automatically different possibilities of describing the paths by adjusting the corresponding third entry of the triple sequences correspondingly, and without truncations and appendances these possibilities could not be equivalent. On the other hand, we will also see the following situation: Let A, B, C three points and, interpreting paths that are associated to reduced word sequences as "algebraically shortest paths" (cf. 0.0(Thm.F)), assume that we found the algebraically shortest path between A and B and now we look for some concatenable path from B to C. For that purpose it seems desirable to find the "algebraically shortest word-sequence" in the relative homotopy class that respects the corresponding description of a point B as adjusted by the corresponding triple sequence that describes our path from A to B. In that context we consequently would like to forbid those truncations, if they would affect the corresponding description of the point B. For that purpose it seems to be a helpful terminology if we now introduce

- ▷ The weak equivalence relations as defined in B.2.9(i) is hereby split up into four different equivalence relation, which will be denoted as the *"free weak"*, the *"fixed weak"* and the two *"semi-free"* weak classes. The definition of the free weak class coincides precisely with what had been said in Section §B.2, the fixed weak equivalence relation results from this relation by forbidding all truncations and appendances, and the two semi-free classes result by forbidding only those truncation/appendance processes that are either performed to the beginning of a word or to the end of it.
- ▷ Analogously the strong equivalence relation as defined in B.2.9(ii) is split up into four different equivalence relations.
- ▷ Since for the firm equivalence relation as defined in B.2.9(iii) truncations had never been a permitted process, a similar split-up of that relation does not make sense. Hence this relation remains as it was defined in B.2.9(iii) and gives the ninth equivalence relation that is to be considered in this paper.

Note that in many contexts it will be possible to treat simultaneously the equivalence relations that now have been arisen by splitting up our old definition, without having to refer to them explicitly: A lot of discussions will be made by distinguishing cases such as "either a certain letter is truncatable or it is not", and in such a context it often does not matter whether the corresponding letter will not be truncatable because we will be considering a fixed or semi-fixed equivalence class, or whether it would not have been truncatable in the free equivalence class, also.

3.7 Remark on the difficulties in directly following our proof-programme 1.21:
As mentioned in 1.21, transformation diagrams will be an intermediary step when it comes to translating the combinatorial information of a transformation pattern back into geometry. In order to construct a homotopy in the course of proving that the weak equivalence relation as defined in B.2.9(i) really reflects the relative homotopy of paths, a homotopy function has to be constructed out of the combinatorial data of the word sequences and their transformation patterns. However, at the present level of having understood the equivalence relations we would run into serious difficulties if we try to take up the project of constructing homotopies out of transformation patterns and transformation diagrams now. This would happen, since by putting the pieces for transformation diagrams in an arbitrary way together, there can arise a lot phenomena which make the construction of homotopy functions rather difficult. Amongst these phenomena are the following:

(i) lines running in a non-monotone way from the top-bar to the bottom bar, i.e. over various maxima or minima.

(ii) Some of the lines may be composed to circular segments which remain entirely in the interior of the diagram and never touch the boundary.

(iii) There may be even some lines running from the left bar to the right bar.

(iv) There may also be lines starting at the left or right vertical bar of the frame and ending at the same bar.

(v) There may also be lines which on their way from the top bar to the bottom bar intersect each other more than once in reversed direction.

In particular the phenomena (i) and (ii) correspond on the combinatorial level to a situation where it might be necessary to make a word more complicated before it can be cancelled down to reach its final destination. We already claimed in the introduction (cf. 0.1) that our equivalence relation of word sequences satisfies that this is not necessary, and that in fact each word can be first cancelled down to a minimal representative of this equivalence class before enlarging operations are applied to make it reach its final destination. In a transformation diagram that is associated to such a sequence of transformations all the minima will be above all the maxima, and hence there cannot be any circular segments as in (ii); and all lines running from the top bar to the bottom bar have to be strictly monotonous in the y-coordinate. In addition, all lines connecting two points with each other that either lie on the top bar or lie on the bottom bar will have to go over precisely one minimum or maximum, respectively. By similar methods of only associating transformation diagrams to nice transformation patterns, we will be able to get rid of the phenomenon (iv) (which corresponds to letters which are appended and truncated later) and the phenomenon (v) which corresponds to phenomena that the same letters may be interchanged more then once. We will not be able to get rid of the phenomenon (iii). However, it will turn out that having such a line in a transformation diagram does not lead to uncontrollable effects.
The above programme makes us now postponing the project of constructing homotopy functions out of a sequence of transformation diagrams, and instead of this we will analyze on a combinatorial basis the properties of our equivalence relations first. After we will have proved the existence of minimal representatives and hence that we can restrict our considerations without loss of generality to transformation diagrams with the properties described above, we will then take up the project of constructing homotopy functions out of transformation diagrams, and by this we will complete the proof that our equivalence relations actually describe the types of homotopies as explained in 1.1.

3.8 Remark on our strategy of proving the existence of almost unique reduced representatives in our weak equivalence classes (cf. Thm.B.2.10(i)):
Note that having unique minimal representatives within each equivalence class and an algorithmic way of constantly minimizing any representative until it coincides with the unique minimal one may seem natural from a naive way of thinking, but it is not in general mathematics. Some disciplines in mathematics (e.g. knot theory, the word problem in general groups, or three manifolds and their surgery representations) appear hard by the lack of an algorithmic way of defining and algorithmically finding a unique minimal representative. Hence even in the most famous example for a theory that has such properties (the calculus of elements in free groups), it needs a certain chain of conclusions (see, e.g. [Ly-S].I.1(page 3)) to establish such properties. As mentioned in 0.1/3.8 we will establish a similar property for our calculus of word sequences, and it even can be understood as generalizing what is standard knowledge for the calculus of free groups. Of course, establishing this generalized result requires a proof, since the situation here is more complicated then for finitely generated free groups: We have more elementary processes, since the theory of free groups does not deal with a set of partially commuting variables or with truncatable letters, and apart from that, wherever we try

to apply our processes to words we have to pay attention that this can be compatibly extended throughout the entire word sequence.

However, subsequently we first will discuss words separately, i.e. deduce that even with the additional transformations B.2.2(ii)–B.2.2(v) unique minimal representatives in each equivalence class exist and that they can be reached by arbitrarily minimizing each word step by step, and then we will discuss in how far this property extends to word sequences (cf. 7.9/7.14). Unlike as for word sequences we will actually get uniqueness when looking at words separately. We will perform the first step by using the same scheme of proof as in [Ly-S].I.1(page 3) for the classical case for the calculus of free generators of groups. This scheme of prove comprises the following steps:

(i) First establish a fixed cancellation algorithm that clearly defines which shortening process is to be applied to a word that permits the application of more than one (cf. 4.1/4.2).

(ii) Deduce that this algorithm does not change a word if it is already short (this will turn out to be a one-line corollary after the corresponding algorithm will have been defined, cf. 4.8).

(iii) Deduce that, if this algorithm is applied to two words which just differ by one elementary transformation process only, then the algorithms then gives the same result (this will require an elementary but lengthy discussion, tracing the corresponding algorithm step by step at the place where the two corresponding words differ from each other, cf. 5.1).

(iv) Then finally conclude that each equivalence class has one unique minimal representative, and that it can be reached regardless whether minimizing processes are applied according to our algorithm or in any other order.

Note that (iv) can be deduced by applying a general scheme of arguments (cf. [Ly-S].I.1(page 3)), they will be explained in the remainder of this paragraph:

The existence of reduced representatives is clear: We have not defined transformations which are neither maximizing or minimizing (by B.2.3), and it is also clear the only minimizing process that does not reduce the number of letters cannot run through a circle. Since there do only exist finitely many words which have the same number or which have fewer letters than the one given which represents an equivalence class, only a finite number of them can be equivalent to the given word, and some of them have to be reduced. What remains to be proven is that there cannot be more than one of these "locally minimal representatives". Assume there would be two of them. Since these two representatives are assumed to be equivalent, there exists a finite sequence of words $\omega_0, \omega_1, \ldots., \omega_n$, such that each two neighbouring words of this sequence differ just by one elementary process. Let $\overline{\omega_i}$ be the corresponding non-minimizable word which our algorithmic cancellation mechanism computes from ω_i. Then (iii) gives that $\overline{\omega_0} = \overline{\omega_1} = \ldots = \overline{\omega_n}$. On the other hand $\omega_0 = \overline{\omega_0}$ by (ii) and similarly $\overline{\omega_n} = \omega_n$. Hence $\omega_0 = \omega_n$ and this contradicts to the assumption that there would exist different reduced words.

Thus we have described a strategy for how to obtain unique minimal representatives for words, however it will still take preparation steps to be performed before we can

set up the algorithm as described above in (i), namely, we have to note that at the present level we have only established equivalence relations for word sequences, but not for the words separately. In addition, it seems appropriate to add a few remarks to the definition of the order of word sequences that was formally already given in Part B (cf. B.2.3/B.2.11). In 3.9–3.11 the corresponding preparation will be completed.

3.9 On the equivalence relations between words:
Since words in this paper are only discussed as elements of word sequences, we have to ask how one could describe the restrictions of the equivalence relations as defined in 1.6 and B.2.1(iii) to words of fixed level indices. Of course, we cannot avoid permitting the elementary transformations B.2.2(i)–B.2.2(v). However, a problem arises since at first sight it appears that the properties 1.6 and B.2.1(iii) are void when restricting them to single words; but on the other hand they had been the essential ingredient in the results 2.3/2.1 that impose a non-trivial condition for the transformations of single words also. On the other hand the results as quoted in the preceding sentence give us a good idea of how we can define our equivalence relation for words: With each equivalence relation for words we think that a list of at most two variables (at most one horizontal and at most one vertical variable) is given, and that if an appendance or truncation of a variables at the beginning of a word should be legitimate, the corresponding variable must be taken from this list. A similar list of at most two variables is given for the truncations or appendances at the end of our word. These lists may or may not intersect.
Hence note that for words, unlike as for word sequences (cf. 3.6), the number of equivalence relations under discussion is not restricted to a small number, but according to the different alphabets from which our words can be taken ($\overline{\mathcal{A}_1}, \overline{\mathcal{A}_2}, \overline{\mathcal{A}_3} \ldots$) and the correspondingly different possibilities of truncatable and appendable variables we are in principle discussing an infinite number of equivalence relations. We will not introduce different notations for for these different relations, since we are hardly in danger of making mistakes due to using the wrong one: In those lemmata where we consider separate words only, we can argue as if fixed lists of truncatable variables are given, and in contexts where words are taken from word sequences and their equivalence is discussed, then according to 2.3 the corresponding list of truncatable variables consists precisely of those corridors that contain the start point or the end point, respectively, of any path realizing our word sequence — at least, when discussing free equivalence relations (cf. 3.6). In the other (semi-free or fixed) cases we can use the same framework, in this case the corresponding list(s) of truncatable variables are just set to be the empty list. That way all (fixed, free or semi-free) equivalence relations are treated by the following lemmata 4.1–7.14, but, since they are treated by adjusting the list of truncatable variables accordingly, they need not be explicitly referred to, even if a lot of cases will be separately discussed within these lemmata.

3.10 Addenda to the definition of the order of words and word sequences in B.2.3/B.2.11:
Recall that in B.2.3 we have defined for each of the elementary operations between our words to be either of shortening or of enlarging type, so that the inverse of a shortening operation is enlarging and vice versa. This then induced a partial order between words,

based on the property whether two words can be transformed into each other by using shortening operations only, or merely enlarging ones, respectively. Similarly we got the corresponding definitions for the order between word sequences by requiring that there exists a compatible sequence of transformations between the two word sequences such that all operations occurring on all of the levels are either of shortening or of enlarging type. Putting the elementary operation into two classes of enlarging and of reducing ones was natural for cancellations, insertions, truncations and appendances, and for transpositions it was based on subdividing pairs of adjacent possibly commuting letters into 8 types according to which of the letters was vertical, which was horizontal, and whether the letters were taken from $\mathcal{A}$ or from $\mathcal{A}^{-1}$. The formal definition was given in B.2.3(1), however according to 1.1(ii)/A.4.6(i) we are now also able visualize the geometric picture of this definition, see Fig.k. Each of the pairs of Formula B.2.3(1) is symbolically represented in one of the subfigures of Fig.k. Each of the subfigures indicates by dotted lines the corridors to be crossed and by solid lines it shows the two alternatives that a path could take to cross both corridors. That way each of the subfigures combines the two product types of one pair of types. The arrow between the two path alternatives which has been drawn in dashed mode finally indicates which of the two corresponding product types is to be considered as the shorter one. The arrow points at the alternative to be considered as shorter.
Note that the moral of our definition is that pushing a path either towards the bottom-left corner or towards the bottom-right corner makes a path shorter. As a consequence of this definition, for a right-angled triangle whose legs of the right angle are horizontal and vertical it might be shorter to go over these legs than to go over the hypotenuse.

3.11 Remark: Observe that by adjusting the pushing directions of paths the way as it had been done in 3.10(Fig.k) we have in principle given up that word sequences behave in a in symmetric way to our symmetric basic square. However, the advantage of this definition is that any transformation appears shortening if and only if the corresponding transformation for the inverse of a word sequence (cf. C.4.9) also is of shortening type. Thanks to this property, after we will have established the existence of reduced word sequences, it will be an immediate *corollary to see that the inverse of a reduced word sequence again is reduced.*

3.12 Remark: Recall what has been proven in 1.9, i.e. that the sequence of transformations that turns the ith of two word sequences into each other, can be extended to the less fine indices, so that the overall sequence of transformations satisfies all conditions of a compatible transformation pattern. However, in 1.9 we had to stress that in the course of such extensions one might have to perform more transformations then just those directly induced from the ith level, even if such additional transformations could only consist of interchanging commuting variables in such a way that this process can be extended by the identity to the $(i-1)^{\text{st}}$ level. Having in 3.10 recalled how important the subdivision of the elementary transformations in shortening and enlarging ones is (cf. B.2.4), it is now time to ask whether it might happen that such an an upward extension to a shortening transformation could not be done without having to use enlarging transformations for smaller indices as well.

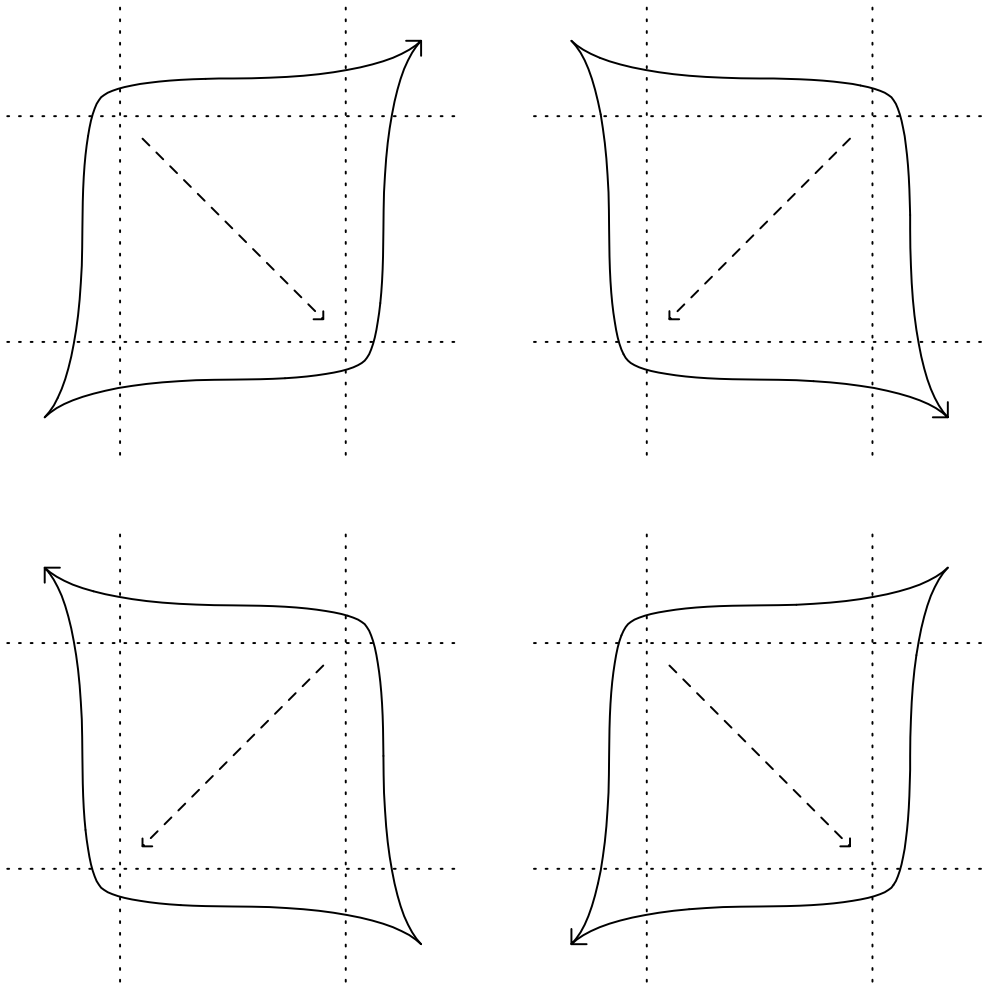


Fig.k (belonging to 3.10):
This figure shows which moves of paths correspond to shortening and which to enlarging transpositions. See B.2.3(iii) for the formal definition of shortening and enlarging and 3.10 for a more detailed description of how to understand this figure.

A partial answer to that question is given by the following lemma 3.13. It is definitely just a partial answer, since, other then in 1.9 we will not be able to cope with any given $(\omega_1, \ldots, \omega_i)$ and $(\eta_1, \ldots, \eta_i)$.

3.13 Lemma: *In the sense of 1.9/3.12 any sequence of shortening operations can be upward extended by using shortening operations only, provided the initial string of the end state, i.e.* $(\eta_1, \ldots, \eta_{i-1})$ *in the notation of 1.9, is not prescribed.*

Proof: As in 1.9–1.11 we can reduce our discussion to filling up the top-right corner of (2×2)-matrices. The proof is lengthy and hence spread over the subsequent Propositions 3.14–3.16 each of which is devoted to handle one of the possible types of elementary operations separately:

3.14 Proposition: *The reduced extension-problem of Lemma 3.13 can be performed in the case of a given truncation on the bottom row of our* (2×2)*-matrix.*

Proof: Without loss of generality, assume that the given truncation is a truncation to the end. According to C.1.5 it can be extended by the identity to the top row, if the affected letter is either a non-split-up letter or just one of two split-up letters both of which are showing up on the bottom row. The remaining interesting case is given when we are truncating the single split-up letter of some root letter, since in this case we must truncate the root letter also to find a compatible extension on the top row. This clearly would give a shortening process; however, the question to be discussed below is whether it is a legitimate one. The following easy observation shows, that it is:
By assumption of this case the disappearing of the root-letter and of the corresponding brace spanned between the two levels of our (2×2)-matrix will not be missed by any letter of the bottom-level. In addition, the disappearance of a brace cannot cause overlaps with other braces that would violate C.1.7. In how far the disappearance of the root-letter messes up the brace structure between the top-level of our matrix and the adjacent upper level will have to be decided at a later state of our inductive process, and hence this is irrelevant for the present discussion. The remaining question is, whether our root-letter actually is an in a truncatable position, i.e. whether it is the last letter of its basic direction and, if it is not the last, whether all letters that come later do commute with our root-letter (note that this condition will also imply that despite having truncated the root letter our word in the top row still satisfies C.1.7). Hence we ask: "Which letters could be placed to the right of our root-letter?" – By C.1.7(i), whatever letters there are, they have to have a brace attached to them. And since by assumption, the braces attached to our root-letter has some ghost-letter in its span, any brace that is attached to a letter on the right hand side of our root-letter must intersect the root-letter's brace. By C.1.10 this proves that all the corresponding letters commute with our root-letter, and hence the root-letter is the last of its basic direction. Hence we found a compatible shortening extension in this case. □

3.15 Proposition: *The reduced extension-problem of Lemma 3.13 can be performed in the case of a given cancellation on the bottom row of our* (2×2)*-matrix.*

Proof: If the given cancellation only affects non-split-up letters, there is not any obstruction for extending this cancellation process by using the identity upwards. If it does, we have to make a distinction, whether both of the affected letters or just one of them is a split-up letter. In case that both are, it is clear that the first/second of our cancelling letters must be a second/first split-up letter, respectively, since the presence of the split-up partners would obstruct the cancellation otherwise. In this case the cancellation process must be upwards extended by cancelling the root letters. The arguments why this will always be legitimate follows by straightforward adaptations from the arguments of 3.14 (indeed, a truncation often appears just like a cancellation against some ghost letter), and hence they will not be exposed any further. So we proceed with the other cases, where we (without loss of generality) assume that the first letter of the

cancelling pair is a split-up letter, but the second is not. As in the preceding case, the split-up partner of the first letter must be placed to the left. Since the second letter of our cancelling pair is from a split-up corridor, the internal compatibility just gives two possibilities why this letter might not be a split-up letter itself: Either the next letter to the right of the corresponding basic direction belongs to the same line of corridors as the second cancelling letter (in this case it has the inverse exponent), or the end of word is reached before some letter of the same basic direction as the cancelling ones has shown up. We postpone the second subcase and proceed with the first. In this case, if our cancellation is performed, the latter letter becomes the second split-up letter, i.e. with respect to the split-up structure it takes over the role that before our cancellation was performed, was hold by the first cancelling letter. This implies that the cancellation process can by the identity be extended to the top-level of our matrix (and hence we to not have do discuss a lot of questions that had to be asked in 3.14), but since some brace between the two levels of our matrix had to be readjusted, we have to ask whether this can violate one of the conditions of Definition C.1.7. Indeed, it cannot, since a closer look tells us that if the brace that had been readjusted by extending its enclosed domain to the right it would now intersect some other brace, they would have to intersect each other before also: If the vertex of the other brace was to the left of the vertex of the readjusted one, the the extension of the domain of our brace to the right had made it even difficult for two such braces to intersect each other, and if the vertex should have been to the right, the extension of our brace to the right is irrelevant for the question, whether a proper intersection takes place (i.e. whether the enclosed domain are disjoint and appear in the reversed order as the vertices). Hence in this case we also succeeded to extend our operation to the top row. The last open subcase of this proposition can be analogously discussed as the preceding structure: With respect to the branching-structure a ghost-letter takes over the role of the new split-up letter, and hence by an analogous readjustment of braces as before the identity can be used to extend our cancellation to the top-row. □

3.16 Proposition: *The reduced extension-problem of Lemma 3.13 can be performed in the case of a given transposition on the bottom row of our* (2×2)*-matrix.*

Proof: Transpositions can clearly be extended by the identity to the less fine levels as long as not both letters that are involved are split-up letters. Even if they are, there can be a lot of situations where despite the interchanging of letters on the finer levels the correspondingly adjusted braces still overlap. In this case again the identity can be used to extend this operation to the upper levels. However, if the corresponding transposition process on the lower level should have been the one that finally makes the overlapping domains of braces not overlap any more, then we have to look at the corresponding root letters. In the lucky situation our root letters might already have been in the order in which they have to be in the situation of non-overlapping braces, in that case we are again done by using the identity. In the other case we have to interchange the corresponding root-letters, and we have to argue in the following that this always is an applicable process:

From 1.15 we know that the corresponding root-letters are commuting. Further, since a single exchange of two adjacent letters cannot have reversed the order of our two brace-domains completely and since the enclosed domains of these braces are assumed to be disjoint afterwards, we see that the corresponding braces intersect before our performing of the crucial given transpositions on the ith level. Now, the geometry of intersecting braces gives that if there should exist a brace which is attached to some letter that is placed between the vertices of the intersecting braces, it has to intersect them both, also. However, by C.1.10/B.1.8 this would imply that the three corresponding letters have all different basic direction, and just two different basic directions are defined. Hence having proven that way that the corresponding root letters are commuting and are in adjacent positions, we now see that our given transposition on the ith level can in any case compatibly extended to the $(i-1)^{\text{st}}$ level, in the latter case by interchanging the corresponding root letters, and in the preceding case by the identity. In particular this implies that a shortening transposition never needs to be extended upwards by an enlarging transposition, since the root-letters have the same exponent and the same basic direction as their split-up letters (cf. B.2.3). □

Since the upward extension problems of 3.14–3.16 all could be solved and hence all types of elementary processes have been discussed, these propositions together with the reduction steps of 3.10/3.11 imply Lemma 3.13. □

§4. A shortening algorithm for words

The contents of this and of the next section are to be understood as the part of our programme that has been explained before in 3.8(i) and 3.8(ii), respectively.

4.1 Description of the algorithm:

(i) Let $\omega = l_1, \ldots, l_n$ be a word in $\overline{\mathcal{A}_i}$, which is to be shortened. At each step of the algorithm one of the letters will be the "letter of consideration". In general this position of our letter of consideration is moving from the left end to the right end of the word. At any intermediary state of the process the segments which lies to the right of the letter of consideration has not been affected by any changes yet, the segment to the left may. At the beginning of the algorithm the leftmost letter, i.e. l_1, is the letter of consideration.

(ii) After a letter has been appointed to be the one of consideration, the algorithm first asks whether this letter may be truncated to the beginning or not. I.e. it asks whether the letter is in the leftmost, leftmost vertical or leftmost horizontal, and, in case it is not the leftmost letter, whether all the preceding letters commute with the one of consideration, and, if so far there still is a chance that the letter may be truncated, if it is in the corresponding truncation list. If the truncation can be performed, it is performed, and then the next letter on the right becomes letter of consideration (in this case the process (ii) is to be applied to that letter). In the other case our letter of consideration is not to be changed. We then proceed with (iii):

(iii) Let l_n be the letter of consideration. Search for the next letter to the left of l_n that is of the same basic direction as l_n. If there is such a letter, denote it by l_k; otherwise move the letter of consideration one step to the right and proceed with (ii). If some l_k is found, we then discuss whether l_k could be cancelled against l_n. This depends on whether l_k is the inverse of l_n (if not, it cannot be), whether it is placed adjacent to l_n, or if not in an adjacent position, whether all letters between l_k and l_n commute with $l_n = l_k^{-1}$. If the the first one and of the two latter questions could be answered by "Yes", then the cancellation is performed, and after having appointed l_{n+1} to be the letter of consideration now, the algorithm proceeds with (ii). If the cancellation could not be performed l_n remains the letter of consideration, and the algorithm proceeds with (iv).

(iv) Discuss whether l_n shall be permuted with its predecessor. We will call this predecessor "l_{n-1}", even if due to some preliminary operation "$n-1$" might not be the index that had been associated with the corresponding letter at the beginning of the process. If $n = 1$, of course, the algorithm proceeds as if the answer to this question is "No!". The answer has, of course, to be "No!" also, if l_n and l_{n-1} do not commute, and, if it would not be shortening process if these letters were interchanged (cf. B.2.3(iii)). If it is decided not to perform the corresponding transposition, l_{n+1} is appointed to be the letter of consideration, and then proceed with (ii). If the letter of consideration has already been the last, the process continues with the final step (v). If it is a shortening and legitimate process to perform the corresponding transposition and hence the corresponding part of our word has been changed into "$l_{n-2}\, l_n\, l_{n-1}$", the algorithm now enters a subloop of the main loop of this process: It then applies the same criteria as explained above to decide whether it may be desirable also in interchange l_{n-2} with l_n, and, if it should be, it does it and then discusses whether also to interchange l_{n-3} with l_n, and so on. This process must get stuck after some iteration steps, either because l_n had been moved to the leftmost position of our word already, or, since the next transposition that could be performed to l_n is either illegitimate or would be of enlarging type. If this process is stuck we then appoint l_{n+1} (the position of this letter had been unaffected by all the preceding transpositions) to be the next letter of consideration, and then proceed with (ii), or by the same criteria as before we proceed with (v).

(v) Finally our algorithm asks, whether the last letter of our word can be truncated to the end, and whether the last letter before that one that belongs to the other basic direction can be truncated to the end of our word. If these questions have decided by similar criteria as in (i) for the truncations to the beginning of our word, and if the corresponding truncations have been performed, then our algorithm terminates.

4.2 Lemma: *The above algorithm always terminates after finitely many steps. At the time it terminates the given word had been transformed into a short word, i.e. into a word to which none of the reducing operations B.2.2(i)–B.2.2(v) could be applied any more.*

Proof: The first claim of this statement is easy to be seen: Our given words are finite and by definition our algorithm has not any opportunity to run into an endless loop.

The second claim is not so evident and will not be proved before 4.7. On the one hand we have to admit that our algorithm seems to have checked each possibility because it has moved the letter of consideration step by step from the left end to the right end of our word, and it has checked each of the possibilities for applying one of the shortening transformations to each letter while this letter was under consideration, but on the other hand also some changes had been applied to that letters of our word that had been placed at some distance to the left of the letter of consideration. Hence there seems to be a fairly realistic possibility for how our algorithm could leave us with a word which still can be shortened by some operation O_0: We have to assume that at the time our algorithm had checked for O_0, this operation had been still illegitimate, and at a later state it had become legitimate due to some of the changes that our algorithm imposed to our word at that state.
Hence we will prove the second claim of our proposition by contradiction. We now assume that our algorithm leaves us with a word to which

(1) still one particular shortening operation O_0 can be applied.

In the following we will trace, step by step, the changes, that our algorithm has imposed to our word, and search for

(2) that operation O_1 which, when viewed from tracing-back direction, obstructs to perform the operation O_0

in time, or, when viewed from the direction into which algorithm runs, "gives us" (i.e. removes the last obstruction for performing) O_0. We will come to a contradiction by taking a closer look at the corresponding situation, i.e. we will ask for each of the types of an elementary operations: "What are the types of obstructions for performing such an operation O_0, that could possibly get removed at a later state in the course of running our algorithm?", and we will discuss for each of the possible obstructions that according to the way our algorithm is set up, such obstructions could have only be removed before our algorithm had checked for the possibility to perform O_0. This will be done in the following four propositions 4.3–4.6 separately for each of the possible type of an O_0-operation. Each time we will assume that our algorithm leaves us with a word to which still a shortening O_0-process can be applied, and each time we will find a contradiction. Hence the proof of 4.2 will not be complete before 4.7.

4.3 Proposition: *The Algorithm 4.1 cannot terminate with two letters still remaining in a cancellable position.*

Proof: Assume the contrary and denote the corresponding letters by l_i and l_i^{-1}. These letters need not be in an adjacent position, however at the time our algorithm terminates only letters commuting with l_i may be placed between them. What could have obstructed a cancellation of l_i and l_i^{-1}? – It must have been some non-commuting letters that are placed between l_i and l_i^{-1}, and these letters must have disappeared at a later state of the algorithm. What could have been the operation O_1, that made these letters disappear? – It could not have been by a transposition, since only letters commuting

with l_i could have been disappeared from the region between l_i and l_i^{-1} by an interchange with l_i or l_i^{-1}. For similar reasons it could not have been a truncation, since, if a letter which is placed between l_i and l_i^{-1} can be truncated, it must have been a letter which commutes with l_i. So the operation O_1 must have been another cancellation. And both of the two corresponding letters to be cancelled (we will call them l_j and l_j^{-1}) must have been placed between l_i and l_i^{-1}, because, since they do not commute with l_i, they could have never get cancelled, anyway. But our algorithm checks at most once whether a pair of letters is to be cancelled, namely, when the second of the corresponding letters is the one of consideration. Hence assuming that l_j and l_j^{-1} are placed between l_i and l_i^{-1}, then our algorithm would have checked the $l_j^{\pm 1}$-pair before checking the $l_i^{\pm 1}$-pair, and this contradicts to our assumption that the $l_j^{\pm 1}$-cancellation is an obstruction that prevents the $l_i^{\pm 1}$-cancellation from being performed. □

4.4 Proposition: *The Algorithm 4.1 cannot terminate when two letters are still in a position such that a shortening transposition can be applied to them.*

Proof: Assume the contrary, and note that the corresponding two letters must have been at all states of the algorithm in the same order, because our algorithm does never perform an enlarging transposition. Hence, again our obstruction must be of that type that at some earlier state of the algorithm the corresponding two letters, which shall be denoted by l_k and l_m, have not been in the adjacent position at which we find them when our algorithm terminates. Hence we ask: "What processes might have made the last one or two letters disappear that, as we assumed, had been placed between l_k and l_m at an earlier state of the algorithm?" – In principle we have seen that full repertory in 4.3. This repertory consists of the possibility that a cancelling pair of letters might have disappeared from the place between l_k and l_m (this case can be ruled out with the same argument as given in 4.3) and of some other operations that assume that there was only one letter, say l_j , placed between l_k and l_m, and that this letter must commute with at least one of l_k , l_m. Without loss of generality assume that the letter commutes with l_m, and hence is of the same basic direction as l_k. Having with "$l_j\, l_m$" two letters of the same basic direction following each other, our internal compatibility property C.1.6 of words leaves us with only two possibilities: Either $l_j = l_k^{-1}$, or that l_j and l_k stand for acts of crossings in the same direction of neighbouring lines of corridors. In the first case we find that at some time when the letter of consideration had already been left of "$l_k\, l_j\, l_m$" our word has had a cancelling pair, and we then continue our discussion with the question, why at this time this pair had not been cancelled. This question will lead us to a contradiction with the same arguments as given in 4.3. In the second case B.2.3(iii) and our assumption on l_k and l_m imply that interchanging l_j and l_m would give a legitimate operation of shortening type. In this case we continue our discussion by considering the case that is was some (far-distance) truncation or (far-distance) cancellation, that had made l_j disappear from its place between l_m and l_k. We ask why at the time where it had disappeared, the interchange between l_j and l_m had not been performed. This brings us possibly into a loop of playing the above arguments of this Proposition 4.4 again and again (finding that there had to exist another letter

between l_j and l_m before), but this loop cannot be endless, because each time at which the above argument has to be reapplied it refers to a shorter sequence of transformation processes in the run of our algorithm. That way we get rid of all possibilities apart from that is was an interchange that made l_j disappear from the place between l_k and l_m. So let now consider the case that such a disappearance of l_j is our O_1-process (cf. 4.2(2)). However, our algorithm only performs transpositions if it is in the state 4.1(iv) and according to the way our algorithm is set up, at the time our O_1-process shall be performed, either l_j or l_m or some letter to the right of l_j and l_m must have been the letter of consideration. The corresponding letter of consideration might definitely have been at some distance apart from l_j and l_m, since the O_1-process might be performed in the course of the subloop of 4.1(iv). However, the subloop-construction of 4.1(iv) guarentees that in any case l_k and l_m would have been interchanged, also. Hence 4.1(iv) is set up in such a way that the corresponding regions that either contain the letter of consideration, or, on which the subloop has operated, are not left with interchanges that still might be performed. □

4.5 Proposition: *The Algorithm 4.1 cannot terminate with some letter still remaining in a position where it is truncatable to the beginning.*

Proof: In principle the situation for a truncation to the beginning of a word is the same as for a cancellation between the corresponding letter to be truncated and some "ghost-letter" in the zero-position. Hence the same arguments as in 4.3 can be applied to rule out that our algorithm leaves us with some letter that is truncatable to the beginning of our words.

4.6 Proposition: *The Algorithm 4.1 cannot terminate with some letter still remaining in a position where it is truncatable to the end.*

Proof: Assume the contrary. Since our algorithm checks the possibilities of truncations at the end of our word at the very last step, any O_1-operation could also just be a truncation to the end (cf. 4.2(2)). Nevertheless, such a situation must carefully be considered, because our algorithm is set up in such a way that it only checks two variables for the possibility to be truncated, and then it terminates. Hence it needs a closer look to rule out that this does not leave us with additional possibilities for truncations to the end of our word.
Hence let us assume that we find a letter that still could be truncated to the end of our word: For having a general picture let us assume that "$l_j\, l_{i_1} \dots l_{i_n}$" is the end segment of our word as it is left when our algorithms terminates, and hence at the state when our algorithm has entered Step 4.1(v) there may have been one or two additional letters to the hand right side of l_j. We call them l_k and l_m. There must have been at least one of these letters, because otherwise our algorithm would have detected that l_j could be truncated. If we assume that only one of these letters had actually been there, it must have been of the same basic direction as l_j, because otherwise the truncatability of l_j again would have to be detected. Then by the internal compatibility of words C.1.6 and the fact that according to 3.9 the corresponding list of truncatable variables

contains at most one item in each basic direction, it follows that this letter l_k must be l_j^{-1}, and hence that this letter is in a cancellable position with respect to l_j at the time our algorithm has entered Step 4.1(v) . Hence we can continue our discussion by asking, why our algorithm failed to perform this cancellation, and then come to a contradiction by using the same line of arguments as in 4.3.
If then we assume that at the time our algorithm had entered Step 4.1(v), our word reads "$\dots l_j\, l_{i_1} \dots l_{i_n}\, l_k\, l_m$", and that either l_k is of the same basic direction as l_j, or that l_m is of the same basic direction as l_j and l_k is commuting with l_j, then the same line of arguments as in the preceding sentence can be applied (note that in this case we could also have the l_k placed before some of the l_{i_ν}. We are hence left with one assumption to be discussed, which will turn out to yield a direct contradiction to the internal compatibility of a word corresponding to C.1.6. That is that l_m is of the same basic direction as $l_j\,, l_k$ consequently of the same basic direction as the l_{i_ν}, but other than the l_{i_ν} it does not commute with l_j : Without loss of generality assume that l_k corresponds to the crossing of a horizontal line of corridors being situated around the x-axis of the (x, y)-plane and that the orientation of this act of crossing is "upward". This horizontal line of corridors must intersect in a hole with the vertical line of corridors containing l_m. Consequently the vertical corridor l_m has its lower end segment at a hole at the zero-level of the y-coordinate, and hence the area covered by this corridor lies entirely in the space of positive y-coordinates. This would imply that all the l_{i_ν}-corridors, which for being commuting have to intersect the l_m-corridor, would also have to lie in the space of positive y-coordinates. On the other hand, the internal compatibility of our word gives that, since we assumed l_m to stand for an upward directed crossing, that these corridors must be placed below the x-axis. This contradiction implies 4.6. □

4.7 Remark: By having handled all possible cases in Propositions 4.3–4.6, the proof of 4.2 is now complete. □
Hence we now know that our Algorithm 4.1 is capable to transform each word into a reduced word by applying shortening operations in a certain order only. Observe that this result only holds as long as we stick to the given order as defined in 4.1, even though in 3.8(iv) we claimed that the same reduced word can be reached regardless in which order shortening operations are applied to a word. This fact still needs to be proved, and this will be the topic of the next Section §5.

4.8 Corollary: *The algorithm 4.1 does not impose any changes to a reduced word.* This is a immediate consequence of the fact that our algorithm just consists of searching in a particular order for facilities for reducing a word and performing the corresponding changes. Consequently, if a word does not permit to be shortened, our algorithm will not come up with any changes. □

4.9 Remark on the phenomenon of *possibly iterated truncations:*
The situation that we find a sequence of letters $l\, l^{-1}\, l\, l^{-1}\, l \dots l\, l^{-1}$ at the beginning or at the end of a word where l is a truncatable letter might seem worth an extra consideration. The question is: "Should these letters be removed by and iterated truncation if the word is to be reduced, or should they better be removed by several cancellations with at most

one letter remaining for a truncation?" – If the word of consideration is a part of a word sequence, then the second solution is the only one that can work, since the first solution could never be extended throughout the entire word sequence. For, apart from the outermost letter each other letter has the full set of split-up letters. If we would attempt to truncate all these letters, according to B.2.7(iii) this could only been extended by truncations throughout the word sequence, and this cannot work for two reasons. At first, the number of letters to be truncated is unbounded, and at second for each level of the word sequence at most one of the split-up letters might be truncatable (cf. 2.3). Hence if our word sequence should permit to get rid of all these letters, with at most one exception this has to be by cancellations. On the other hand, if we are not discussing the corresponding word as a part of a word sequence, but separately based on the equivalence relation defined in 3.9, there is not any obstruction to get rid of all these letters by iterated truncation. What has been explained in the previous sentences does not give a contradiction to this fact, it just proves that the method of iterated truncations must be non-extendible.

In this context one should observe that a lot of shortening operations that could be applied to a separate word might be non-extendible if this word is put into a word sequence. The case discussed above was somehow particular, since in that case a general argument exits which tells us that this type of operation can never be extendible, while in the usual case one will have to take a look at the entire word sequence before one can decide such a question. When viewed from that side, it is not the purpose of the above algorithm 4.1 to find an extendible way of shortening a word to a reduced word (by the way, not each word of a reduced word sequence needs to be reduced), the following steps 7.1–7.14 will be devoted to that task. In that context of this paper it is just the purpose of the algorithm 4.1 to prove that there exists at least one way of coming to a reduced word by using shortening operations only, and in the overall context one has to take into account that the way found by our algorithm may not be extendible through a concrete word sequence (and in some cases this even can be seen directly by general arguments). The case of having such a sequence $ll^{-1}ll^{-1}\dots ll^{-1}$ of truncatable letters at the beginning or at the end of our word is such a case: Note that our algorithm was made up in such a way that it mainly performs cancellations if this sequence appears at the end of a word, but if it appears at the beginning of our word it definitely performs the non-extendible version of iterated truncations.

§5. Compatibility of the Algorithm 4.1 with the elementary processes

5.1 Lemma: *The Algorithm 4.1, if applied to two words w and w' which just differ by one of the elementary process B.2.2(i)–B.2.2(v), reduces w and w' to the same short word w''.*

Proof: The proof of this lemma is lengthy and is comprised of all paragraphs of this section. It is a combinatorial proof which rests on looking at all possibilities for how w

and w' could differ separately, and on tracing the algorithms in each of these cases. As a first step we will in 5.2 explain the principles that are common to all subcases to be discussed, and then we will look to those subcases in separate propositions:

5.2 On the scheme of proof of 5.1:
Without loss of generality we will assume in 5.3–5.25 that the elementary process that brings one from w to w' is a shortening one, i.e. w' differs from w either by interchanging two letters in a shortening way (cf. B.2.3(iii)) or by deleting one or two letters. In the remainder of this introductory paragraph we restrict our language to the deletion-case, even if most of the statements will also hold if w and w' differ by one elementary transposition. Roughly speaking, the proof rests on discussing a lot of cases separately, and it is the purpose of the following paragraph to explain the common outline of each of these cases:
We do not have to discuss the cases separately in which the letters to be deleted in w can be found at the beginning, in the interior or at the end of w, since the first and the last of these cases can be regarded as a special case of the middle one in that sense that the subword comprising the letters that precede or follow the ones to be deleted is empty. Hence we do not loose generality if we think that in some initial phase our algorithm clearly runs the same way if applied to w and to w' : For, our algorithm moves the letter of consideration from the left to the right and it decides whether to apply shortening operations just by looking at that part of the corresponding words that is left of the letter of consideration. Hence it cannot apply different processes to w and w' as long as the first of the letters of w to be deleted has not become the one of consideration. At this state clearly our two words still differ by the same elementary process as they did when originally given, even if the subwords to the left of the crucial place might have been altered. Whatever will be said about these subwords in the remainder of this proof refers to how they look like at the present state of our algorithm (i.e. when in w the first of the letters to be deleted has just been appointed to be the one of consideration and in w' the same for letter with the same position number); in that context it is irrelevant how w and w' had looked like in the start state of our algorithm.
In the present situation a smooth run of our two algorithms would be given, if the next applicable operation that our algorithm detects for w should be the one that brings w to w', so that after having performed this operation the w-run of our algorithm sees that same word and is in the same state as the w'-run in the state described before. In that case we could from the deterministic character of the algorithm conclude that the the two runs terminate with the same word, also. Unfortunately, what has been described in the *preceding two sentences is just the trivial amongst a lot of non-trivial cases,* the non-trivial ones arising by the assumption that the letters of w to be deleted permit other shortening operations apart from the one that brings w into w', and that our algorithms checks and performs these operations before it checks for the operations that brings w into w'. However, the key argument will be the same as in the trivial case: We will discuss all these possibilities separately, and we will find for each of them that after a few steps the w-run of our algorithm sees the same word and is in the same state as the w'-run is after having performed some slightly different steps.

Hence, the four main cases, that have to be separately discussed in the following are defined by the type of operation that turns w into w'. In this context we need not to discuss far-distance cancellations or -truncations, since each such operation could be replaced by sequence of transpositions and an elementary cancellation, and the latter operations will be discussed. Each of these four main cases may require the discussion of some subcases as explained before. Usually these subcases will be defined according to what type of operation (if any) can be applied to the first letter of those to be deleted inside w. Other than for the main cases we are not able to exempt far-distance operations in that context. Of course, there could be more then one type of operation applicable to the leftmost of the letters to be deleted. However this does not cause that additional cases have to be discussed. For, we can define of subcases according to what type of operation is the one detected by our algorithm since by 4.1 our algorithm checks for the various types in a clearly defined order. Further, we hereby agree to simplify our language accordingly: In 5.3–5.25 we will use expressions like "a cancellation/transposition can be applied to l" for "the first shortening operation that our algorithm finds being applicable to l is a cancellation or transposition", respectively.

5.3 Proposition: *The Algorithm 4.1, if applied to two words w and w' which just differ by a truncation to the beginning, reduces w and w' to the same short word w''.*

Proof: Following the scheme as explained in 5.2, there are no subcases to be discussed in this case, since our algorithm always checks for that type of operation at first place. I.e. if inside w the corresponding letter is appointed to be the one of consideration, our algorithm detects that it is truncatable and truncates it. When the next letter is appointed to be the one of consideration, the two runs of our algorithm agree. □

5.4 Remark *on how the algorithm 4.1 might react if w and w' differ by a cancallation:*
Denote by ll^{-1} the corresponding cancelling pair of w. Since our algorithm cannot detect this cancellation unless l^{-1} has been appointed to be the letter of consideration, we have by 5.2 to discuss subcases according to all types of operations for l. However, if none of of them is possible for l, then we are in the trivial case discussed above: Then our algorithms finds that l cancels against l^{-1}, since it checks only for one other operation before, namely whether l^{-1} is truncatable to the beginning. But this is impossible as long as l is not gone. Note that for the same reasons l cannot be truncatable to the end (at least not in the crucial state of our algorithm, cf. above 4.1(v)). Hence we are done by discussing in the following three subcases in 5.5–5.7 as separate propositions:

5.5 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a cancelling pair of letters so that these letters could also be truncated to the beginning* (cf. 4.9).

Proof: If this truncation is performed, then in its w-run our algorithm will detect, that l^{-1} is truncatable, also, since all conditions are satisfied: l^{-1} is clearly in the list of truncatable variables, there cannot be an obstruction for performing this truncations to the left of the former l, the obstruction originally given by l is gone, and this truncation

is the first operation for which our algorithm checks. After the l^{-1}-truncation will have been performed, the w- and w'-run of our algorithm have to agree. □

5.6 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a cancelling pair of letters so that the left of them could also be cancelled in another way.*

Proof: Recall that by 5.4 we wanted to denote by "ll^{-1}" the cancelling pair that does occur in w but not in w'. According to our particular assumption now, to the left of l there has to exist some other letter l^{-1} ("the left l^{-1}"), which should not be mixed up with the l^{-1}-letter to the right of l. If the left l^{-1} is in an adjacent position with respect to l and when the cancellation between l and the left l^{-1} can be performed, then clearly the w-run of our algorithm is in the same state as the w'-run was when the left l^{-1} had been appointed to be the letter of consideration. Hence we are done in this subsubcase. It remains to be discussed in what way our algorithms behaves if the left l^{-1} is in a non-adjacent position to l, i.e. if a finite sequence of commuting letters are placed between that l^{-1} and l. There are two subsubcases to be discussed separately:

(i) All of these letters have the same exponent. In this case we may assume that it is not a shortening operation to interchange the left l^{-1} with the leftmost of that sequence of letter, since this would have been done by our algorithm before. This gives the following clear picture of how the w-run of our algorithm will proceed after having cancelled the left l^{-1} against l : The right l^{-1}, which then is appointed to be the letter of consideration cannot be truncated to the beginning, since this would otherwise have been done to the left l^{-1}. For similar reasons there cannot be a cancelling partner for our right l^{-1} somewhere on the left hand side of our picture. However, our algorithm will find it a sequence of shortening operations to move the right l^{-1} to the position of the former left l^{-1}. And when this is done, the w-run and the w'-run of our algorithm will agree.

(ii) Now let us discuss the case that there are different exponents amongst the letters between the left l^{-1} and l. We will find that this implies that the corresponding sequence of letters has a cancelling pair, and hence this case yields a contradiction, since our algorithm, while passing over the corresponding sequence would have detected and eliminated this cancelling pair (cf. 4.1(iii)/4.3):
Within our sequence of letters find two adjacent ones with different exponents. From the compatibility of word sequences (cf. C.1.6) it can be concluded that these letters belong to the same line of corridors. Now assume, that we do not have a cancelling pair, i.e. the corresponding letters are different letters from the same line of corridors. The geometrical picture in this situation (recall that the internal compatibility demand C.1.6 is just an encoding of this picture) gives that our path between its exit from the first act of crossing on its way to the second one has to pass the hole that separates the two different corridors on its back side. This passage implies the crossing of some corridor of the other basic direction, and hence we found a contradiction to the assumption that we are discussing a sequence of letters all of which do commute

with l and hence have to be of the the same basic direction opposed to that of l. Hence this case can actually not be given.

Since in Case (i) our conclusion could be shown and Case (ii) was contradictory, this proposition is now proved. □

5.7 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a cancelling pair of letters so that the left of them is transposable by a shortening transposition.*

Proof: The corresponding transpositions are performed by the w-run of our algorithm and leave our word at some intermediate state with l being placed further to the left than at the time it had become the letter of consideration. Our algorithm then proceeds by appointing l^{-1} to be the letter of consideration. It finds l^{-1} to be not truncatable to the left (l, even if in a new place, is a sufficient obstruction), but it finds that l is a cancelling partner. Since l had only been moved over commuting letters, by an act of far-distance cancellation l can still be cancelled against l^{-1}. And after that will have been performed, the w-run will agree with the w'-run. □

5.8 Remark *on how the algorithm 4.1 might react if w and w' differ by a transposition:*

We will denote the letters to be transposed by l_1 and l_2, i.e. w sees "$l_1 l_2$" and w' sees "$l_2 l_1$", where $l_1 l_2 \mapsto l_2 l_1$ is a shortening operation.

This case in principle requires the discussion of 25 different subcases, since we have to make a distinction according to which operations might be applicable to l_1 when in the left position of our interchanging pair and which might be applicable to l_2 when in the same position. In principle, each of these combinations defines a different subcase, and there are four types of elementary operations, and the assumption that none of them is applicable is a fifth case. This makes $5^2 = 25$ cases. In general they cannot be reduced by symmetry arguments to $\frac{5 \cdot 6}{2} = 15$ cases, since the fact that the interchange of our two letters is either of shortening or of enlarging type depending on the precise sequence of our two letters prohibits a general symmetry of our cases. However, we will not discuss the 25 cases separately, but we will group together those where similar arguments can be applied and will handle in 5.9–5.17 each group of cases as a separate proposition:

5.9 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by an elementary transposition where one of the two letters is truncatable to the beginning when in the left position.*

Proof: The truncatable letter must also be truncatable to the beginning when in the right position of our interchanging pair, since the other letter, because commuting, cannot give an obstruction for truncatability. If the truncatable letter comes at first place, our algorithm will truncate it and then perform to the second letter whatever shortening operation might be performed to this letter if it would have come first. On the other of our two words where the non-truncatable letter comes first, the same operations are applied to it at first place. Since the result of these shortening operations cannot mess up truncatability, after these operations have been performed, our algorithm then

detects that the other letter can be truncated and deletes it. Then both runs of our algorithm do proceed in the same way. □

5.10 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by an elementary transposition where one of the two letters is truncatable to the end when in the left position.*

Proof: As explained at the end of 5.2, this implies that no other shortening operation can be applied to this letter if it comes at the first place of the interchanging pair. It is an easy consequence, that, apart from possibly performing the corresponding interchange, a shortening operation can also not be applied to our letter, if it comes at the second place of our interchanging pair (truncations to the beginning and cancellations are not obstructed by the first letter of our pair, and transpositions only depend on the left adjacent letter, i.e. require that our pair is interchanged before). Hence the discussion of this case does not need more arguments as the discussion of the two cases below, where we will assume, that to one of our two letters a shortening operation cannot be applied at all, since by 4.1(v) our algorithm does not ask for such a truncation till it has reached its very last phase and has moved the letter of consideration beyond the rightmost letter of our word. I.e., since the two subsequent cases will give that the two runs of our algorithm operate by the same states on the same words if they have once passed over our interchanging pair, we can be sure that the same decision on our truncatable letter is made in the final phase of the corresponding runs of our algorithm.□

5.11 Proposition: *With notations as introduced in 5.8, the Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by an elementary transposition where a shorting operation cannot be applied to l_1 when in the left position.*

Proof: In this situation the w-run of our algorithm might in its next step either apply the same cancellations to l_2 that could be applied if l_2 had occurred at the first place of the interchanging pair, or it might interchange l_1 and l_2, and then perhaps move l_2 further to the left. In any case l_2 will be moved to the same place to which it goes in the w'-run, where it comes at first place. One aspect still needs to be shown in this paragraph, namely that to l_1, when inside w' coming at the second place, shortening operations cannot be applied. Despite the assumption of this Proposition 5.11 such operations might be applicable if whatever has happened to l_2 has given way to them.
The possibility of a cancellation of l_1 can be ruled out in this context, since the possible move or disappearance of l_2 or its cancelling partner cannot haven given way to such an operation (i.e. l_1 could have been cancelled if in the left position, also, and this would contradict to the assumption of this Proposition. However, if the predecessor of the first letter of our interchanging pair should have been some letter l_2^{-1}, then in the w'-run this letter disappears and l_1 sees another left adjacent letter as in its original position within the w-run. Hence we have to discuss whether at this state of the algorithm transpositions could be applied to l_1. So we ask what the predecessor of l_2^{-1} could be. Since it must be from the same basic direction as l_2 if some interchange with l_1 should be applicable, by 5.6(ii) there are just two possibilities left: it could be some

letter l_2 again, or it could be a letter ${l'_2}^{-1}$. Since the first possibility can be ruled out immediate (it would leave us with some cancelling pair in that part of our words over which our algorithm has passed already), we are left with the second one, i.e. that we find some letter from the neighbouring lines of corridors as l_2 with the same exponent as l_2^{-1}. However, if we assume that it was a shortening operation to interchange "${l'_2}^{-1}l_1$", it would also be a shortening operation to interchange the pair "$l_2^{-1}l_1$", that can be found in the w-run at the time our algorithm has entered "l_1l_2". But this then would contradict to the assumption of this Proposition 5.11. Hence l_1 remains unaffected even in the w'-run, and hence the two runs of our algorithm will agree after having passed over our interchanging pair in this case, also. □

5.12 Proposition: *With notations as introduced in 5.8, the Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by an elementary transposition where a shorting operation cannot be applied to l_2 when in the left position.*

Proof: By assumption, we know that our algorithm in its w'-run passes over l_2 on to l_1 without imposing any changes. If l_1 should be cancellable, it performs the same cancellations to l_1 as it does perform in the w-run, when l_1 occurs in the left position of our cancelling pair. If l_1 should not be cancellable, in this case our algorithm cannot perform any transposition to l_1 in the w'-run, since l_1 is assumed to be in the short order with respect to its predecessor l_2, already. Hence in the following we have to give some arguments, that l_1 cannot be moved further to left in the course of the w-run. If it could, then with similar arguments as in 5.11 we have that the predecessor of l_1 could only be l_2^{-1} or l'_2. Since the first case contradicts to the assumption of this Proposition 5.12, we are left with the case where the w-run of our algorithm sees "$l'_2l_1l_2$", i.e. where the predecessor of l_1 is from same basic direction and the neighbouring line of corridors as l_2 and has the same exponent. This gives that, even if "l'_2l_1" should be a commuting pair, it would not be a shortening operation to interchange them, by assumption of what has been the shorter order of our interchanging pair "l_1l_2".
Similar, as in the preceding Proposition 5.11 we are left with discussing that l_2, when being the letter of consideration in course of the w-run of our algorithm, cannot be affected by this algorithm apart from being interchanged with l_1 (if that letter should still be there). l_2 clearly cannot have a cancelling partner (the existence of such a partner would contradict to the assumption of this proposition, the letter would have also been found in the course of the w'-run), but we still have to discuss whether the possible disappearance of a cancelling partner of l_1 does make it possible to move l_2 further to the left as it has been placed in the w'-run (where this cancelling partner had disappeared after our algorithm had passed over l_2 already): Of course, a further moving of l_2 cannot take place if l_1 should have been gone by a far-distance cancellation, since then l_2 sees the same left adjacent letter as in the w'-run. Hence we have to assume the that left neighbouring letter if our interchanging pair is l_1^{-1}. Now according to 5.6(ii) the left neighbour of l_1^{-1} can only either be some other letter l_1 or it could be ${l'_1}^{-1}$, since it has to be of that basic direction so that it could be commuting with l_2. Now the "l_1"-case can be ruled out, since the short parts of our words would then contain a cancelling

pair, and hence we are left with asking, whether some interchange of "$l_1'^{-1} l_2$" could also take place. Since according to the use of notation as introduced in the preceding cases, $l_1'^{-1}$ has the same exponent and belongs to the same basic direction as l_1^{-1}, if interchanging "$l_1'^{-1} l_2$" as discussed in the present situation was of shortening type, then interchanging "$l_1^{-1} l_2$" would also be. But since in the relevant case these two letters can be found in the w'-run at the entering of the crucial state (cf. 5.8), this would yield a contradiction to the assumption of this Proposition 5.12. Hence there is not any chance that in the course of the w-run l_2 is moved further to the left than to the first position of our interchanging pair "$l_1 l_2$". Hence in this case after having passed over the interchanging pair both runs of our algorithm will work in identical states over identical words. □

5.13 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by an elementary transposition where a cancellation process can be applied to both of our letters when in the left position.*

Proof: Since our two letters are of different basic direction, the corresponding cancelling partners must be different letters, too, hence the two cancelling processes do not interfere. This case can quickly be discussed by noting that in both runs of our algorithm both cancellation processes are performed:
Since l_1 and l_2 are commuting, their order cannot obstruct our algorithm from finding their cancelling partners; and since the algorithm is checking the possibilities for cancellation at the second place already, there is only one chance for our algorithm to make a decision different from cancelling the two letters of our interchanging pair in that order in which they appear in w or w' : Namely, if after the first cancellation is performed, the other letter would suddenly have become truncatable. But this cannot happen, since our letter is not the leftmost of its basic direction, there is at least its cancelling partner on the left hand side of it! □

5.14 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by an elementary transposition where, when in the left position, a cancellation process can be applied to one of our letters and a transposition to the other.*

Proof: From the calculus of basic directions we get that the corresponding cancelling partner must be in the left adjacent position to our interchanging pair. For, if our cancelling pair would disappear by a far-distance cancellation (even if the corresponding partner appears at the left position of the interchanging pair), then the letter to the left of our interchanging pair would belong to the other basic direction as the cancelling letter, and hence belong to the same basic direction as the moving letter. But according to B.1.8/B.2.2(ii) a letter cannot be moved over letters of its own basic direction. Hence the two runs of our algorithm appear as follows in this case:
If the moving letters comes at first place, it is at least moved over the cancelling partner of the other letter, perhaps it can be moved even further. Then the cancellation of the other letter and its partner are performed. For the other run these cancellations are

performed at first place, and then the second letter of our interchanging pair becomes the letter of consideration. Its left adjacent letters are, apart from the first one that has been deleted already, precisely those that it also sees in the w-run, and this gives that it is now moved precisely as far to the left as it is moved in the other run. Hence in this case also, after having passed over the interchanging pair the two runs of our algorithm of consideration are in identical states while operating on identical words. □

5.15 Remark: By the assumptions as discussed above in 5.9–5.14, apart from one exception all of the 25 subcases of the cases as described in 5.8 have been covered (some have even been covered twice). The single exception is given by the assumption, that to each of our two letters, if it appears at the left place of our interchanging pair, other interchanges with letters left of it could be performed. But this assumption immediately yields a contradiction. For, the calculus of basic directions directions give that the letter that appears in both runs of our algorithm to the left of our interchanging pair has to be of different basic directions than both of the interchanging letters. However, since those letters already occupy different basic direction and the range of values for basic directions is two by definition B.1.2(i), this case can never occur.

Now for proving 5.1 (cf. 5.2) we are left with discussing what happens if w' results from w by truncating one of the w-letters to the end. Again we have to ask whether our algorithm might find other operations applicable to the truncatable letter l than the corresponding truncation, and this gives the subcases to be discussed in the following:

5.16 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a truncation to the end where the corresponding letter is also truncatable to the beginning.*

Proof: This turns out to be the only really lucky of the remaining cases, since in that case the truncation to the beginning is performed in the w-run, and then after that the w-run and the w'-run operate in identical states on identical words. In the remaining cases we will not find the two runs of our algorithm operating on identical words after having passed over the crucial state, and hence we usually will have to spend more care on the various subcases than for any of the cases 5.2–5.16 as discussed before. Since the result is the same no matter whether a particular letter is cancelled to the beginning or to the end, we are happy with the outcome in this case (cf. 4.9 and 7.16(ii)/7.17(2.ii)).□

5.17 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a truncation to the end where the corresponding letter l is also cancellable.*

Proof: In that case we are aiming at pointing out that it is not worse than the standard case 5.19 below: After the cancellation is performed, w and w' can still be regarded as differing by a single letter which is truncatable to the end. Namely the cancelling partner l^{-1} of l still appears in w' but does not in w, we definitely know l^{-1} to be truncatable to the end of w', since we know that whatever appears between the former l and the end of w is commuting with l (since l was assumed to be truncatable), and since we

assume that l and l^{-1} are a cancelling pair, that same holds for the subword bounded by l^{-1} and the former l. Hence within w' the letter l^{-1} comes out to be truncatable to the end, and this reduces the discussion of that case to the standard case 5.19 which will be discussed below. □

5.18 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a truncation to the end where the corresponding letter l is also movable by shortening transpositions.*

Proof: Of course our algorithm performs these processes and moves our letter to the left. However, none of them can affect the truncatability of l, since all letters which due to such moves might additionally show up between l and the end of w must be of commuting type with respect to l. Hence we can, as in the case of 5.17, regard our case as being reduced to the standard case 5.19 below: w and w' differ, after our algorithm has passed the crucial place still by just one truncatable letter, even if this letter has been moved around. □

5.19 Remark: *On how to treat the standard case of the truncation to-the-end cases:*
In this case we assume that our algorithm 4.1 is set upon two words w and w' such that w with respect to w' contains one additional letter l which is truncatable to the end, and such our algorithm can apply none of the above shortening processes that have been discussed in 5.16–5.18 when it appoints the truncatable letter to be the one of consideration. However, what makes this case in some sense more complicated than any of the cases 5.2–5.16 is the fact, that our algorithm in its w-run does not perform the truncation of l to the end when l is the letter of consideration, and hence it passes over the crucial place at which w and w' differ without having made our two words identical. This is, because our algorithm has by 4.1(v) been set up in such a way that it does not ask for truncations to the end at this state already; the corresponding questions are postponed until the last letter has become the one of consideration, and possible shortening operations to that letter have been performed. Hence we cannot argue as in all the preceding cases ("if the corresponding place containing the letters to be deleted or to be transposed is passed, then the two runs of our algorithm are in the same track and hence they have to end with the same word w'' also"), but we have to ask in what way the w-run might differ from the w'-run due to the fact that w contains this additional letter l, and whether changes that are imposed to w after l has been passed might affect the truncatability of l. Again we have to discuss subcases separately according to which operations might occur in the corresponding phase of the runs of our algorithm, and this happens below in 5.20–5.22:

5.20 Proposition: *The Algorithm 4.1 if put on two two words w and w' that just differ by some letter l that is truncatable to the end cannot end up with a different result due to truncations to the beginning that are performed to the right of the position of l.*

Proof: Since the letters between l and the end of w must all be commute with l, the presence of l can not mess up if some of these letters should be truncatable to be

beginning of w. Hence, if that should be given, the w-run and the w'-run of our algorithm perform the same acts of deletions (i.e. our two words still just differ by the presence of l) and in addition the disappearance of such letters cannot destroy the truncatability of l. □

5.21 Proposition: *The Algorithm 4.1, if put on two words w and w' that just differ by some letter l that is truncatable to the end, cannot end up with a different result due to cancellations that are performed to the right of the position of l.*

Proof: Similar arguments as in 5.20 can be used for cancellations: All cancellations amongst letters to left of l and between letters to the left of l and to the right of l come out the same for w and for w', thanks to commutativity of these letters with l. □

5.22 Proposition: *The Algorithm 4.1, if put on two words w and w' that just differ by some letter l that is truncatable to the end, cannot end up with a different result due to transpositions that are performed to the right of the position of l.*

Proof: There are now two different types of transpositions possible which we will be discussing in different items (i) and (ii). First observe that the letters between l and the end of our word cannot commute with each other, since they are all of the same basic direction. However, inside w some of them could be interchanged with l and in w' some of them might be interchanged with what comes left of the l-position. We are looking at both possibilities separately:

(i) If transpositions between l and letters on the right hand side of it should take place, they do not cause harm. They just move l closer to the end and hence they do not affect its truncatability. In addition, we note that even if such moves should take place (and of course only for w), the difference between the two words of consideration is still given by the fact that w has one truncatable letter more than w'.

(ii) This paragraph is now devoted to argue why our algorithm cannot find in its w'-run transpositions that it performs between letters on the left hand side of the l-position and on the right hand side of the l-position, while the presence of l within w prevents our algorithm from performing similar transpositions for w at the same time. Hence let us assume that such transpositions are performed in the course of the w'-run. In this case the letter to the left of the l-position must be of the other basic direction as the letters to the right of the l-position (we will denote the first of them by l_2), and hence it must belong to the same basic direction as l. Now according to 5.6(ii) there are just two possibilities for what the corresponding letter could be: l^{-1}, or some letter l'. The case "l^{-1}" need not to be discussed here (this has been done in 5.17 already), and hence we are left with the case that it is l', i.e. a letter that has the same basic direction and the same exponent as l. Hence, if it should really be a shortening and applicable process to interchange l_2 with l', it is also some operation of shortening type to interchange l with l_2 in the course of the w-run of our algorithm. The crucial question is whether l commutes with l_2 so that this operation is regarded as applicable then. We will below answer this question by "Yes", but we will discuss before why this answer will imply that even in this case the w- and w'-runs of our algorithm come out the same (apart

from the additional appearance of l) : We are assuming in this passage that the w'-run discovers and performs an interchange of l_2 with l' when l_2 is the letter of consideration. If in the w-run of our algorithm now l_2 is the letter of consideration, the answer "Yes" to the above questions implies that l is to be interchanged with l_2. Such operations have been discussed in (i). However, after this interchange is performed, our algorithm does not proceed by asking whether l could be moved further to the right; instead according to 4.1(iv) it enters a subloop in which it checks how far l_2 could be moved to the left. And due to that effect we get that within the w-run l_2 (and any other letter which might be in the l_2-position after the corresponding move of l_2) is moved precisely as far to the left as it is moved in the course of the w'-run — provided we will be able to see that

(1) l and l_2 do commute.

In order to give the corresponding argument, we have to use the facts that w and w' are both words that satisfy the compatibility demand C.1.6, and by C.1.6 this may be interpreted by a geometrical picture:
By knowing that l' and l_2 are adjacent letters of different basic direction in a compatible word, we get that there has to exist a lattice square where l' and l_2 give two adjacent sides of that square. Now use the fact that we also know "$l'll_2$" to be part of a compatible word. Since l' and l are of the same basic direction and have the same exponent, l must give the boundary of the lattice square just considered that lies opposed to l'. This geometrical set-up can be rephrased as follows: If our path, after having crossed l', turns to a certain direction (let us say: turns right) it sees the corridor l_2. If it, instead of turning right, continues straight on, crosses l, and then turns right, it still sees l_2. This implies that where the lines of corridors containing l and l_2 meet, there cannot be a hole. For, in that case our paths would see different corridors, according to whether it turns right before or after their crossing of l. And since the corresponding corridors l and l_2 do not meet in a hole, they meet in crossing region of corridors and this gives by B.1.8 that the associated variables l and l_2 are defined to be commuting. By hence having completed this postponed proof for (1), we also finished the discussion of (ii), and hence of Proposition 5.22. □

Having successfully proved Propositions 5.20–5.22, the first half of the proof-programme 5.19 is completed. The present result can be summed up that way, that the two runs of our algorithm in the cases of 5.16–5.19, if they have to pass over the truncatable letters, continue in an analogous way, and enter the final state 4.1(v) while operating on words that just differ by some truncatable letter. □

5.23 Remark: *On what remains to be proven in the truncation to-the-end cases 5.16–5.22:* Unfortunately 5.16–5.22 do not yet cover all aspects that need to be looked at when our algorithm is applied to two words that just differ by a truncation to the end. Recall that in 5.19 we already stressed that these cases are more difficult than the cases 5.2–5.15 before, since in these cases after passing over the position with the truncatable letter l the w- and the w'-run do not operate on the same word, as in all the other cases. For, according to 4.1 the truncation of l is not to be performed when l is to appointed to

be the letter of consideration, actually it is not to be deleted before the final Step 4.1(v) of our algorithm. What has been shown by the above Propositions 5.16-5.22 is just the following: After passing the l-position and till reaching the end-phase (v) of 4.1 the w- and the w'-run, even if they are not operating on identical words, do essentially perform the same changes to our words.
This leaves us with now having only to discuss what happens when our algorithm enters and is in the Phase 4.1(v), and if our two words just differ by one letter that is truncatable to the end (according to 5.21 and 5.22 this might not be the same letter as when the two words where given or it might not be at the same position any more — yet we will continue to denote it by l). What still needs to be seen is that in that final phase either the w'-run does not truncate any letter and the w-run truncates l only, or that the w'-run truncates one letter while the w-run deletes the same letter and in addition deletes l. This will be discussed in Propositions 5.24/5.25 separately for the cases where or where not l is found at the final position of w. In this context note that by 4.1(v) our algorithm is only supposed to ask twice whether or not a letter can be truncated to the end. Hence, unlike as in all preceding cases (5.2–5.15), even if now the w-run should detect and delete l, we cannot argue that then the two runs of our algorithm would automatically have to come up with the same result since they are then operating on the same words. They may be operating on the same word, but even if the words agree, the states of the algorithm do not.

5.24 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a letter l that is truncatable to the end and that is placed at the last position of w.*

Proof: By 5.23 we may, without loss of generality, assume that the w- and w'-runs both enter Phase 4.1(v) while still operating on the same words that just differ by some letter. According to 5.2 we use w, w' and l to denote those words and letters that the two runs of our algorithm see when entering Phase 4.1(v).
Since l can be found at the end of w, the first question of Phase 4.1(v) of the w-run detects and deletes l. Now the w-run and the w'-run see identical words, but as explained in the above remark they are unfortunately not in the same state, since the w-run has to ask one more question while the w'-run still asks two questions. So we have to continue in tracing the runs of our algorithm: When the w'-run asks whether the last letter is truncatable, it cannot detect a letter that is from the same basic direction as l. For, if such a letter should be detected, since it has to be on the one hand from an at most one-item list (cf. 3.9), and since on the other hand we know that appending l to our word gives the valid word that occurs in the w-run of our algorithm, the corresponding letter could only be some letter l^{-1}. But finding such a letter in a truncatable position in the w'-run gives that the same letter is in a cancellable position against l in the w-run. Hence our algorithms would have detected and performed the corresponding cancellation in the course of the w-run before entering Phase 4.1(v), and thus this situation cannot be given. Hence the first truncatable letter that is found in the course of the w'-run in Phase 4.1(v) is of the other basic direction than l is. Hence the same letter (or none, if there should be none) are detected and deleted, when the w-run asks its second question

of Phase 4.1(v). The w-run terminates after this second question, the w'-run might still have one more question to ask. Whether this question will or will not have to be asked depends on whether the corresponding letter has been at the rightmost position of w' at the beginning of Phase 4.1(v). If yes, it took even the w'-run two questions to detect this letter l', and it terminates with the same word as the w-run, and otherwise it still will ask one more question. This question will then be for a truncatable letter of the same basic direction as l had. With the same arguments as before such a letter could only be l^{-1}, and in most of the cases we get the same contradiction as before, since this letter would give a cancelling pair with l in the w-run. There is also an exceptional case given by the assumption that l and l' do not commute. In that case we have to face the possibility that the non-commuting of l and l' makes it impossible to cancel l against l^{-1} in the w-run, but it does not obstruct an iterated truncation of l' and l^{-1}. Note that such non-commuting truncatable letters can occur, even if this phenomenon is restricted to the case of paths which have their start- or endpoint at the corner of a hole (cf. 2.3). However, if we have such non-commuting truncatable letters, then a sequence such as

(1) $\qquad l^{-1}$ — letters of the other BD, amongst them l' — l

cannot occur, since this would imply that our path performs more than half of the act of encircling the hole at whose boundary it ends. In addition, the middle sequence of (1) cannot contain letters with different exponents by the same arguments as in 5.6(ii). I.e. during that sequence w passes along the same hole (without return), but at the beginning and at the end of that phase it is yet assumed to be in the same corridor. This geometrical picture gives that the behaviour as in (1) implies a contradiction to the compatibility demand C.1.6 for any word. Since this exceptional situation cannot be given, we finally see that the second question of the w'-run of our algorithm cannot cause an additional truncation, and hence the w-run and the w'-run terminate with leaving the same word w'' in this case. □

5.25 Proposition: *The Algorithm 4.1 reduces two words w and w' to the same short word w'', if w and w' just differ by a letter l that is truncatable to the end and that is not placed at the last position of w.*

Proof: By B.2.2(v), now the rightmost letter must belong to the other basic direction than l. Hence at the beginning of Phase 4.1(v) both runs of our algorithm ask whether the same letter is truncatable and give the same answer. Then the w-run of the algorithm asks the crucial question that gives the deletion of l. With the same argument as used in 5.24 the same question when asked by the w'-run, cannot return a truncatable letter since any such letter would give a cancelling pair in the course of the w-run. □

This was the last case to be discussed for a proof of Lemma 5.1. Since in all cases we found that the w-run and the w'-run of our algorithm 4.1 terminate when leaving us with the same word, 5.1 is now proved. □

§6. The obstruction to uniqueness of reduced word sequences

6.1 Overview of the results of this Section:
The following lemma 6.2 should be viewed in connection with the upward-extension results 1.9 and 3.13. Observe that those results do not imply that a sequence of transformations on the ith level, that bring one from some word ω_i to some word η_i and that consist of firstly applying only shortening and then only enlarging operations (4.8 and Sect.§5 were the last steps of the proof that such a sequence of transformations can be chosen if there should exist any, cf. 3.8), can be compatibly extended to the lower indexed words of ω and of η with an analogous mixture of shortening and enlarging operations. Let $\widehat{\omega_i}$ be the minimal word that is passed in the course of transforming ω_i into η_i. By 3.13 it only follows that the transformations that bring ω_i into $\widehat{\omega_i}$ can be compatibly extended to some shortening transformations on the upper index levels, and the same for those transformations, that turn η_i into $\widehat{\omega_i}$. These sequences of transformations end with some initial substring of a word sequence $(\widehat{\omega_1}, \widehat{\omega_2}, \dots, \widehat{\omega_i})$ and $(\widehat{\eta_1}, \widehat{\eta_2}, \dots, \widehat{\eta_i})$ where by construction we have $\widehat{\eta_i} = \widehat{\omega_i}$, but in view of Remark 6.7 and Example 6.8 we cannot expect equality for the other indices. From the arguments of 1.9 it follows that the

(1) corresponding words can only differ by transpositions that can be inserted in such a way into a transformation pattern that they can be synchronized with the identity on the ith level.

By using similar arguments as in the proof of 2.9(iii)–(iv) it follows that hence none of these sequences of transformations can contain a pair-interchange, since the compatibility properties imply that if four letters $A\,B\,C\,D$ are to be transposed into $C\,D\,A\,B$, then the corresponding split-up letters $a_j, b_j, c_j, d_j \quad (j \in \{1,2\})$ do only occur in a compatible order (cf. C.1.7(i)) to "$A\,B\,C\,D$" or to "$C\,D\,A\,B$" if b_1 comes before c_2 or if d_1 comes before a_2, respectively. Since the order of letters of the same basic direction may not be reversed, and only the first and last split-up letters of each basic direction might not be contained on the next finer level, we hence get that a pair-interchange "$A\,B\,C\,D \mapsto C\,D\,A\,B$ induces at least one pair interchange on a next level also, namely of $a_2\,b_1\,c_2\,d_1$ into $c_2\,d_1\,a_2\,b_1$. Hence pair-interchanges can never be synchronized to the identity, that we see in the transposition phase in the bottommost line. However, even without pair-interchanges there might still occur a lot of transpositions, and we would be very much interested in bringing these transpositions into such a order that all shortening ones are performed before the enlarging ones. Unfortunately we will have to admit that in general this cannot be done, and what has been described in (1) will remain the "best" result that actually can be used in the remainder of this paper. The following Lemma 6.2 gives us "almost" the statement that would be needed to obtain a better result, i.e. such a result that with the techniques of 7.9–7.13 would be needed to obtain unique reduced representatives for our weak equivalence classes. Even though Lemma 6.2 (more precisely: the part of Lemma 6.2, that has not been already proven by this Remark 6.1) will not have to be quoted in the remainder of this paper, we will prove it in its full generality, since that way we are precisely lead to the obstruction that

keeps us from having unique reduced representatives in our weak equivalence classes (cf. 6.8).

6.2 Lemma: *(third upward extension lemma, cf. 1.9 and 3.13)*
Let $(\omega_1, \ldots, \omega_i)$ and $(\eta_1, \ldots, \eta_i)$ two tuples of words as they might occur as initial substrings of word sequences (i.e. the relevant conditions of C.1.5 are satisfied up to the index level i). In addition assume that $\omega_i = \eta_i$ and is a reduced word. Then there exists a transformation matrix $\widehat{\Omega}$ as in Formula 1.9(1)) such that the ω_i give the leftmost column, the η_i give that rightmost column, and that the bottom row consists of identical entries ω_i. In addition, the transformations that occur in the other rows can be grouped in such a way that in the left part of each row only shortening transformations can be found, in the right part only enlarging ones, and in the part in between only transpositions that do not contain pair-interchanges (cf. 2.9(iv)). In addition, in the row above the bottommost one all shortening transpositions can are situated to the left of all enlarging ones.

Proof: All claims of this lemma apart that concerning the order of transpositions follow from applying 1.9 to the empty sequence of transformations (cf. the corresponding explanations of Remark 6.1).
Hence we just need to consider the middle part of the row above the bottommost one, and we can assume that with ω'_{i-1} and η'_{i-1} words are given that are in principle readings from some path that realizes ω_i. With the same arguments as in the corresponding state of the proof of 1.9 (in particular 1.13) we see that ω'_{i-1} can be transformed into η'_{i-1} by a sequence of transpositions, and that they can be synchronized with the identity on the ith level. We draw the transformation diagram of these processes (cf. 1.3), avoiding unnecessary crossings. Since we do not have to deal with cancellations and truncations, there will not be any cups, caps or any lines disappearing at the sides from our diagram; we will only see lines that connect points at the top-bar with points at the bottom bar. In the following propositions 6.3–6.6 we will separately discuss the various types of patterns of lines that could be found in such a picture, aiming at pointing out firstly how some special cases can be treated and later why the more difficult ones can be treated by analogous methods. That way the proof of 6.2 will be complete after Proposition 6.6:

6.3 Proposition: *A sequence of transpositions that can be synchronized to the identity on the next finer level can be grouped in such a way that all shortening transpositions are performed before all enlarging ones if in the transformation diagram each line intersects only one other line* ("$\diagdown\!\!\!\diagup$-only case").

Proof: By definition of a transformation diagram (cf. 1.3) each of the intersection points in the diagram corresponds to a transposition. By our assumption on the intersection pattern in this proposition, the various transpositions do only involve disjoint sets of letters. Hence they can be performed an in any order, in particular we can choose some order that firstly only performs the shortening ones and postpones the others. By having

seen that such an order exist, we are done in this special case. □

6.4 Proposition: *A sequence of transpositions that can be synchronized to the identity on the next finer level cannot contain a pair of parallel strings that crosses some other pair in the transformation diagram* ("$\exists$⫻-case").

Proof: Let A, B, C, D the corresponding letters at the top bar and accordingly we read at the corresponding segment of the bottom bar "$CDAB$". Use $a_1, a_2, b_1, b_2, c_1, c_2, d_1$ and d_2 to denote the corresponding split-up letters. We search for an order of those split-up letters that is compatible the same with "$ABCD$" and with "$CDAB$". Since our the corresponding transpositions of our pair of letters could only have matched the rule B.2.2(ii) if $BD(A) = BD(B) \neq BD(C) = BD(D)$, (recall that "$BD$" is an operator by B.1.2(i)) we also see, that a_1, a_2, b_1 and b_2 must be of the same basic direction, and c_1, c_2, d_1, d_2 of the other. Hence the order of any of those 4-tuples cannot be mixed up on the bottom bar, the question will just be in which order we find the next letters coming which of the pairs. Thus we can regard the letters "a_1, a_2, b_1, b_2" as fixed "coordinates" that tell us how far to the left the letters of the other 4-tuples are placed. And in that language we see that, if the order of our small letters has to be compatible with "$ABCD$", then the leftmost place for c_2 is between b_1 and b_2. If, on the other hand, the order of our small letters shall be compatible with "$CDAB$", then the rightmost place for d_1 is between a_1 and b_2. This accumulates our conditions on the order of letters to a pile that cannot all be matched any more. Consequently, the assumption of this proposition cannot be given. (Note that we did not essentially refer to the split-up letter a_1, c_1, b_2 or d_2 in the estimates, hence the above arguments also hold in the semi-split case, when not all of the corresponding split-up letters may be present). □

6.5 Proposition: *A sequence of transpositions that can be synchronized to the identity of a reduced word on the next finer level can be grouped in such a way that all shortening transpositions are performed before all enlarging ones if all path-components of the transformation diagram consist of one line that is intersected by possibly more parallel lines* ("⫻-only case").

Proof: In principle, we want to apply the same argument as in 6.3: Each path-component of our graphs represents transformations that can be performed independently of the transformations associated with the other path-components, and, by finding that the transformations belonging to one path-component all are of shortening or all are of enlarging type, we will reduce this case to the same key arguments as in 6.3. However, the claim of the preceding sentence requires some combinatorial analysis:
If we see a "⫻"-figure, we see one line of type "\"; denote the letter that belongs to that line by "L". All other lines are of type "/", and we denote their associated letters by $B_1, B_2, B_3, \ldots, B_k$. Since we assume that our transpositions are legitimate, we have to have $BD(B_1) = BD(B_2) = \ldots = BD(B_k) \neq BD(L)$. Denote the split-up letters of L by l_1 and l_2, and, in order to avoid double-indices, denote the first split-up letter of each B_j by b_j, and the corresponding second split-up letter by c_j (it will turn out that

semi-split cases can be treated by essentially the same arguments, just by an according adaptation of the language). Similarly, as in the proof of 6.4 we can use the b_j and c_j as coordinates for the positions of the l_ν. For having some order of letters that is compatible the same with "$LB_1B_2B_3\dots B_m$" as with "$B_1B_2B_3\dots B_kL$", we get that the rightmost position for l_1 that is permitted is between b_1 and c_1 and analogously the leftmost position of l_2 is between b_m and c_m.
The crucial question of this paragraph is whether the B_i do all have the same exponent. For, if they have, we are done, since this would imply that all transpositions of consideration are of the same (shortening or enlarging) type, and, as mentioned above, this is the desired result of this proposition. Hence let us now assume the opposite, and let us discuss this assumption till some contradiction arises:
Hence assume that B_j and B_{j+1} have different exponent. Since according to the assumptions of our transformation diagram there cannot be another letter in between, by 5.6(ii) we have the $B_{j+1} = B_j^{-1}$, i.e. that ω'_{i-1} and η'_{i-1} are not reduced. On the other hand by using the notation of Lemma 6.2 $\omega_i = \eta_i$ was assumed to be reduced. Hence there must in particular be some obstruction why c_j cannot be cancelled against b_{j+1}, even if these letters must have different exponent and are from the same line of corridors. Such a geometrical situation has been discussed in 5.6(ii), with the result that any obstruction for cancelling c_j against b_{j+1} implies the existence of some letter l' of the other basic direction between b_j and c_{j+1}. However, then the letters of the corresponding basic direction would read (at least) "$l_1l'l_2$", and this gives a contradiction: For, l_1 and l_2 have been assumed to be split-up partners, and hence they are letters taken from adjacent parallel lines of corridor. Whatever letter of the corresponding basic direction crops up between l_1 and l_2, it has to belong to one of the corresponding lines of split-up corridors and hence l_1 and l_2 could then not be split-up partners any more. As explained above, this contradiction implies that we are able to handle the case treated in this proposition appropriately, also. □

6.6 Proposition: *A sequence of transpositions that can be synchronized to the identity of a reduced word on the next finer level can be grouped in such a way that all shortening transpositions are performed before all enlarging ones* ("the general case").

Proof: Note that by 6.4 we cannot see "diamonds" (i.e. ⨳-figures) in our transformation graph. Hence, the general picture is actually not that far away from 6.5: It arises by taking several types of path components as seen in 6.5 and sliding them a bit apart so that there is at most one intersection between two neighbouring of them. This makes our path-components bigger and their structure a bit more complicated as in 6.5. However, we can still apply the same essential argument as already in 6.3 and in 6.5: If all transpositions that are associated with the one path-component are of the same (shortening or enlarging) type, we are done. And that the latter condition is given, has now to be proved by an induction-argument: As we march along inside our path component, it has to be shown that each crossing corresponds to the same type of transposition as the last crossing. And since that induction argument is precisely given by our computations of 6.5, we are done in the general case, also. Hence this proposition

and with it Lemma 6.2 is proved. □

6.7 Remark *on why the result 6.2 can essentially not be improved:*
Looking at the preceding proofs in 6.2–6.6 we may wonder whether it was actually necessary to restrict the claim about a suitable order of the middle sequence of transpositions to the line above the bottommost one. More precisely we ask: "Is it definitely hopeless to extend the arguments as given in the preceding proofs 6.3–6.6 in such a way that they can iteratively be applied to bring an appropriate order into all of the corresponding lines of $\widehat{\Omega}$ for 6.2?" – It will not be hard to see that if it was possible, then in 7.9–7.13 all the going via tunnels would not be necessary, and instead the proof-techniques of 7.9–7.13 would straightforwardly give that our weak equivalence classes then contained unique reduced representatives. However, this cannot be true as the following example 6.8 will show. Hence what might have appeared in the preceding proof as a technical obstruction for just applying the corresponding arguments inductively, should indeed be considered as fairly close to the borderline between right and wrong: From a technical point of view we could not have inductively used the corresponding arguments, since in 6.5 we explicitly had to refer to the fact that the corresponding entries in the line below are reduced. This was by assumption given for for the bottommost line and hence our arguments worked for the line above. However, we cannot not assume that an arbitrary word from a word sequence is reduced, even if the associated path is in an "algebraically reduced" (cf. A.4.10) position; the corresponding geometrical pictures had been described in 1.1(i) and 1.1(ii) already. And hence we could not have used the same arguments as before to get a similar order into the transposition as for the line above the bottommost one. Observe that the *following example for a word sequence which cannot have a unique reduced minimal representative* uses precisely the same effect:

6.8 Example: Let A and B be a vertical and a horizontal corridor of $\mathcal{N}_i$, respectively, which intersect each other in a crossing region of corridors. Assume that their split-up letters are denoted by a_1, a_2, b_1 and b_2 respectively. Now assume that some segment of a word sequence on the $(i+1)^{\text{st}}$ level reads "$a_1 b_1 b_2 b_2^{-1} b_1^{-1} a_2$". Note that this word might definitely belong to a short path (for a picture cf. E.4.19(Fig.p_1)), and all we have to assume is that on the $(i+2)^{\text{nd}}$ level there is some hole cropping up in the upper part of the A-corridor so that our path runs around that hole. Such a hole then would make the cancellation of b_2 against b_2^{-1} non-extendible as explained in 1.1(i). Apart from that the word looks fairly short already, observe that in particular a transposition between a_1 and its succeeding $b_?$-letters and between a_2 and its preceding $b_?^{-1}$-letters would be of enlarging type according to B.2.3(iii). Hence we need not to assume that there are other holes around apart from the one mentioned before. Now ask what could be the ith corresponding word. According to the principles of reading the ith word from the $(i+1)^{\text{st}}$ one it has the contain the letters A, B and B^{-1}, and the order is fairly free since our path goes through the corresponding crossing regions. However the "symmetric" arrangement "BAB^{-1}" is not short, since an interchange of A and B is of shortening type. Since the $(i+1)^{\text{st}}$ word need not be adapted when this interchange is performed, it

is also clearly an extendible process. Unfortunately an according interchange of A with B^{-1}, when applied to the symmetric arrangement, is also of shortening type. Hence we are stuck on the ith level with two "short" orders, namely "ABB^{-1}" and "$BB^{-1}A$", for which there apparently is no way to reduce them any further, a cancellation of B against B^{-1} cannot be performed since it is non-extendible!

§7. The uniqueness theorem for firm equivalence classes of reduced word sequences

7.9/7.14/7.17 are the uniqueness theorems and have been vaguely described as "Theorem F" at the beginning of this paper, already. There are still some preparatory steps needed in order to prove them, the most important of which is the introduction of a nested graph of word sequences. Observe that the result of the uniqueness theorem is for word sequences only; it cannot be extended to triple sequences as we will explain in 7.16!

7.1 Lemma: *If ω' is a word sequence, each of whose words can be derived from the corresponding word of ω by firstly applying a sequence of shortening processes and then transpositions only, then ω' is weakly equivalent to ω.*

Proof: The non-trivial statement of this lemma is that we did not assume our sequence of finite sequences of transformation processes to be compatible (cf. 1.2/B.2.8), and hence it is the non-trivial task of this proof to construct out of the given information and the internal compatibility of the word sequences ω and ω' such a compatible sequence of finite sequences of transformation processes that can establish the equivalence between ω and ω' in the sense of Definition B.2.9(i):
Compatibility can be obtained by a conclusion essentially using the Königslemma (cf. C.5.10). In order to apply this graph-theoretical scheme of conclusions for our purposes, we construct the following tree: The vertices of our tree are given by finite sequences of words $(w_1, w_2, \ldots, w_k)$ that for some $i \in \mathbb{N}$ satisfy that w_1 is the ith word of ω, that w_k is the ith word of ω' and that w_j results from w_{j-1} by either applying one elementary shortening operation as defined in B.2.2(i)–B.2.2(v)/B.2.3 to w_{j-1} $(j \in \{2, 3, \ldots, k\})$, or, for bigger j, one transposition. By 1.9 the compatibility properties of a word sequence imply that such a finite sequence of words giving a "reduction scheme" induces a reduction scheme for the $(i-1)^{\text{th}}$ words of our word sequences ω and ω', also; in particular 6.2 gives that the property of applying shortening transformations only and in the final phase (not necessarily shortening) transpositions can be realized on the $(i-1)^{\text{st}}$ level, also.
We then define that two vertices which are given by induced reduction schemes as in the above sentence are to be connected by an edge. Note that by definition any vertex is associated with a level index, and that all edges of our tree are are joining vertices with adjacent level indices. Since the reading of the $(i-1)^{\text{st}}$ word not always is unique (cf. C.1.7(i)), there may be more than one edge emanating to the level above. Due to this

phenomenon we do not necessarily obtain a tree, but our graph fits into the framework as described in C.5.10. Hence the existence of an infinite branch of that graph follows from the same arguments as the classical Königslemma (cf. C.5.10). Such an infinite branch then gives the desired compatible reduction scheme that establishes equivalence between ω and ω'.
Note that the graph as described in this proof was constructed according to the same principle as our transformation graph in 1.18. However, in this proof we would have not been done by simply quoting 1.18, since the 1.18-graphs had been constructed out of the data of a compatible transformation pattern, already, whereas in this proof the compatibility-demand was not a part of the assumption, but had what to be constructed. □

In particular note the following special case of the above Lemma 7.1, where ω' is the empty word sequence:
7.2 Corollary: *Let ω be a word sequence each of which words can be transformed to the empty word by elementary operations that only are of shortening type. Then ω is weakly equivalent to the empty word sequence.*

Proof: It is straightforward how Lemma 7.1 has to be applied in order to get the statement of this corollary and most of the claimed properties come out directly, also. However, there is one point which perhaps deserves some attention: Recall that Lemma 7.1 does not leave us with shortening transformations only, but after the shortening transformations have been performed, a sequence of transpositions some of which might also be of enlarging type had to follow. Our Corollary 7.2 does not claim the existence of such a phase of transpositions. Yet it follows from 7.1, since 7.1 implies that this phase of transpositions can not be followed by a phase of any other operations. Now, our target-word sequence ω' is empty. Since transpositions cannot delete letters and they are not followed by other operations that could do this job, the phase of transpositions must be empty in the 7.2-case. □

7.3 Remark: *Although graph-theoretical arguments* as used in 7.1 will be our most important tool when it comes to prove the existence of compatible cancellation or transformation patterns, the concept of graphs as used in the proof of 7.1 is not sufficient for the desired proof of our uniqueness theorem 7.9/7.14. For that purpose the subsequent paragraphs 7.4–7.6 will be devoted to define a more general concept. However, the remainder of this paragraph is devoted to explain why *we are not done with the concepts as developed so far:*
Observe that when discussing equivalence transformations between word sequences, we have to deal with two aspects both of which might be suitable for being described by a graph-structure: On the one hand the possible skipping between words by applying elementary transformations to them might be suitable to be reflected by a graph structure (as done in 1.18/7.1), which identifies words with vertices, draws edges, where two words just differ by an elementary transformation, and reflects whether the corresponding transformation is of shortening or of enlarging type by putting an orientation on the corresponding edge. On the other hand it might be in a similar way helpful to

have another aspect, namely the skipping between the various words of a word sequence and in particular the question which could be the possible $(i+1)^{\text{st}}$ word of a word sequence for a fixed $\overline{\mathcal{A}_i}$-word being described by a graph structure. Note that none of the graphs as constructed so far in this paper can be used for that purpose: The graphs as constructed in C.3.4/C.5.5/1.3 are treating letters as edges, not words (and, apart of that, the first two of them were set up in such a way that they just did reflect the insertion structure of a single word, not of two words to be transformed into each other). The graph-concept of 1.18 actually describes both of our relevant aspects, but it reflects the second aspect just by the line-structure of the graph, not by its edges. Hence this concept is only suitable for describing one way of how to transform two weakly equivalent word sequences into each other, never to give an overall picture of the corresponding equivalence class. Indeed, the graph as constructed in 7.1 was set up to reflect both of our relevant aspects. However, this concept of a graph was based on the construction principle that just the second aspect was described by the graph structure, while the first aspect, i.e. the different possibilities to cancel a word down to the empty word sequence had just been displayed by associating different possibilities with different vertices. Since in other contexts within our proof-programme we cannot do with such a structure, in the following we will develop another graph-theoretical concept which definitely describes both aspects of equivalence relations of word sequences by graph-theoretical means. Since this cannot be done by an ordinary graph, we define a more sophisticated concept below in 7.5:

7.4 Remark on how many nested graphs of word sequences will be constructed for a given alphabet:
Note that in 7.5 we will define nested graphs for providing an overall picture of the equivalence classes of all word sequences that can be constructed on a fixed alphabet $\overline{\mathcal{A}}$. Roughly speaking, the path-components of such a graph will correspond to the weak equivalence classes of word sequences. Hence in the remainder of this paper we will usually restrict our considerations to a fixed path-component of such a nested graph, but when constructing the corresponding graph we have to consider all word sequences that can according to C.1.6–C.1.7(iii) be constructed on the alphabet $\overline{\mathcal{A}}$ such that the associated path has a certain start point and a certain endpoint. *I.e., given a fixed alphabet, we will construct one nested graph for each pair of given points in X.* When constructing our graph, we have to consider all words that satisfy the corresponding conditions for their start- and end-square as defined by C.1.6, i.e. where the corresponding first and third entries of the entire triple are lattice squares that contain our start- and endpoint (whether only one or more than one of these squares are permitted depends on whether the prescribed start- or endpoint does or does not lie in a corridor, or, if it does, whether we are working in an accordingly free or fixed equivalence relation, cf. 3.6/3.9). Since we want to restrict our graph constructions to paths with prescribed endpoints, this gives by 2.3 an at most four-items list of truncatable variables (recall that by 3.9 such a list is needed to define the corresponding equivalence relation for words). We will also build up nested graphs when working with our strong equivalence relation, confer 7.14 for how to select the set of vertices in that case.

7.5 Definition of the basic skeleton for a *nested graph of word sequences:*
We start our construction as follows: We first only consider the set of possible words on $\overline{\mathcal{A}_1}$, i.e. the set of all words may occur as the first word of one of our word sequences of consideration, if the start- and endpoints of the associated path are given (cf. 7.4). Each of those words gives rise to a vertex of our graph. If two of those words happen to be equivalent and if it takes only one elementary process B.2.2(i)–B.2.2(v) to transform these two words into each other, the corresponding vertices are to be joined by an edge. We define a partial order between the vertices of our graph, taking into account that each of our elementary operations is either of shortening or of enlarging type. The partial order is defined so that that the vertex a is considered to be bigger than the vertex b, if one can go from b to a by using edges that correspond to enlarging operations only. We then extend our partial order of vertices arbitrarily to a total order and, while respecting this total order we associate to each vertex the value of a height-function. Then our graph is drawn into the (x, y)-plane and the y-coordinate is identified with the height. This can be done in any case, if we take into account that we might get some intersections of edges which are not to be considered as vertices. Note that none of our edges can be entirely horizontal, since each of them corresponds to some operation, and each operation is either of shortening or of enlarging type.
We then thicken our graph up by drawing thick edges and thick vertices, i.e. each vertex is replaced by a round disk, and each edge by a corridor that connects the corresponding thick vertices. Unlike as for $\mathcal{N}_i$-corridors these corridors can never be horizontal and will in general not be vertical either. We then produce a similar graph for the second level index, and this time we wish to draw it inside the graph just produced, i.e. we want to draw each vertex of the second graph inside a thickened vertex of the first graph and each edge either inside a thickened vertex or (if it connects two vertices that have been drawn inside different vertices of the first graph) at least inside the region covered by two thickened vertices and one thickened edge that joins them. It is a demand of that definition 7.5 that a vertex may only be drawn into some other vertex if the words that are associated with our two vertices satisfy the compatibility demands C.1.6–C.1.7(iii). In addition, an edge may only pass through one other edge, if the transformation processes that are represented by those edges are compatible in the sense of B.2.7.
In this context we have to particularly pay attention to the phenomenon as seen in 1.13, that there exist changes that can be performed on some level and that do not impose changes on the finer levels. This phenomenon has to be translated into our graph picture in such a way, that different vertices on the first level may have identically labelled vertices of the second level inside. Such identically labelled vertices that sit inside adjacent vertices of the upper level, are also to be connected by an edge. And these edges corresponding to identical operations are also oriented, namely they inherit the orientation from the corresponding thick edge that contains them. Having found this way for how to treat operations that can identically be extended to finer lattices, it just takes a straightforward quoting of the combinatorial results of 1.9/3.13 to see that the second level of our graph can actually be drawn as required: By 1.9 each elementary

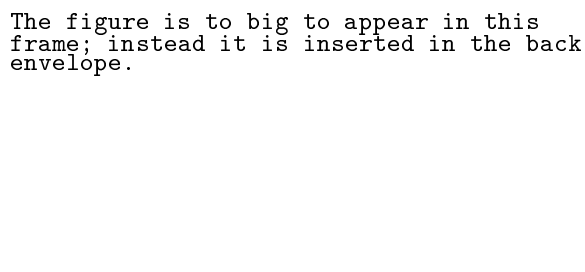


Fig.1 (belonging to Definition 7.5/7.6):
This picture gives an impression of how three adjacent levels in some part of a realization diagram might look like. Of course, only a small part of such a graph could be pictured: Firstly, only three adjacent levels are contained in the figure, secondly only a small selection of vertices of each of the levels could be drawn, and even the selection of vertices inside one of the bigger fat vertices could not be drawn completely. For the sake of keeping the picture simple we also resisted to indicate that lots of small corridors or tunnels might be emanating from the thick vertices that have been drawn. Note that lines which are neither circles nor do have a parallel line are actually not a part of the realization diagram; they are to be regarded as pointers that attach the labels to the corresponding thick vertices.

operation that is applied to a word either extends to the identity on the upper level (in this case we get those edges on the second level that stay inside the thick vertex of the first level), or it corresponds to a sequence of operations on the first level, so that at most one of them has to be synchronized with the given operation on the second level and the other just correspond to the identity on the second level. Hence in our graph picture all these operations can be reflected by edges that are drawn inside edges of the first level, as required. We then thicken our second graph also and draw our third graph corresponding to the variety of words of the third level inside the thickened graph of the second words, and continue that way for any of the words of our word sequences of consideration. A nested graph of word sequences results from iterating these steps infinitely many times, so that the final picture can in reality never be drawn, since it has arbitrarily fine features. In the final picture each vertex and each edge are thickened, and most of them will contain infinitely many other edges or vertices inside.

However, we are not happy with this picture as developed in 7.5 (yet). This is why the picture as described so far only gives the "basic skeleton" for a nested graph. Indeed, *a nested graph of word sequences will arise by using the basic skeleton as defined in 7.5 and by inserting new objects, namely "tunnels" according to the subsequent Definition 7.6:*

7.6 Definition of how to insert *tunnels* into nested graphs:
In principle, tunnels could be regarded as a special kind of edges, but they are usually not meant to be included if we will use the phrase "edge". In our nested picture they will also be drawn thick, just with the same diameter as the vertices to be connected. In our Fig.1 we will distinguish tunnels from edges by drawing the latter ones solid, the first ones dashed. However, the criterion which vertices are to be connected by tunnels

is a different one from the corresponding criterion for edges: Two vertices are to be connected by a tunnel, if and only if they represent words that consist of the same set of letters so that these words can be transformed into each other by using a sequence of valid transpositions only that does not contain a "pair-interchange". In that context a "pair-interchange" would be given if four letters that in the first word appear in the order *"$a \dots b \dots c \dots d$"* come out as *"$c \dots d \dots a \dots b$"* as a result of our transformations. Usually we think of tunnels as fairly horizontal in our picture. On the other hand there is no warranty that the slope-conditions of our edges always permit to draw all tunnels exactly horizontal; however when going through a tunnel, the slope of the picture will not matter. Some other differences between tunnels and edges have to pointed out here: By definition edges had only to be drawn if the corresponding vertices to be connected could either be found inside the same or inside adjacent vertices of the preceding level. Hence, wherever and edge occurs in our graph picture, it will be properly surrounded. In contrast to this, vertices have been connected by tunnels without caring whether the surrounding vertices also satisfy the condition for being connected by a tunnel. Hence, while in forthcoming proof for edges we often will have to ask, whether a certain transformation process will give rise to an edge in our graph or whether the insertion of the edge into our graph is obstructed by some misfortunate behaviour of the surrounding vertices, for tunnels we will often have to ask the converse question: "Is a tunnel actually properly surrounded by other tunnels and hence appears with a similar nesting situation as all edges by definition do?"

7.7 Properties: Having defined nested graphs in 7.5/7.6 it is not hard to observe how some of the combinatorial concepts as developed before can be represented in such a graph:

(i) *Any word sequence that fits into the framework according to which a nested graph has been constructed is represented by a "sequence of nested vertices".* The meaning of "framework" in this context depends on according to which equivalence relation the corresponding nested graph has been constructed. In case it was a free equivalence relation, our word sequence has to represent paths with a corresponding start- and endpoint; in case it was a fixed one, our word sequence has to belong to a triple sequence with the corresponding data in the first and third entry. Finally, a "sequence of nested vertices" is defined to be a sequence which on each level consists of one thick vertex such that each of the vertices is contained in each of its predecessors.

(ii) *A compatible transformation pattern for two word sequences that are represented in the nested graph is given by a "stabilizing sequence of edge paths".* Such a sequence of nested edge-paths consists of one edge-path on each level of the nested word that satisfy the following conditions:

- ▷ The start- and end-points of those edge-paths are nested sequences of vertices, so that in particular the ith path does also start and end in the same vertex of the $(i-1)^{\text{st}}$ as the $(i-1)^{\text{st}}$ one.
- ▷ According to our nesting picture of graphs, apart from the first one, each edge-path uses the same edges of the predecessor-level as its predecessor-paths.

(iii) Each compatible transformation pattern that consists of shortening transformations only, is represented by a nested system of edge paths that uses downleading edges only.

7.8 Remark: When recalling the above definition 7.4–7.6 one may wonder whether it would have not been a more suitable concept just to set up one nested graph for all word sequences that can be defined on a given alphabet $\overline{\mathcal{A}}$, and then to leave it to the path-connectedness properties and to the nesting structure of the corresponding edges to find those (kind of) path-components the represent the weak equivalence classes. This paragraph is devoted to explain, why such a concept cannot straightforwardly work:
Recall that in the above Definition 7.4–7.6 we did at one place refer to 3.9, and simultaneously point out that according to our assumptions we could refer to a list of truncatable variables that is finite on each level index. If we would attempt to construct the "overall nested graph of word sequences" (for a particular alphabet) we would have to regard all truncations of arbitrary variables as legitimate. Then in view of B.2.1(iii)/1.6 only those paths in such a graph could correspond to valid transformations of word sequences, which, when viewed with respect to the nested edge-structure, only involve a bounded number of truncation edges. On the other hand, the vital proof-arguments concerning nested graphs that are used below are stabilization properties, i.e. the observation that, while still finding better ways on the finer levels of such graphs, the corresponding edge-paths for smaller level indices remain stable. And, if in the course of such proofs we would have to make sure, that we are not taking up more and more truncation edges when finding connecting paths on finer levels, then our proof-arguments would be messed up severely. For that reason we resisted to look at the overall nested-graph picture of word sequences on a fixed alphabet.

Observe that the following result 7.9 had already been described as Theorem B.2.12 and as "Theorem F" in the Introduction

7.9 Theorem: *The four weak equivalence relations for word sequences as introduced in 3.6 satisfy that within each equivalence class there exists precisely one firm subclass containing all reduced representatives.*

Proof: Given two word sequences ω and ω' which are equivalent in the sense of the corresponding relevant relation. By 7.4–7.7(i) there exists an associated nested graph that represent both of these word sequences as systems of nested vertices. The proof of this theorem is lengthy and hence will be split up over 7.9–7.13. As a first step of this proof we use graph-theoretical methods in order to achieve a

7.10 Construction of a candidate of the reduced representative of two word sequences ω and ω' that are equivalent in the sense of a weak 3.6-equivalence relation:
We apply our shortening algorithm 4.1 to each of the words of ω. We then try to find, starting at the ω-vertex some monotonously downleading edge-path on the corresponding level of the graph that realizes the corresponding sequence of shortening transformations. On the top-level such an edge-path always exists by definition of our graph, since on that level each word (in question) is represented by one and only one vertex. For the higher levels there might arise problems when trying to realize some transformation sequence of a word as a path in our graph, since here the same word may

be represented by more than one vertex (different vertices with the same label sitting inside different vertices of lower indices), and apart from that, only those connecting edges have been drawn for which there had been some place inside the edges or vertices of the preceding level. However, here we can use the result 3.13 since this guarantees that for any system of shortening transformations, provided the target-sequence is not prescribed, there is always one possibility to upwards extend the transformations. That way we get that all nesting conditions of edges and vertices can be satisfied and some downleading edge-path can be drawn that realizes the given sequence of transformations. We call the corresponding edge-paths $\rho_{i,i}$. Here i stands for the corresponding level-index, and in particular the first index "i" is to tell us, that this path is realized on the ith level of our graph, and the second index tells us, that this path is induced by the given reduction pattern for the ith word of ω. We will also consider paths $\rho_{i,j}$ for $i < j$, and according to the convention as just explained, these paths arise by looking at the way that $\rho_{j,j}$ has taken in our nested graph and then giving the record which edges of the ith level our path $\rho_{j,j}$ has used by finding its way through our nested graphs (recall that all edges of the jth level had been drawn inside some edges of the ith level). Alternatively, one could rephrase the definition of $\rho_{i,j}$ by referring to the 3.13-extension of the reduction pattern for ω_j : The path $\rho_{i,j}$ is the path that is associated with the ith line obtained by extending upward the given reduction pattern for ω_i as constructed in 3.13. That way we construct a double-indexed system of edge-paths $(\rho_{i,j})_{i\geq j\in\mathbb{N}}$, and, roughly speaking, our candidate for a reduced representative of ω shall arise as a kind of limit of the endpoint of these paths, obtained by stabilization.

Hence we ask whether for any fixed i always $\lim_{j\to\infty}(\rho_{i,j})$ exists by stabilization. The most important property that in this context can be used is the finiteness of the relevant part of our nested graph: We can restrict our considerations to a fixed level and, apart from that, we need only consider the system of those vertices and their connecting edges that arise from ω_i or from ω_i' by shortening operations. The corresponding system of words is restricted to those words that can be constructed out of the same or a smaller set of letters as ω_i or ω_i', and there are at most finitely many combinations that could give such word. Even, if in addition we take into account that identically labelled vertices may sit inside different vertices of less fine levels and hence that for getting a honest picture of the ith level we have by reconstructing the possible lower indexed words to compute the number of duplicates, we still get a finite number of combinations, only. On the other hand, finiteness alone does not guarantee convergence, but it guarantees accumulation. And hence we are able to define our candidate for a reduced representative of the ω-vertex, namely as follows:

Step 1.: Consider all the paths $\rho_{1,j}$. Since there are infinitely many downleading paths in a finite graph, some path has to be infinitely many times used. Define ρ_1 to be this path.

Step 2.: Now consider that subsequence of paths $\rho_{2,j}$ that remains inside the thickened path ρ_1. By construction this still must be infinitely many (since ρ_1 had been induced by infinitely many paths); and since our graph of the second level (and hence in particular the subgraph inside ρ_1) is finite, there again will be some path which occurs

infinitely many times in the sequence. If there should be more than one, we decide arbitrarily to call one of them ρ_2. Since ρ_2 was obtained by corresponding stabilizing properties, ρ_2 must end inside the thick vertex that marks the endpoint of ρ_1.

Step 3.: By iterating this scheme of conclusion we manage to construct a whole system of paths $\rho := (\rho_i)_{i\in\mathbb{N}}$ that satisfies all conditions of a "stabilizing system of down-leading edge-paths" in the sense of 7.7(iii). The endpoint of this stabilizing system of paths then is a nested system of vertices that by 7.7(i) represents a word sequence. We will call this word sequence $\widehat{\omega}$, and this is our candidate for a reduced representative of ω. By 7.7(ii) the ρ-paths represent the corresponding reduction scheme that proves that ω and $\widehat{\omega}$ are actually weakly equivalent. Note that apart from stabilizing properties implicitly the "upward-extension results" 1.9/3.13 and Königslemma type of conclusions had been used to construct the corresponding transformation pattern.

The same construction scheme then is applied to construct $\widehat{\omega}'$, a candidate for a reduced representative of ω'.

Recall that, since (cf. 6.8) we had to give up hope that in a general context we can expect $\widehat{\omega}$ and $\widehat{\omega}'$ to be the same word sequence, all we can hope for is to prove that these two word sequences are at least firmly equivalent. This will be done in the next three steps 7.11–7.13 of the proof of 7.9:

7.11 Proposition: *The reduced representatives $\widehat{\omega}$ and $\widehat{\omega}'$ that have by the method as described in 7.10 been constructed out of two equivalent word sequences ω and ω are in the corresponding graph of nested word sequences connected by an infinite system of nested tunnels.*

Proof: By assumption, $\widehat{\omega}$ and $\widehat{\omega}'$ are weakly equivalent, and that way our constructions and assumptions together give that $\widehat{\omega} \sim \omega \sim \omega' \sim \widehat{\omega}'$, i.e., that $\widehat{\omega}$ and $\widehat{\omega}'$, are weakly equivalent, also. Hence the words of $\widehat{\omega}$ can be transformed into the words of $\widehat{\omega}'$, by firstly going over shortening transformations only till the reduced representative in the 3.9-sense is reached, and then by using enlarging transformations again in order to get to the $\widehat{\omega}'$-word. These enlarging transformations can be constructed by applying our algorithm 4.1 to the $\widehat{\omega}'$ and by using the fact, that the 3.9-equivalence classes had unique reduced representatives. In 6.7 we had pointed out that we will face compatibility obstructions if we should seek to have compatible transformation patterns with similar properties for the entire word sequences $\widehat{\omega}$ and $\widehat{\omega}'$, also. Yet we discuss below, whether our transformation pattern for the words give rise to some kind of stabilizing sequence of edge-paths in our nested graph.

In particular our attention is drawn to matrices as they had been constructed in 6.2. When we try to realize the lines of one these matrices by a system of edge-paths on the corresponding levels, we get that the following: As in the first part of the proof, the initial segments (as long as all transformation processes are shortening) can clearly be realized by a nested system of paths that start at the $\widehat{\omega}$-vertex, and similar for the corresponding terminal segments; they give a system of paths that ends at $\widehat{\omega}'$. This gives us in principle two new (finitely) nested vertices in our graph, namely at the end of the first system of paths and at the beginning of the second system. Since the corresponding definitions of

"middle phase" in 6.7 and of tunnels in 7.6 had been set up in perfectly matching way, we can conclude that on each of the levels of consideration the corresponding vertices are connected by a tunnel. That way we see that our $\widehat{\omega}$-vertex and our $\widehat{\omega}'$-vertex can be connected by a generalized system of nested paths that contains on segment of nested tunnels on all arbitrarily fine finite levels. With similar arguments as in the first part of this proof (finiteness of the relevant part of the nested graph on each level, stabilization argument, upward extendibility) one can find as a connection of the $\widehat{\omega}$-vertex and the $\widehat{\omega}'$-vertex even some stabilizing system of nested edge-paths with one tunnel-segment in it. Denote this system of paths by ρ''. A closer look gives that the first and the third segment of ρ'' must be empty, since $\widehat{\omega}$ and $\widehat{\omega}'$ had been constructed with some minimality property, and if the corresponding segments were not empty, then they would provide a facility of how to extend ρ and ρ', as constructed before. I.e. ρ'' is in fact just an infinitely nested system of tunnels. □

7.12 Remark on why the result of Proposition 7.11 does not immediately imply Theorem 7.9:
If one recalls the definition of tunnels in 7.6 and that the condition "no pair-interchanges" (cf. the proof of 2.9) implies a natural global bound on the nesting depths of the corresponding transformation diagram, it might at first sight appear as if having found that such a tunnel-system connects $\widehat{\omega}$ and $\widehat{\omega}'$ automatically gives that $\widehat{\omega}$ and $\widehat{\omega}'$ are firmly equivalent, as desired. This turns out to be true, but it is not as immediate as it might seem at first sight:
Recall that in 7.5/7.6 our nested graphs had been set up in such a way that a nesting of vertices and of edges automatically implies that the corresponding compatibility property for the associated words or transformation processes had been fulfilled; on the other hand tunnels had not been inserted with a similar nesting condition, they had been individually drawn on each level without caring about the other levels. Hence having found an infinitely nested system of tunnels just implies that on each level some sequence of transpositions without pair-interchanges exists, but it does not imply that they are all compatible. This is, what still has to be proved below. However, this will definitely be the last step in the course of this proof. And, it will follow below in Proposition 7.13 from recycling the arguments as used in Proposition 7.11, already:

7.13 Proposition: *The reduced representatives $\widehat{\omega}$ and $\widehat{\omega}'$ that have by the method as described in 7.10 been constructed out of two equivalent word sequences ω and ω are firmly equivalent.*

Proof: In principle, we have to find some stabilizing system of edge-paths that connect $\widehat{\omega}$ and and $\widehat{\omega}'$. These edge-paths may have upward and downward leading edges, but they may use transposition-edges only. In addition, we want only to use such edge-paths that correspond to transposition-sequences that do not perform pair-interchanges (since this is a property that persists in a stabilization process and then implies the desired boundedness of the nesting depth). The ingredients for adapting the conclusions of the first part of this proof to our context are given: Since we only transpose letters but

never insert some, we try to find our paths just in a finite subgraph on each level, and the existence and the upward extension of each path are the result of 6.2. "Existence" in this context must be understood as follows: The first row of each matrix can clearly be realized by a path since here we never have different vertices with the same levels sitting inside different fatter vertices. And, since the transformations as listed in the matrices of 6.2 had already been synchronized, it follows that all remaining rows of these matrices can be realized by some edge-path inside the preceding one, i.e. we do not get trapped in the sense that on some levels we might just end at the wrong vertex with the same label. Having observed this, the stabilization arguments of the main part of our proof give the existence of a stabilizing system of edge-paths that connects $\widehat{\omega}$ and $\widehat{\omega}'$. Since according to 6.2 the transposition phases of the corresponding rows were free of pair-interchanges, and since we had used the corresponding transformation in the course of our stabilizing process, it follows that on each level the sequences of transpositions as given by our stabilizing system of edge-paths does not contain pair-interchanges. Then with the same arguments as used in the proof of 2.9(iii)–(iv) it follows that the nesting depth of the corresponding sequence of transformations is bounded, and hence that our stabilizing system of edge paths implies firm equivalence. Thus, even if $\widehat{\omega}$ and $\widehat{\omega}'$ might not have been the same vertex, they are at least firmly equivalent. This obstruction completes the proof of this proposition. □

Since 7.13 can also be applied to any two reduced word sequences, the uniqueness of the firm subclass and their independence from the method 7.10 of construction also follows. Hence 7.13 implies Theorem 7.9. □

7.14 Corollary : *The four strong equivalence relations for word sequences as introduced in 3.6 satisfy that within each equivalence class there exists precisely one firm subclass containing all reduced representatives.*

Proof: Indeed, this result follows as a corollary of the proof-techniques of 7.9, since they can also be used for finding minimal representatives in the strong equivalence classes. In these cases the nested diagrams must be constructed so that they only contain readings of a fixed path (cf. 2.17). Then Lemma 2.5 gives that any stabilizing sequence of edge path which exists in such a graph can only describe a strong equivalence relation. On the other hand, since the techniques of upward extension in 1.9 were based on considering readings (cf. the analogous discussion in the proof of Lemma 2.5), having to deal with smaller graphs than in the proof of Theorem 7.9 does not obstruct the existence of all upward extensions that are needed for imitating the scheme of conclusions in the proof of Theorem 7.9. □

7.15 Corollary: *If two reduced word sequences cannot be transformed into each other by merely interchanging commuting letters of each word so as to avoid pair-interchanges on each of the levels, they cannot be weakly equivalent.*

Proof: Consider two reduced word sequences. If they are firmly equivalent, they are weakly equivalent, also (cf. B.2.9). However, if they were weakly equivalent despite not being firmly equivalent, this would contradict to the result of 7.9. The corollary will

follow, since on the one the hand the arguments as given at the end of the Proof 2.7 imply that "no pair-interchanges" in this context give a "bounded nesting depth", and on the other hand we will below be giving the arguments that in this particular context any firm equivalence can be performed without pair-interchanges:
2.9(iii) already implies the weaker result, that triple interchanges can never occur in a transformation pattern that belongs to a strong or firm equivalence. In addition, 2.9(iv) also rules out most of the possibilities for pair interchanges, namely all apart from those that belong to a cancelling pair of letters. Further, it it not hard to extend the proof-techniques of 2.9(iv) in such a way that they rule out a pair-interchange $AA^{-1}BC \longmapsto BCAA^{-1}$ in any situation where on some level there occur letters between the split-up letters of A and of A^{-1} which do not commute with A. Hence a strong equivalence transformation can only contain a pair-interchange, if one of the corresponding pair is a cancelling pair and, in addition, satisfies all conditions of 3.4(i)–(ii) so that the cancellation process could be compatibly extended throughout the entire word sequence. Since this situation cannot be given for a reduced word sequence, we indeed get that the non-avoidability of pair-interchanges in this case already rules out that our word sequences could have been weakly equivalent. □

7.16 Remark, why we do not obtain in any case minimal unique firm subclasses of triple sequences within our weak equivalence class:
Recall that by C.1.9 word sequences are in principle only used as an abbreviation for triple sequences, and that triple sequences are the only combinatorial concept that is capable of uniquely describing any path in the plane as developed so far. If we want to discuss in how far our uniqueness result for word sequences gives also rise to triple sequences, we have to recall that not always the triple sequence was uniquely determined by the word sequence, cf. C.1.9. In principle there are three different cases:
(i) Word sequences, which contain vertical and horizontal letters. This may be regarded as the standard case, and in this case the corresponding triple sequences are uniquely determined by the word sequences. Hence the uniqueness result for word sequences in this case gives rise to a uniqueness result for triple sequences as well.
(ii) Word sequences which only contain horizontal or only contain vertical letters. These word sequences correspond to paths, that are entirely vertical or entirely horizontal, respectively. In this case we could have the phenomenon, that various parallel paths with identical behaviour in the non-constant coordinate could have been described by the same word sequence. In this case we have to discuss, whether amongst these paths and amongst the first and last entries there could also be more than one entry describing the same path. — Well, there are, if the constant coordinate of a path for some level falls inside the region of a corridor. In that case, by C.1.3 the zero-movement in the direction where the coordinate is constant could also be regarded as a partial act of crossing a corridor, or, in case our path lies on the boundary of a corridor, as two partial acts of crossing into inverse directions. However, such partial acts of crossings cause truncatable letters of the corresponding direction, and then our word sequence could be shortened. A minimal word sequence will hence have none of these truncatable letters, so that the first entry of the corresponding triple sequence

will have to associate the start- and the endpoint of our path to some side of the corridor. If it should be not the same side, by 3.2/3.3/C.1.6 our word sequence will see a truncatable letter, and this cannot happen in a reduced word sequence. Hence it has to be the same side; however, there is still a choice between two different of the sides, and this choice causes that in that case different minimal triple sequences can vary by more than just containing different but firmly equivalent word sequences. However, note that according to (i) and (iii) this will essentially remain the only possibility for such additional non-uniqueness properties of minimal triple sequences.

(iii) If our word sequence is empty, and hence our path is constant, then the phenomenon as described in (ii) could affect the vertical coordinates and the horizontal coordinates, none, one or both of them.

Recall that the above arguments only affect free equivalence classes. *Hence in fixed equivalence classes we have minimal triple sequences that are unique up to firm equivalence of their enclosed word sequences, and in the semi-free cases also* (since the above arguments gave that skipping between to two possible minimal representatives requires the readjustment of both, the first and the last entry of the corresponding triple).

The above observation does also have its impact on what might conjectured as being a corollary of the proof-techniques as used in 7.9–7.13, namely that any two weakly equivalent word sequences might be transformed into each other by firstly going over shortening transformations only, then over such transpositions only that give a firm equivalence, and in a final phase having only to use enlarging transformations. In that form the result would only be correct if we neglected our convention C.1.9, i.e. that word sequences are only used as abbreviations for triple sequences. When we do not want to forget about this convention, we have to add the phenomena as discussed above. Thus we see that only the following weaker result is valid:

7.17 Corollary: *Let ω and ω' be two weakly equivalent word sequences. Then there exists a compatible transformation pattern that can be subdivided into three phases such that, if $\widehat{\omega}$ and $\widehat{\omega}'$ denote the intermediary states that our word sequence acquires at the end of the first and the second phase, respectively, the following conditions are satisfied:*

(1.) *The transformations that turn ω into $\widehat{\omega}$ are all of shortening type.*

(2.) *The transformations, that turn $\widehat{\omega}$ into $\widehat{\omega}'$ are either —*

(i) *only such transpositions that make the corresponding pattern a pattern of firm equivalence — or*

(ii) *one or two pairs of processes per level, where some letter is firstly appended at one end of the corresponding word and then truncated at the other end again. If on one level we actually should see two pairs of these processes on a single level, then the two involved letters must be commuting.*

(3.) *Finally, $\widehat{\omega}'$ is transformed into ω' by using a compatible pattern of enlarging transformations only.*

Proof: By having constructed the corresponding stabilizing sequence of paths in the course of the proof of 7.9 it is clear that when neglecting adjustments of the first and third entries of our triple sequences, we get the above result (without, of course, noticing

(2.ii)). Now recall that our Phases 1 and 3 have been obtained by constructing explicit transformation patterns in the proof and that in the course of our discussion of word sequences the corresponding processes B.2.2(iii)/1.1(iii), that adjust the first and third entries of triple sequences, have usually fully been involved. Hence we cannot have missed any such adjustments as in (2.ii) here, in particular, since those adjustments require a combination of shortening and enlarging processes. However, Phase 2 in the above claim had been constructed in a different way, namely by an indirect proof that showed that the corresponding word sequences have to agree up to such transpositions of their variables that can be performed by firm equivalence. That way we had missed that such readjustments as 7.17(2.ii) might be necessary on the level of triple sequences. However, the following arguments show that the necessary adjustments can never exceed what has been stated above: If the corresponding $\widehat{\omega}$- and $\widehat{\omega}'$-words should contain letters of different basic direction, then, as explained in 7.16(i), the corresponding adjustments of triple sequences cannot be performed. On the other hand, if our words should contain letters of at most one basic direction only, then by B.1.8 these letters are all non-commuting, and hence transpositions cannot be performed. This justifies that we have stated our corollary in the above "either-or"-form. Further, by the same arguments as in 2.1 the list of truncatable letters is bounded, so that any attempt to perform more of the 7.17(2.ii)-processes as permitted would only result in undoing some of them by the inverses again. Since this never is necessary (cf. 2.17/2.4), what has been claimed above is hence sufficient to transform any two weakly equivalent triple sequences into each other. □

7.18 Lemma: *Let ω be as in 7.2. Then ω has a compatible reduction scheme which brings ω to the empty word sequence and which does not contain any elementary transpositions.*

Proof: By assumption there exists a compatible reduction scheme to transform ω into the empty word sequence, but this reduction scheme may contain elementary transpositions also. We then try to transform such a reduction scheme into one which does not have transpositions anymore. In principle we do this, on the one hand, by omitting all transpositions, and on the other hand by performing the same cancellations as in the scheme originally given, even if the corresponding variables now are in different places. However, since we could only perform transpositions if the two variables to be interchanged had been defined to be of commuting type, our omitting of transpositions cannot have made a legitimate cancellation illegitimate, it could only have turned a cancellation of adjacent variables into a far-distance cancellation or vice versa. (See A.4.8(Fig.d) for a cancellation pattern that avoids transpositions.)
As in 7.1 we first have to apply the above arguments to each word of ω separately (on that level the arguments given above are complete) and then we can use the Königslemma to obtain compatibility for our new reduction scheme, also. As in 7.9–7.11 the Königslemma type of conclusion can be best played to a nested graph as constructed in 7.5/7.6: A system of shortening operations for a word which avoids transpositions, is a monotonous downleading edge-path on a fixed level that avoid certain edges. Since

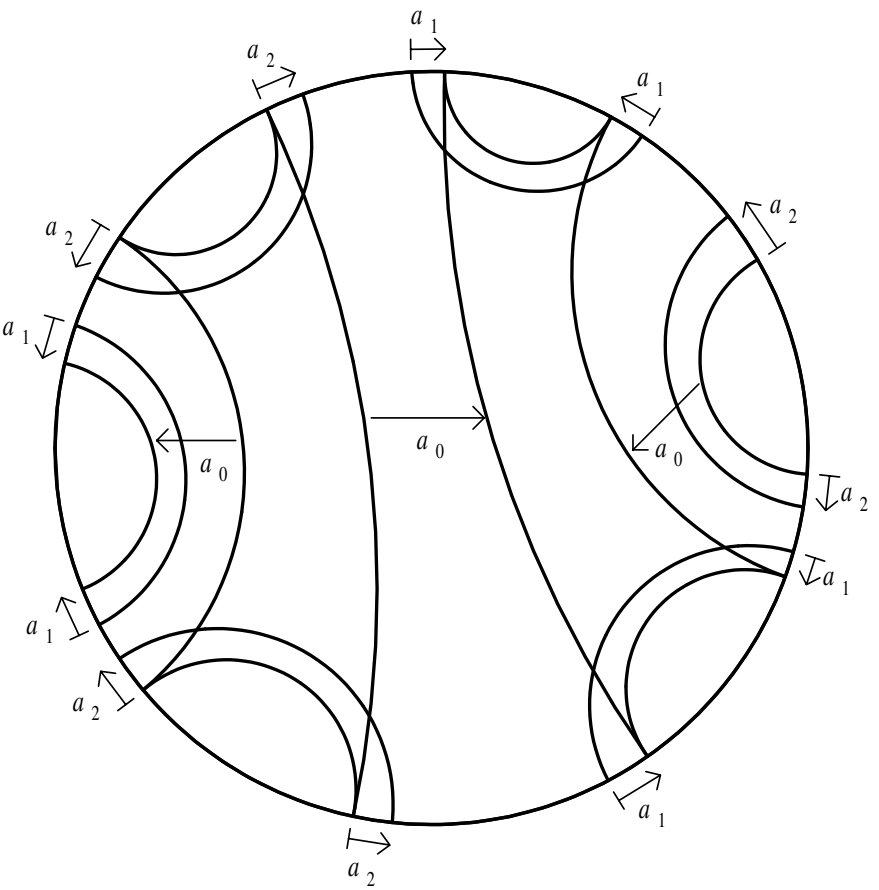


Fig.m$_1$ (belonging to 8.1):
This figure gives an example for a realisation diagram and it pictures two of its levels, cf. Fig.m$_2$. In particular, both of these figures provide an example for the turning-away phenomena as described in 8.3(ii).

we know that at least one facility to avoid transpositions exists on each level, we get a sequence of such edge-paths that, as in the proof of 7.9, can be turned into a stabilizing system of nested paths. Since this stabilization was based on skipping to a subsequence and then using combinatorial results to fill in the gaps of the subsequence, this type of conclusions definitely requires the result of the discussion of 3.13/7.2, i.e. that, other than in the general case of 1.9, it never takes an insertion of adjusting transpositions for less fine levels to synchronize shortening cancellations or truncations. □

§8. On Realization Diagrammes

Recall that it was a part of our proof-programme 1.21 to turn the combinatorial information that is contained in a transformation pattern of word sequences (cf. 1.20) into a continuous function giving the relative homotopy between the associated paths. In 1.21

we have already pointed out that transformation diagrams must be considered as one intermediate step of this process; and the realization diagrams that are to be defined and discussed in this section must be understood as the main tool that will help us to complete this step of our proof-programme. These diagrams are set up so as to provide a combined picture of transformation diagrams and of split-up graphs.

8.1 Definition of realization diagrams: *Case (i):* Let $u, v : [0,1] \to X$ be two paths and let $\omega = (\omega_i)_{i\in\mathbb{N}}$, $\eta = (\eta_i)_{i\in\mathbb{N}}$ be two word sequences which can be read from those paths. Assume that we are given a compatible transformation pattern in the sense of 1.17 that brings ω into η. Note that this set-up implies that there exist in principle $3\cdot\infty$ diagrams to visualize the relations between the corresponding words ω_i and η_i : Namely, on the one hand there exist infinitely many split-up diagrams (cf. C.3.4) for each of the word sequences that visualize the relations between adjacent words of these word sequences, and on the other hand there exist infinitely many transformation diagrams (cf. B.2.5/1.3) that represent the corresponding elementary transformation processes that turn ω_i into η_i. The idea of a realization diagram as it shall be described below is putting the information of all these diagrams into one single picture (of course usually only finite partial aspects of such diagrams will then be considered). We describe below how such a diagram is build up:
$\mathbb{D} := \{(x,y) \in \mathbb{R}^2 \mid \|(x,y)\| \leq 1\}$ is the domain of definition for any realization diagram. The upper semicircle $\{(x,y) \mid y = +\sqrt{1-x^2}$ is via $(t, \sqrt{1-t^2}) \to \frac{t}{2} + \frac{1}{2}$ bijectively associated with $\mathrm{Dom}(u)$, i.e. the domain of definition for u, and the lower semicircle is via $(t, -\sqrt{1-t^2}) \to \frac{t}{2} + \frac{1}{2}$ bijectively associated with $\mathrm{Dom}(v)$. Hence the reading domains for any of our letters of ω and of η can be found as segments of $\partial\mathbb{D}$. If in the corresponding transformation diagram for ω_i and η_i two letters have to be connected by a line in our transformation diagram, the corresponding reading domains are now to be joined by a corridor. For that purpose we think that $\mathbb{D}$ is equipped with a metric that makes $\mathbb{D}$ to be used as a support of the hyperbolic plane $\mathbb{H}^2$ according to the Poincaré Disk Model, and that the boundaries of the corresponding corridors are then straight lines with respect to hyperbolic geometry. I.e. from a Euclidean point of view the boundary lines of our corridors are circle segments that are perpendicular with respect to $\partial\mathbb{D}$. For those reading domains which correspond to truncated or to appended letters the boundaries of the corridors emanating from these letters are then drawn to (−1,0) or to (+1,0) depending on whether the corresponding letter is placed at the beginning or at the end of the word, respectively. This convention in analogous to the conventions that the corresponding curves of transformation diagrams have to end at a side bar of the frame. Observe that within $\mathbb{D}$ near (−1,0) or near (+1,0) we also can get corridors which are degenerated to semidisks: I.e. one point of $\partial\mathbb{D}$ then must be regarded as the degeneration of the other corridor boundary. Pictures which look like this will arise when at the start point $u(0) = v(0)$ both paths are already in the reading domain of the same corridor. Then the parameter values where those paths exit from that corridor will be connected by a hyperbolic straight line bounding a Euclidean semicircle. Another typical situation for such a geometry might be if $u(0)$ is considered to be inside the reading domain of a corridor, but $v(0)$ is not. In the standard situation of this case the

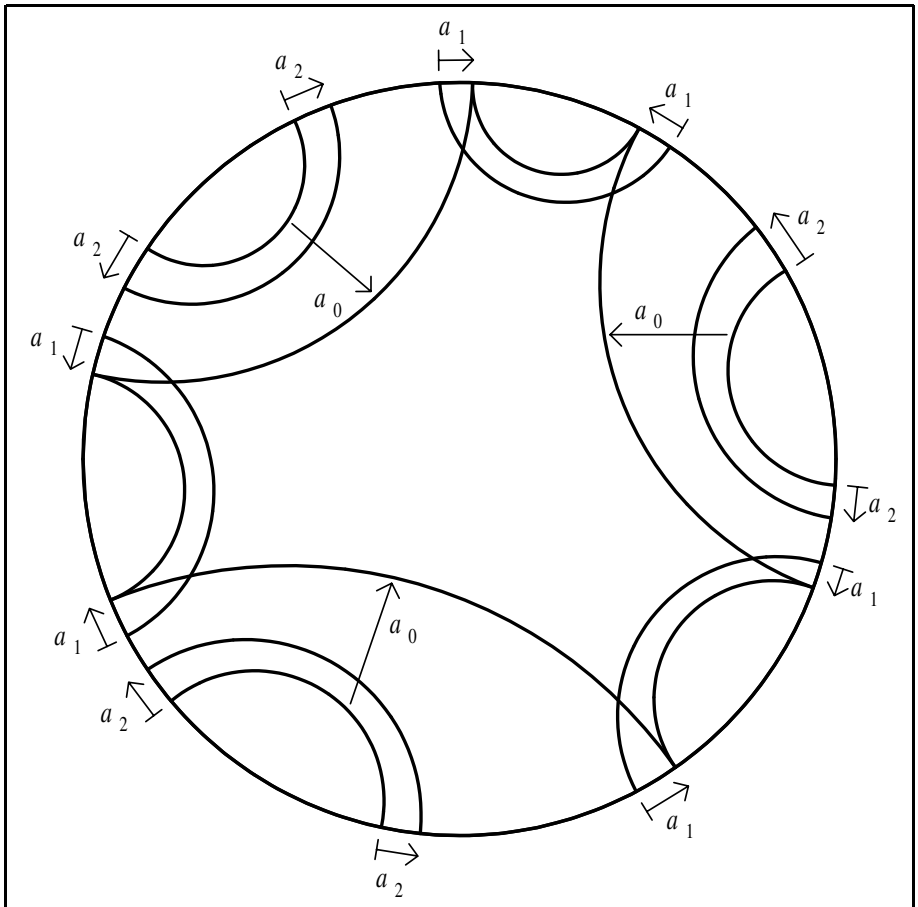


Fig.m$_2$ (belonging to 8.1):
This figure should be viewed in connection with Fig.m$_1$. It provides an alternative way of how a realization diagram might look like that is associated with the same reading that we saw in Fig.m$_1$.

corresponding first letter of ω will have to be truncated in the course of transforming ω into η, and then the parameter value where u leaves the corresponding reading domain is to be connected by a semicircle with (−1,0). However note, that it cannot be ruled out that we also get some *"peaky corridors"* where one of the intervals of $\partial\mathbb{D}$ is degenerated to one point. For example, this happens if some initial sequence of ω is to cancelled and if this sequence is followed by a letter to be truncated after these cancellations have been performed. The endpoints of the boundaries emanating from the reading domain of the truncatable letter will have both to be at $(-1,0)$, and this gives a peaky corridor.

Case (ii): We draw similar realization diagrams when we are only given one path w with an according word sequence η, and a cancellation pattern for that word sequence. In that case the entire circle $\partial\mathbb{D}$ is identified with $\mathrm{Dom}(w)$ and $(-1,0)$ is used as a parameter value for the start- and the endpoint.

8.2 Addendum zu 8.1: *(Can there be double-peaky corridors?)*

At this place we should also try to imagine whether we could get corridors with two peaky ends, since in that case there would by considerable difficulties in realizing such corridors according to the hyperbolic geometry. Note that in the corresponding situation for transformation diagrams (cf. 3.7(iii)–(iv)) we pointed out that we might have to deal with lines that start at one of the side bars and end at the same or the other side bar, and if we try to construct our realization diagram by taking the corresponding lines from a transformation diagram and straightening them according to the hyperbolic geometry, we might well end up with pictures like that of double-peaky corridors. On the other hand, note that according to the above definition (i.e. connect the boundary points of corresponding reading domains by lines and regard the enclosed domains as corridors), such double peaky corridors do not arise. We will take the latter point of view in our definition. However, note that a good deal of our attempt to understand the variety of phenomena that within a realization diagram can hold, will be based on discussing how they do arise from transformation diagrams, and in this context one will have to take the disappearance of such lines that in transformation diagrams connect two points both lying on one of the side paths into account. In that context we will have to return to considering the (3.7(iii)–(iv))-lines, and we might use the term "double peaky corridors" to refer to the corresponding phenomena.

8.3 Lemma: *Consider a realization diagram that according to 8.1/8.2 is constructed from a transformation pattern of word sequences that satisfies all properties as described in 7.17. Use all the corresponding notation as in 8.1. Then the corridors of that diagram satisfy the following two properties:*

(i) If two corridors of the same level index have domains at which their interiors intersect (if they cross or overlap), then these corridors belong to commuting variables.

(ii) Let A, a be letters of ω_{i-1} and ω_i, respectively (or of η_{i-1} and η_i, respectively), such that a belongs to a split-up-corridor of A. Let $\widehat{a}$ and $\widehat{A}$ be the corresponding corridors of our realization diagram that emanate from the reading domains of A and a, respectively. By construction, at the corresponding reading domains the place where $\widehat{a}$ emanates is inside $\widehat{A}$, and due to matching endpoints it is associated to one of the sides of $\widehat{A}$. Then either does $\widehat{a}$ lie entirely inside of $\widehat{A}$ with the corresponding boundaries entirely matching, or $\widehat{a}$ intersects the boundary line of $\widehat{A}$ to whose side it is associated and it may in a similar way intersect the boundary line of a corridor the belongs to some other letter $A^{(-1)}$ in its end phase, but apart from that it does not overlap with any other corridor that is associated to a variable of $\overline{\mathcal{A}_{i-1}}$ of the same basic direction as A.

The Proof of this lemma is lengthy and hence split up into 8.3–8.10. In 8.4–8.9 we will thoroughly describe the Proof of (ii), while the proof of (i) will be restricted to a brief description in 8.10 to adapt these arguments for that purpose. Essentially the proof will be based on a discussion of what happens when a transformation pattern is turned into a transformation diagram (cf. 1.20/1.18/1.3), and when this is turned by 8.1/8.2 into a realization diagram. Hence the first step of the proof consists of the following:

8.4 Analysis of the intersection pattern of lines and corridors in transformation- and realization diagramms:
It is an immediate corollary from the Definitions 1.3 and 8.1 that in order to transform a transformation diagram into the entries of the corresponding realization diagram of the same level it only needs the following steps:
(1.) Thicken the curves of a transformation diagram up to corridors.
(2.) If the reading domains that give the end segments of corridors of the realization diagram overlap for some path, the ends of the corresponding corridors that are all disjoint in a transformation diagram have to be slid aside so that the adjacent corridors belonging to commuting variables may then partially overlap.
(3.) The vertical side-bars of the frame of the transformation diagram have to be collapsed, in addition the upper and the lower bar have to be bend to become semicircles, and apart of that a homeomorphism of the frame has to be applied taking all the corridors with it and mapping the interior of the frame to $\mathring{\mathbb{D}}$.
(4.) The geometry has to be straightened up, i.e. each corridor as it looks as a result of translating step by step the transformations of the words into pictures, and applying the distorting homeomorphism has to be replaced by a corridor connecting the correct end domains and having straight boundaries with respect to the hyperbolic geometry.
In particular, a good deal of deducing 8.3(ii) will be based on looking at transformations diagrams on the one hand, and on analyzing what could happen if such diagrams are according to (1.)–(4.) transformed into realization diagrams. In that context the following observations are relevant:
▷ Clearly the Processes (1.) and (3.) cannot affect whether the according domains do intersect or not.
▷ The process (2.) is also irrelevant for our discussion, since our discussion only affects the intersection of corridors that are associated with variables of the same direction.
▷ The process (4.) will give the most difficult discussion, since if things are straightened so as to match domains at the sphere at infinity, it is hard to tell how pictures in a certain bounded domain in the interior might change. However, on the other hand hyperbolic geometry tells us, that lines and corridors connecting domains on the sphere at infinity will or will not intersect depending on the order of the points to be connected. In view of this the process (4.) may be regarded as a process which may undo crossings or slide them to other places but which cannot generate crossings if there had not been any before. Of course we will not get around to discuss the possible effects of combining various transformation diagrams, each of which just shows one elementary process (cf. 3.7).

We have not discussed yet, how we want to bring the corridors of the level indices $i-1$ into our picture. For that purpose we continue by
8.5 Refining the Analysis 8.4 to considering two adjacent levels:
In principle this refinement shall happen by applying the same processes of C.1.7(i)–(iii) that give the reading of the $(i-1)^{\text{st}}$ word of a word sequence from the ith word of a word sequence to our transformation diagram with lines thickened up to corridors. We have to take into account that the $(i-1)^{\text{st}}$ word of a word sequence is not entirely determined

by the ith word, but for our aspects it in many cases is. Since the corridors in realization diagrams connect reading domains that are defined on $\partial\mathbb{D}$, all the problems in which all the variables of overlapping reading domains appear in a word sequence are irrelevant here. Hence the only remaining uncertainty for the $(i-1)^{\text{st}}$ word affects the first and the last letter of our word (in principle, of course: the first and the last letter of each of the basic directions), where, if our word should contain only one of the two letters giving a split-up pair, on the $(i-1)^{\text{st}}$ level the corresponding letter might or might not be there. Hence we will have take both possibilities into account in the subsequent discussion.
Consequently, let us now proceed by considering the transformation diagram that reflects all transformations that are applied to the ith word by a given pattern. Hence the corresponding transformation diagram results by putting various transformations diagrams showing elementary processes above each other, and consequently such a diagram is subdivided by various horizontal bars that arise from the frames of its elementary components. We already think of the lines in our transformation diagram as thickened according to 8.4(1.). We now try to add the information of the corresponding diagram for the $(i-1)^{\text{st}}$ level to our picture where the corresponding thickened lines of the $(i-1)^{\text{st}}$ picture shall in our picture be drawn as hatched corridors. The reading mechanism as described in the preceding paragraph permits to draw parts of the hatched domains, namely in the respective neighbourhoods of the top bar, of the bottom bar and of each of the horizontal bars of the old elementary frames. Drawing these hatched areas as a preliminary step will in principle give that, wherever a horizontal bar is intersected by two neighbouring thickened lines that belong to a split-up pair of letters, the area covered by these thickened lines and the area in between is hatched, at least in a neighbourhood of the corresponding horizontal bar. Having put these hatchings in our diagram leaves us with finding the appropriate method for how to connect them inside each of the elementary frames. The following Propositions 8.6–8.9 are devoted to discuss how these hatchings have to be connected (recall that we are not free in our choice, because how to connect them is in principle defined by the corresponding transformation diagram of the $(i-1)^{\text{st}}$ level) and to what extend the boundaries of the hatching domains will match with our thickened line of the ith level. We proceed by discussing these questions separately according to the type of process that is represented by one of our elementary frames:

8.6 Proposition: *Transpositions cannot originate an intersection behavior of corridors in a realization diagram that is not covered by the statement 8.3(ii).*

Proof: In principle, this proposition already follows, because 8.3(ii) is a statement about corridors of the same basic direction, and thus we may restrict our picture accordingly. Hence transpositions are not visible for us. If however, just for moment, we recall the corridors that are associated with the other basic direction back into our picture, we see that the intersecting of our corridors of different directions does not have an impact on the split-up structure. Even if such intersections should take place, we are neither forced nor tempted to extend the vertical boundaries of the hatching domains in such a way that they would not follow the corresponding corridor boundaries of our thickened

line, even if they intersect corridors of the other basic direction. Hence we need not worry about transpositions. □

8.7 Proposition: *Apart from coinciding corridor boundaries cancellations can only origin a turning away phenomenon in the corresponding realization diagram that is covered by the statement 8.3(ii).*

Proof: Of course, if we come to see non-coinciding corridor boundaries and hatching domains, it must be a split-up letter which is to be cancelled. Let a_2 be the corresponding $\overline{\mathcal{A}_i}$-variable, and assume that the corridor that is associated with a_2 comes into the region of consideration from the top. If a_2 should belong to a split-up corridor, the next letter (at least the next letter of the same basic direction that corresponds to the adjacent corridor in our picture), must be the corresponding split-up partner of a_2, we denote this corridor and variable by a_1. On the other hand, since a_2 was assumed to be involved in a cancellation process, the next letter on the other side of a_2 must be a_2^{-1}. We now have to distinguish cases according to the properties of the letter that follows this a_2^{-1} and that is of the same basic direction as a_2. There will be two cases which shall essentially be distinguished by calling this letter either "$a_1'^{-1}$" or "a_2'". There are only these two cases, since by C.1.6 this letter can only be from the same line of corridors as a_1 or as a_2, and we have chosen our notation in such a way that it should be clear which notation is for which situation. We cannot rule out that $a_1' = a_1$ or that $a_2' = a_2$, but on the other hand this is not enforced by our situation. It could be that the dashed corridors are just from the same line of corridors as the undashed ones, but that they do not represent the same corridors. Let us now discuss the corresponding two subcases separately:

CASE ONE: Our forth letter is a_2', i.e. the corresponding segment of the ith word on the top bar reads "$\dots a_1\ a_2\ a_2^{-1}\ a_2' \dots$". The corresponding segment of the $(i-1)^{\text{st}}$ word on the top bar is "$\dots A \dots$", where A is the root-letter in "$a_1\, a_2$". According to C.1.7(i), the letters "$a_2^{-1}\, a_2'$" do not correspond to any letter of our i^{th} word. On the bottom bar of the elementary frame of consideration we have on the ith level (since a_2 has been cancelled) only "$a_1\, a_2'$", and they are now the split-up variables that belong to the corridor A. Consequently the hatched area marking the A goes vertically through the region of the consideration and the corridors that are associated with the variables a_2 and a_2' first turn away from the hatched area and then leave the region of our elementary frame by crossing the top bar at the places that are marked by the letters a_2^{-1} and a_2', respectively.

CASE TWO: The forth letter is "$a_1'^{-1}$". Consequently the $(i-1)^{\text{st}}$ word on the top bar is "AA^{-1}", and at the bottom bar, after the corresponding cancellation of a_2, we get on the ith level "$a_1\, a_1'^{-1}$", while the corresponding segment of the $(i-1)^{\text{st}}$ word on the bottom bar is empty. Consequently, while on the ith level we see a cancellation of a_2, on the $(i-1)^{\text{st}}$ level we see a cancellation of A. Hence the hatched area marking the A-corridor also performs a cup-figure in the segment of the diagram under consideration. And the two corridors belonging to a_1 and a_1^{-1} which are approximately vertical in the segment of our diagram under consideration, appear then as corridors turning

away from the hatched area. We should point out that in case of $a_1 = a_1'$ it could happen that these variables are cancelled on the next step below, and then within the straightened realization diagram, these corridors need not appear as corridors which turn away from the hatched area any more. However it could also happen that in the bigger diagrams these corridors go further vertically down, and then they must be regarded as corridors that are turning away. □

Having completed the discussion of both cases, we see, that cancellations can only originate coinciding boundaries and the *"turning-away figures"*, as described above. However, we still need to discuss what could globally come out if such turning-away figures may be plugged together to a bigger diagram. This will happen in the proof of the next proposition:

8.8 Proposition: *Cancellations cannot originate an intersection behavior of corridors in a realization diagram that is not covered by the statement 8.3(ii).*

Proof: If turning-away figures are plugged together, it could happen for one hatched corridor which is going from the top bar to the bottom bar that there are some corridors that join on from one side and later turn away (and this may happen more then once, even if in a straightened picture then the corresponding on- and away-turning corridor would not touch that hatched area any more), and it could happen that the corridors that have turned away later rejoin the hatched area again.
Without assumption 7.17 the corridors having turned away might also be combined to circular lines, but since we had been able to quote 7.17 when stating the assumptions of that lemma, this is ruled out. However in any case we should point out that it is impossible that any corridor of the ith level intersects a corridor of the $(i-1)^{\text{st}}$ level of the same basic direction. This cannot happen, since there is not any local picture which permits such intersections (at least, we will see this after having discussed the truncation cases below), and on the other hand such a crossing cannot arise by straightening an according combination of turning-away pictures if one looks at them as follows: Note that according to the above pictures, since the left split-up corridor can only turn away to the left and the right corridor can only turn away to the right, what could happen to a small corridor that has turned away from the hatched area of a big corridor can be understood as follows:
Since in our picture corridors of the same level index cannot intersect each other, from a combinatorial point of view our corridors cannot do more than to subdivide the region of our realization diagram into various polygons. And a small corridor of the next level, when turning away, enters this region. Without loss of generality, assume that the act of turning away was a turning left. If now this corridor is going to enter the same or another hatched area, then according to having discussed all possible pictures above, we can be sure that our corridor will join on the hatched area by turning left again: There had not been a local picture that permitted an act of crossing, and if our corridor would join on by turning right, then the left side of our small corridor would go in the interior of the hatched one, and the right side would coincide with ("match") the boundary side

of the hatched area. The combination of having first turned away by turning left and then joined on by turning right, would give that the matching/non-matching conditions would be just the other way round in the hatched area in which our small corridor is before turning left. Now, such a situation gives a contradiction as follows:
Recall that in the original picture of combined elementary transformation diagrams, after each essential move the corridors have to pass one of the horizontal bars where they give valid markings of reading domains. And according to the reading conditions B.1.4 for each split-up letter it is clear whether the start point of the reading domain coincides with the start point of the root-letter and the end point does not, or vice versa. This corresponds to the fact that our corridor lattices $\mathcal{N}_i$ had been constructed in such a way that precisely one of the boundary lines of the split-up corridor coincides with the boundary line of the root-letter, and the other was placed in the interior of the domain covered by the root-corridor. Hence the combination of turning away by turning left and joining on by turning right is impossible, so that our small corridor, if it happens to come upon a hatched area again, can only join on by turning left again. Then it is again on such side of the corridor-boundaries of our polygon that, if it should turn away again, it would just again enter the interior region of the same polygon. Hence there is not any chance for our small corridor ever to leave that polygon, and that way we have by now seen that cancellations cannot originate any illegitimate intersection behavior of corridors as claimed by this proposition. □

Observe that considerations as used for the proof of 8.8 will implicitly be also used in the subsequent proof of 8.9, so that in that proof we can mainly focus on arguing that the consideration of truncations does not bring essential new pictures into play.

8.9 Proposition: *Truncations cannot originate an intersection behavior of corridors in a realization diagram that is not covered by the statement 8.3(ii).*

Proof: In principle there are more than two subcases to be discussed, but we will gather together all subcases just under two headlines. The two subcases are, whether the first two letters of a word (of course, the discussion for the end of the word is analogous) are a split-up pair, or whether our word starts with just a second letter of a split-up pair appearing. Note that the following considerations will not be based on looking at the realization diagrams, since the phenomena that we see in these diagrams may be a superposition if various elementary processes. Hence, as in the preceding propositions 8.6–8.8, we look at the transformation diagrams and have to keep in mind that the corresponding picture of a realization diagram will arise from performing the steps 8.4(1.)–8.4(4.) and from concatenating the straightened versions of various elementary frames of transformation diagrams. Observe that all corridors of truncated variables cover areas of a transformation diagram whose left boundary is a part of the left vertical bar of the frame. This is due to the fact that by 2.1–2.3 the endpoint of the corresponding path cannot be moved by our process, and only those letters can be truncated that contain this endpoint. Hence, if a letter is to be truncated or appended, the corresponding reading domain must include the initial point of the parameter domain of

our path, and in a transformation diagram such an interval extends to a hatching area which is bounded by a part of the left vertical bar. These phenomena will give that a lot of pictures that we see in this case just appear like the picture that we saw before in 8.7, with some part of the pictures being cut of. In the following we use the letters A, a_1 and a_2 in the same way than we did in 8.7.

CASE ONE: The first two letters (of the basic direction under consideration) of our word are "$a_1\, a_2$". Hence the corresponding word of the $(i-1)^{\text{st}}$ level must start with "A". Since only one of our letters $a_1\,, a_2$ can be truncated, it must be a_1. Based on these assumptions A might or might not be truncated simultaneously. If it is not truncated, then our diagram sees the a_1-corridor disappear at the left bar from our region of consideration, but this is not a turning away process, since the corresponding part of our diagram is entirely covered by the region that has been hatched in order to mark the A-corridor. In this case the right boundary of A and the right boundary of a_2 coincide for the entire elementary frame. In the other case where we have to discuss the possibility that A in addition is truncated, then we see a turning away phenomenon: The a_1-corridor stays within A and disappears together with A from our region of consideration, but the a_2-corridor then is turning away from A, similar as in 8.7. Note that in this subcase, where both letters of the split-up corridor show up, we are by C.1.7 not allowed to truncate none of the a_i, but A only.

CASE TWO: Now the word on the level $(i+1)$ just starts with the letter a_2, and the word on the level i starts with A. In the case to be discussed here the conditions of C.1.5 rule out just one of the the four combinatorial possibilities: If a_2 is truncated, then A must be truncated also. Since on the other hand we want to see at least one truncation, in fact only two possibilities remain to be discussed: A must be truncated, and the a_2-corridor may or may not be. In the first case we see in our picture the hatched areas marking the A-corridors disappearing via the left edge of our frame, and the a_2-corridor follows the same route (i.e. there is not any turning away phenomenon, the split-up corridor stays in the hatched area), and in the second case the A-corridor looks as before and in this case the a_2-corridor as the only visible split-up corridor turns away from it and remains in the picture as in the first subcase.

Hence the truncation case gave in principle only phenomena as they are already known from the cancellation discussion (apart from the aspect that now in 8.9 a part of the old 8.7-figures have been cut away at the boundary of our picture). Hence in particular the discussion of 8.9 does not overthrow the result we deduced at the end of 8.8: There cannot be some crossing of the corridors of the $(i-1)^{\text{st}}$ level with corridors of the ith level of the same basic direction. Note that in the latter case the left boundary of the a_2-corridor might or might not coincide with the left bar of the elementary frame. This corresponds to the fact that, since a_2 is actually not truncated in our picture, it could be that the corresponding act of crossing is completed or is not. □

Having discussed all cases in 8.6–8.9, the proof of 8.3(ii) is complete. □

8.10 Proof of 8.3(i): In principle, 8.3(i) follows from the same scheme of arguments that have been used to deduce 8.3(ii) in 8.4–8.9 before (i.e. by translating the situation

from transformation diagrams to realization diagrams according to the steps 8.4(1.)–8.4(4.)). However the situation is much simpler here, since we do not have to discuss how transformation diagrams of adjacent indices are related to each other, when we attempt to draw them into the same frame. In this case the lines of transformation diagrams are directly thickened to giving corridors. Then taking into account what has been said at the end of 8.4 (that even the straightening process cannot generate crossings that have not been there before), the corresponding desired statement follows directly from the corresponding properties of transformation diagrams: Cancellations and truncations are pictured without any crossings, and since transpositions and far-distance processes are only permitted if the corresponding variables have been regarded to be "commuting", only such corridors may intersect in realization diagrams. □

With the following paragraphs 8.11–8.20 we now perform Steps (ii) and (iii) of our proof-programme 1.21.

8.11 On how to define the interpolating functions H_i of a realization diagram: $H_i|_{\partial\mathbb{D}}$ will be given by the parametrizations of u and of v (cf. Def.8.1) and $H_i|_{\mathbb{D}}^{\circ}$ will have to be defined in the following. Note that on the level i only finitely many boundary lines of corridors are contained in our picture. Hence the system of these lines gives a kind of cell-subdivision of $\mathbb{D}$. The main steps of the definition of the ith interpolation function H_i will be given in 8.13–8.19 when this cell-subdivision will be used to extend the definition from $\partial\mathbb{D}$ to each level of the skeleton of this cell-subdivisions separately. However, as a preparatory step, the subdivision as originally defined in 8.1 has to be modified in one its aspects:

8.12 Inserting double-peaky corridors into realization diagrams:
The origins for double-peaky corridors have been described in 8.2. At that time we agreed not to draw them, but pointed out that there might be two types of such corridors: The first of them belongs to lines as described in 3.7(iv), that within a transformation diagram start and end at the same side. Such lines correspond to letters that are first appended and truncated at a later state of the transformation process. However, since by quoting 7.17 in 8.3 we had been able to restrict our considerations to such transformation patterns that in the first part only perform shortening operations and postpone all enlarging ones to the second part, we can meanwhile rule out the existence of such lines. This does not hold for the second type of lines, that might give rise to double-peaky corridors, cf. 3.7(iii): We could perfectly have letters in our transformation pattern that are firstly appended at the beginning and then truncated to the end again (cf. 7.16(ii)). The corresponding line of such a transformation process is a line the connects the two vertical bars of transformation diagrams, and such a line would within a realization diagram gives a double-peaky corridor that has to connect the points $(-1,0)$ and $(+1,0)$. For the purpose of the definition of interpolation functions H_i in 8.11–8.19 we want to draw these corridors, also. We hereby decide to use as the appropriate geometric form the form of a single line, however, this line is, unlike the other lines in our picture, not be treated as the boundary line of some corridor, but as the degeneration

of an entire corridor. We use the second half of this paragraph for explaining our vital interest of having the double-peaky corridors in our picture:
It is a vital property in the course of the well-definedness proof of H_i that the "boundary-path" (i.e. what is defined by u and by v on $\partial\mathbb{D}$) does never cross a corridor without going through the reading domain of some letter of the associated word sequence. According to the reading conditions of B.1.4 this is given as long as we consider segments in the interior of the upper or of the lower semicircle of $\partial\mathbb{D}$. However, for segments that contain $(-1,0)$ or $(+1,0)$ this property might not be given, since the situation could be as follows: Assume that $u(0) = v(0)$ is some point inside a corridor l , but that the paths emanate to different directions, so that neither the u-reading nor the v-reading acknowledges the corresponding partial act of crossing. Then the transformation process that turns ω into η has to contain some truncation or appendance of the $l^{(-1)}$-letter so that the first entries of the corresponding triples can be adjusted. If this letter should either be present in ω or in η, will see an ordinary peaky corridor in the realization diagram. However, if this letter should be truncated to the end again, then drawing some double-peaky-corridor into our diagram is the only facility that can prevent us from having some act of crossing a corridor by our $\partial\mathbb{D}$-path without having to go over the reading domain of some letter also. For that reason we want the double-peaky corridors in our diagram.

Having modified our cell-subdivision by additionally inserting the double-peaky corridors, we can now proceed with our programme 8.11:
8.13 Definition of the interpolating functions of realization diagrams on the zero-skeleton (cf. 8.11).
The zero-cells of our cell-subdivision are intersection points of corridor-boundaries. By 8.3(i) only boundaries of corridors belonging to commuting variables of different basic directions can intersect. Note that each of the corridors of a realization diagram is associated to some letter of $\overline{\mathcal{A}}_i$, if one recalls that the end segments of such corridors correspond to such reading domains. If we take a closer look we get that in the standard situation even the boundaries of all corridors of the realization diagrams are associated to the boundaries of the $\mathcal{N}_i$-corridors. This is an immediate consequence from recalling the definition of these corridors, and that in the two standard cases (corridors that connect reading domains of $\mathrm{Dom}(u)$ and $\mathrm{Dom}(v)$ with each other or just two reading domains within one of these domains of definition) our picture does not provide a reversion of order. Consequently, the endpoints of reading domains that are connected by a corridor of a realizations diagram, lie on the same boundary of the corresponding $\mathcal{N}_i$-corridor. In the context peaky corridors must be treated as a special case: From the non-peaky end we get according to the standard definition some association to the sides of $\mathcal{N}_i$-corridors as well; but we have to take into account that the u- or v-value at the peaky end is that of the start- or endpoint of u and of v, and hence could be on be on any of the sides or in the interior of the $\mathcal{N}_i$-corridor. We now proceed by defining of our function H_i: On all corridor boundaries of a realization diagram that are associated to a horizontal letter of $\mathcal{A}_i$ we define the y-coordinates of H as a linear interpolation of those y-coordinates that are already defined on $\partial\mathbb{D}$, and the same for the x-coordinates of H_i and those boundary lines of corridors that are associated to vertical $\mathcal{A}_i$-letters. By the

above arguments this definition gives just a constant function for non-peaky corridors. However it may give a non-constant function on the boundaries of peaky corridors. This partial definition of the components of H_i implies gives the desired definition of H_i for the zero-cells, since here the x- and the y-coordinate of H_i are defined. Note that the corresponding H_i-value is in any case contained in X_i (cf. A.3.4), since the combination of some value obtained by linear interpolation of the y-coordinate of a horizontal corridor and the x-coordinate of a vertical corridor gives some point in the crossing region of those corridors — and according to 8.3(i) only such pairs of $\mathcal{N}_i$-corridors that intersect in a crossing region can have associated realization diagram corridors that intersect each other.

8.14 Definition of the interpolating functions of realization diagrams on the one-skeleton (cf. 8.11):
Recall that for each one-cell of the cell-subdivision of $\mathbb{D}$, either the x-coordinate of H_i or the y-coordinate of H_i has already been defined by 8.13. The other coordinate can now be defined by linearly interpolating the values that are already given at the two endpoints of the corresponding one-cell. Note that the accordingly defined values of H_i must be all contained in X_i, since both H_i-values that have been defined on the zero-cells bounding one-cells, are by construction contained on a boundary line or inside of one of the corridors of $\mathcal{N}_i$, and a direct interpolation of that line cannot go through a hole integrated into our corridor lattice $\mathcal{N}_i$.

The task of also finding a definition of our function H_i for the interiors of the two-cells of the cell-subdivision of $\mathbb{D}$ will be done by proving that the path that has been constructed by completing the definition of H_i on the boundary domain of the corresponding two-cell is nullhomotopic within X_i. This will happen in the following paragraphs 8.15–8.17 by looking at the various types of two-cells separately:
8.15 Definition of the interpolating functions of realization diagrams on the first type of two-cells (cf. 8.11), i.e. on the crossing regions of realization diagrams.
Observe that the H_i-values that have been defined on the realization diagram corridors had all been lying in the associated $\mathcal{N}_i$-corridor. Consequently, all boundary values on the crossing regions of realization diagram corridors are defined inside the crossing regions of the corresponding $\mathcal{N}_i$-corridors, and since that is a convex region without holes, for that type of two-cells the boundary path is nullhomotopic as required. This also holds, when we are considering such crossing regions that are adjacent to $\partial\mathbb{D}$, so that a part of the boundary is then given by the corresponding part of $\partial\mathbb{D}$, and hence the corresponding boundary path is then partially given by the corresponding segment of u or of v. However, since the $\partial\mathbb{D}$-part in such a case had been the intersection of reading domains, the corresponding contribution to the boundary path must also be some path inside the corresponding crossing region of $\mathcal{N}_i$-corridors. Note that corresponding special cases with $\partial\mathbb{D}$-segments of cells will also exist for the other types of two-cells, and since analogous arguments will hold in each of these special cases, we will below in 8.16–8.17 not mention them explicitly any more.

8.16 Definition of the interpolating functions of realization diagrams on the second type of two-cells (cf. 8.11):
We now consider a two-cell which is the segment of a realization diagram corridor that is bounded by other crossing regions or by the corresponding $\partial\mathbb{D}$-segments. In that case we can similarly as in 8.15 conclude that the values given by the boundary path are all contained in the associated $\mathcal{N}_i$-corridor. Observe that they need not all to be contained in the associated segment of the $\mathcal{N}_i$-corridor, but may have gone into the adjacent crossing regions of corridor, but since an $\mathcal{N}_i$-corridor is free of holes integrated into $\mathcal{N}_i$, it is a convex region of X_i in any case. Hence the boundary path, whatever it may look like, is nullhomotopic within $\mathcal{N}_i$.
Note that the above *claim* that the *boundary path of the realization-diagram corridor-segments remains in the corresponding $\mathcal{N}_i$-corridor segment and its adjacent crossing regions,* was not essential in the preceding case 8.15 (and hence we did not care to give the complete argument for that claim); but since it will become vital in the following case 8.17, this claim shall be proven below:
Consider the corresponding realization-diagram corridor and all those corridors that are associated with the other basic direction. The latter corridors cannot intersect each other in our realization diagram and that eases to understand the corresponding picture. Without loss of generality let us assume, that the latter type of corridors is associated to the vertical letters. Hence our $\partial\mathbb{D}$-path remains in certain x-intervals in the end-segment of those corridors, and in the gap-lines in between it is usually in the complementary x-intervals allowing to enter but not to cross the adjacent ones. According to the reading conditions B.1.4 there is not any chance for the x-values taken on by the $\partial\mathbb{D}$-paths to be in other domains then claimed above. In our extension steps 8.13–8.14 we have extended these values along the boundary lines of the realization diagram corridors, usually in a constant way, and for the exceptional peaky corridors in such a way that only a linear interpolation between the boundary of the x-domain of horizontal corridors and one inner point of those corridors could have taken place. This interpolation was obeyed, when the x-values had been defined on our corridor of consideration that is associated with the vertical variable. And hence we see that the boundary path of our corridor segment of consideration is in the standard case restricted to the x-values of a gap-line between two vertical corridors and can at most go into the regions of the adjacent corridor.

8.17 Definition of the interpolating functions of realization diagrams on the third and last type of two-cells (cf. 8.11):
In this case we have to consider those two-cells of our realization diagram that do not fall inside the region of any of the corresponding corridors. Note that this type of regions can have a more complicated geometrical structure than the two types of cells as discussed in 8.15/8.16: Segments and crossing regions of realization-diagram corridors essentially are combinatorially rectangular regions or, at least what is left of such a region if some part is cut of be the boundary of $\partial\mathbb{D}$ (or if some side is collapsed, if a peaky corridor has to be considered). However, such an assertion cannot be made for those types of regions that are outside of any of the realization diagram corridors, since even a system

of non-intersecting corridors in $\mathbb{D}$ can yield ideal n-gons of arbitrarily high n. We should be aware of this effect if we now take a closer look at the corresponding regions:
So let us consider one of those cells that is bounded by $\partial\mathbb{D}$-segments and realization diagram corridors, but is disjoint from any of them. In this case we look at the boundary path and ask whether there is any chance for such a path to cross any (horizontal or vertical) $\mathcal{N}_i$-corridor. One sees, that there is none, because the corresponding $\partial\mathbb{D}$-segments do not contain crossings (otherwise they would be reading domains and hence fall in some realization-diagram-corridor) and in the standard case those boundary parts of our two cells, that are given by realization diagram corridor just contain segments of a path that moving along an $\mathcal{N}_i$-corridor, but do not cross them. In the exceptional situation where we have to deal with the boundary of one of our peaky corridors, as seen before our boundary path may go partially inside some of those corridors, but the combinatorial analysis gives that it cannot cross such a corridor, either (the same, by the way, may happen on the $\partial\mathbb{D}$-segments as well). Hence we conclude the boundary path of such a cell has to stay inside of some extended lattice square of a $\mathcal{N}_i$-corridor. Such an extended lattice square need not be convex, because there can be holes in each of the four corners, but it is at least star-shaped (cf. 8.18). Hence this boundary path is nullhomotopic in X_i as well.

8.18 Addendum to 8.17: Calling extended lattice squares "star-shaped" regions, as done in the preceding paragraph has been a bit negligent, since such regions are only star-shaped if they are considered as open domains. On the other hand we had stressed in C.1.3(iii) that for reading purposes all domains must be considered as closed domains. If accordingly the closure of some extended lattice square is considered and if at least one of the four corner-regions of the lattice square should consist of a hole E, then our lattice square will not be star-shaped; due to the connection as given by the outer part of ∂E it will not been even simply connected. Hence the proof-arguments as given above left us with some gap, and this gap will have to be closed below by giving the reasons why the boundary path as considered above can never pass along the outer part of ∂E:

- ▷ A segment of the $\partial\mathbb{D}$-path cannot do this, since, especially when taking the corresponding acts of crossing on finer levels into account, we see that such a behaviour of a $\partial\mathbb{D}$-path would result in acts of crossing that are not compatible with what could give a segment of the boundary path of a region as above considered in 8.15–8.17.
- ▷ A segment of the boundary path that is given by some ordinary (i.e. non-peaky) corridor can also never be responsible for passing along the outer part of ∂E : The constant behaviour of the corresponding coordinates of an ordinary realization-diagram corridor makes it even impossible that our boundary path enters the region of the adjacent corridor; in situations where that case only occurs, our boundary path will actually stay inside the pure lattice square.
- ▷ Peaky corridors can also never make our boundary path pass along the outside of ∂E. Other than for the ordinary corridors the segments of our boundary path that lie inside a peaky corridor may have left the corresponding lattice square and gone inside the adjacent corridor. But since the corresponding values are still some interpolation of some value taken from the inside value and some other value, the corresponding

segment can never pass along the outside boundary of that corridor for an entire segment.

▷ The preceding argument does not hold for double-peaky corridors. Hence at first sight it might appear as if double peaky-corridors actually could give a boundary-path segment on the outside boundary of ∂E. Since the fate of being appended and later truncated to the other end of a word might happen to a vertical and to a horizontal letter we might have up to two double peaky-corridors in our picture (due to the optimization-assumption of 8.3/7.17 such truncations and appendances cannot affect more letters or be undone later). However, since double-peaky corridors are always spanned between $(-1,0)$ and $(+1,0)$, they can never intersect in a realization diagram. However, to pass along both segments of the outside of ∂E it would take two intersecting double-peaky corridors in our realization diagram. Hence these types of boundary paths can also not cause a boundary path that passes along the outside of ∂D.

Hence we get that our boundary path can never pass along the outside part of any hole such as E. Having seen this, we know that it remains inside a star-shaped region of the extended lattice square, so that the arguments of the main part of the well-definedness proof of Definition 8.17 in 8.17 then suffice to give the desired result.

8.19 Remark: Having seen in 8.15–8.18 that we got nullhomotopic paths for any of the types of two-cells in our realization diagram, we hence managed to extend H_i to the entire domain of $\mathbb{D}$. Such a continuous function can be regarded as a nullhomotopy for the $\partial\mathbb{D}$-path within X_i. The following paragraph 8.20 will then prove the existence of a nullhomotopy even within X by constructing it as a limit function of the H_i.

As a final preparatory for step for Step (iii) of our proof-programme 1.21 we note that the *H_i-functions as defined by now permit the following estimates* of their values:

Note that according to the construction of 8.13–8.14 the boundary lines of realization diagram corridors that are associated with vertical/horizontal letters have been used as interpolation lines of the x- and y-coordinates of our path u and v. Since in the course of Steps 8.15–8.17 of that construction only homotopies in convex or star-shaped domains have been used, these homotopies need only to take on values inside the corresponding regions. Hence the x- or y-values as defined on the boundary lines of corridors do also provide estimates for the corresponding x- and y-values of H_i in the adjacent 2-cells.

8.20 Lemma: *Any sequence of interpolating functions for the same realization diagram uniformly converges. The limit function takes on only values in X.*

The Proof follows from in principle applying the same arguments as in C.3.1 (when approximating paths realizing single words only have been used to construct paths realizing word sequences) to the information provided by 8.3 and 8.19. In the following we are going to consider the x- and y-coordinates of our H_i-functions separately. Hence, since we are considering only one of our coordinates, we can forget about the realization diagram corridors that are associated with the other basic directions, and hence we do not have intersections of our realization diagram corridors in our picture any more. The only remaining intersection phenomena that have to be dealt with accordingly are those

of corridors of the same basic directions of adjacent indices. By 8.3(ii) even the variety of intersection phenomena of such intersections is bounded to a single phenomenon: The realization diagram corridor that is associated with a split-letter may turn away from the domain bounded by its root-corridor (so that in the course of this turning-away process also two boundary lines have to intersect each other). Hence we can ask: "If some point P lies with respect to the level index i in a certain region that is bounded by some of the corridor boundaries, in which region with respect to the level index $i+1$ could it lie then?" – Since we have got rid of the phenomenon of intersecting realization diagram corridors, there are essentially just two case which have to be considered separately, namely that P could lie inside a realization-diagram corridor C, or outside of all of them.

- ▷ In the "inside"-case we can quickly argue as follows: Recall that by 8.19 the boundaries of the realization diagram corridors give estimates on the corresponding (x- or y-coordinate) of H_i. In principle this estimate also holds for the outer boundaries of the corresponding split-up corridors. Since by 8.7 these split-up corridors could have only turned away to the outside, the corresponding regions for which this estimate holds could only have become bigger and hence the estimate holds the same for $H_{i+1}(P)$ as for $H_i(P)$. Thus in any case the region of C of the level index i is still inside the region bounded is by the corresponding boundaries of the realization-diagram corridors that are associated with its split-up letters.
- ▷ The situation is slightly more complicated for a point of $\mathbb{D}$ which does not lie in a realization diagram corridor so that the corresponding coordinate of x and y is associated with a gap-line (cf. B.1.2(ii)):

As stressed in 8.8, the corresponding region may be thought of as a kind of ideal polygon. The fact that the inner split-up corridors of the bounding realization-diagram corridors may have turned away and hence run through the interior of our polygon makes it impossible to use the same arguments as before. On the other hand, the outer boundaries of the corresponding split-up corridors may only have turned away to the outside, and hence there is not any chance that these boundary lines run through our polygon. Hence the corresponding estimates for our $H_i(P)$-value that can be provided by the outer boundary lines of the bounding corridors still holds for $H_{i+1}(P)$. This implies that the corresponding coordinate of $H_i(P)$ may not be in the same lattice square, but it may have gone to the adjacent corridors, and hence it still is in the extended lattice square. This effect just gives a factor "3" in the corresponding estimates and an additional factor "2" comes from superposing the estimates for the x- and for the y-coordinate (we have seen similar effects in C.3.1), and such an additional factor $2 \cdot 3 = 6$ does not overturn that $(\|H_{i+1}(P) - H_i(P)\|)_{i\in\mathbb{N}}$ can be majorized by a geometric regression, similarly as in C.3.1). Hence the H_i are a sequence of uniformly convergent functions whose limit, with the same arguments as in C.3.6, is defined in X, and hence prove that the $\partial\mathbb{D}$-path is nullhomotopic. □

§9. Combinatorial characterization of nullhomotopy

9.1 Theorem: *If and only if ω is as in Corollary 7.2, then ω describes a nullhomotopic closed path.*

Proof:

"$\Longrightarrow$" is an immediate corollary of its predecessors in Sections §7 and §8, in particular of first having constructed a compatible reduction scheme (cf. the proof of 7.1) and then having turned it via realization diagrams into a nullhomotopy (cf. 1.21/8.20);

"$\Longleftarrow$" has to be proved by using that by assumption we are told that for a particular level index i the corresponding word ω_i cannot be reduced to the empty word in the 3.9-sense. Then it has to be proved that, when applying a simpler concept of word sequences by using a slice-concept of corridors instead of the lattice-concept (cf. 9.2), such a system also gives a reading for our path that cannot be reduced to the empty word, i.e. is *"non-exhaustible"*. The paragraphs 9.3–9.10 below will be devoted to give the corresponding arguments. However, before we enter this discussion, we remark that after having obtained such a reading from a slice-system of corridors, we will be done: Since the finitely punctured plane X_i has a finitely generated free fundamental group, and since reading words from a slice-system corresponds to obtaining a description of the homotopy class of a path in a concrete free basis of such a group, finding a description that for a concrete path cannot be reduced to the empty one implies that the path cannot be null-homotopic, even not in the space X_i which has fewer holes then X.

Hence we are left with making more precise with defining the slice-system of our corridors and with showing that "non-exhaustible for the full reading at our i^{th} corridor lattices" implies "non-exhaustible for the thinned slice reading". The subsequent items 9.2–9.10 are devoted to this task and will together imply Theorem 9.1.

9.2 Definition of a slice-system of corridors:

We will call a system of corridors "slice-picture", if it is not constructed according to rules B.1.1 (according to which our network $\mathcal{N}$ has been constructed), but according to the following rules:

(i) Each hole is precisely associated with one corridor, which emanates from this hole.

(ii) All corridors are parallel and go into the same direction.

(iii) The one of the two end segments of a corridor which is not determined by (i) lies either at some other hole or it lies at the boundary of the exterior region.

(iv) Holes may be contained inside the corridors of other holes and hence corridors may lie inside other corridors. However, in analogy to B.1.2(v) we forbid partially overlapping corridors or holes which only partially lie in some corridors. This guarantees that in (iii) all our corridors find some hole-boundary as an end-segment.

Fig.n gives some impression for how such a system of corridors might look like. Observe that in a plane with a finite number of holes words which report the crossings with a slice-system of corridors in principle give the classical way for how to describe the free fundamental group of this space in a concrete free basis. For that reason, even though usually in this paper we are not just working with a slice system but with an entire

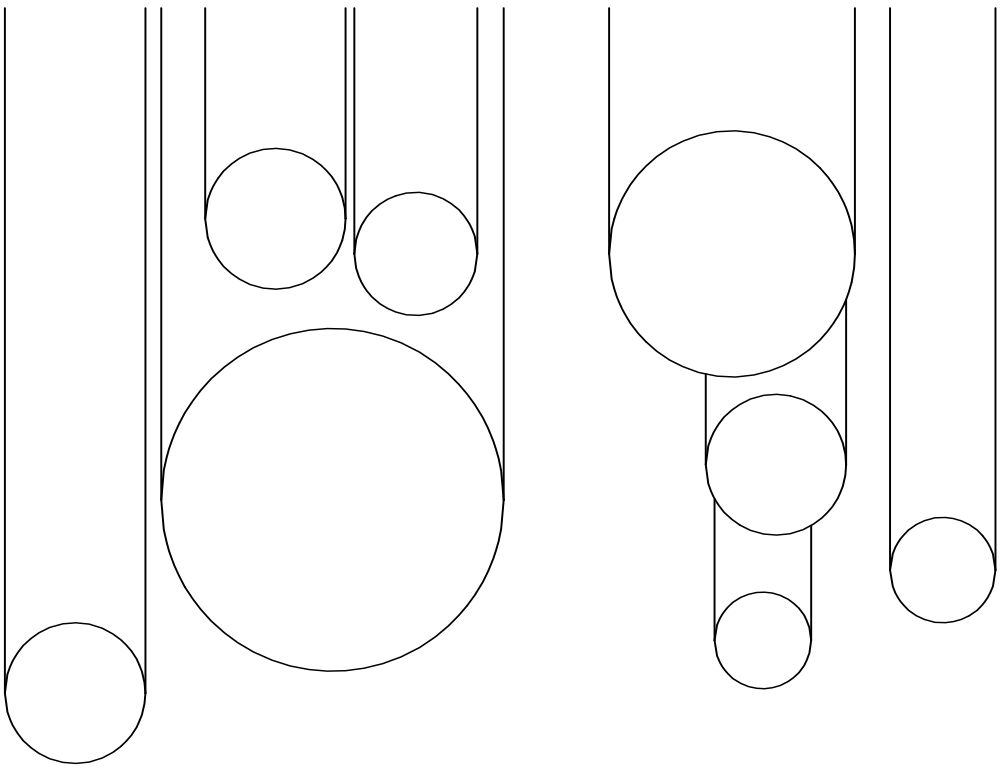


Fig.n (belonging to 9.2):
This figure pictures the probably easiest concept of holes brought into a nice shape and corridors attached to it: a "slice system of corridors". Each hole has been given a round shape and has been connected to infinity by precisely one corridor. Even if this concept of placing corridors has in general not been used in this paper and "corridor networks" have been constructed instead (cf. B.1.1(Fig.g)), sometimes as an intermediate state between the classical combinatorial description of the fundamental group of punctured planes and corridor networks such slice systems need also be considered (e.g. in the proofs of 9.1, 11.6 and 11.9).

network of corridors, restricting our corridor network to a slice-system will in some situations be a helpful tool.
This might be a suitable place to rise the question what made us abandon the classical concept of slices and skip to building up a network of corridors in this paper (as described in B.1.1/A.3.6–A.3.12). Well, for a slice-system of corridors as in Fig.n any finite word can be realized by a path. But if, similar as for our word sequences, we think of a situation where more and more holes are cropping up and we are given a sequence of words that continues to report additional acts of crossing the newly cropping up corridors, we may easily find ourselves with a list of demands that all together cannot be realized by a continuous path anymore. In A.4.5(Fig.b) a similar situation has already been described. I.e. for a concept of word sequences that would be based on slice-systems of corridors we would have to differentiate between "realizable word sequences" and "non-realizable" ones, and there is little hope that for arbitrary patterns of holes

combinatorial rules can be found that tell is which word sequences are realizable (i.e. actually correspond to a continuous path), other than we could do for our concept in C.1.5/C.3.1. The almost one-to-one correspondence between word sequences and paths that was established in C.2.1/C.3.1/C.2.6/C.4.1 is the backbone of our proof-strategy, and one of the steps that helped establish this correspondence could be performed in C.3.3/C.3.5(1)/C.3.17(1): Since a word sequence is read from horizontal and vertical corridors and hence contains information on the x- and the y-coordinates of paths, the information that is contained in word sequences could in C.3.3/C.3.9 be turned into the information that a certain sequence of paths does uniformly converge and that thus there has to exist some limit path. Such a strategy could have hardly been made work with word sequences that were merely to read from a slice-system of corridors.

9.3 Definition of the slice system of corridors as it is set up for the proof of Theorem 9.1:
Given a fixed level index i so that the ith word of our word sequence ω_i cannot be cancelled up to the empty word. We then in particular look at the finitely punctured plane X_i (cf. A.3.4), and for many purposes we will associate our "non-exhaustible" word ω_i with the corresponding edge-path in the dual lattice graph as described in C.1.6. Let ρ be this edge-path. Since our assumptions do not provide a picture for how the associated path of ω actually looks like, we will consider the behaviour of our path ρ instead.
In the following we will associate to each corridor of X_i either the attribute "essential" or "non-essential". We regard all horizontal corridors as non-essential, and in addition some of the vertical ones also. The essential vertical corridors are selected as follows: For each hole (apart from the exterior region) one of the emanating upward directed corridors is considered to be essential, all the other corridors that emanate in that direction are inessential. Recall that the corridors that emanate to the sides have been regarded as inessential, either; however, whether downleading corridors are essential or inessential can only be predicted by studying the pattern of holes that lie below the one of consideration. Finally, all corridors that emanate from the bottom bar are inessential. If the index of a hole is i, then the upward directed corridor that emanates from that hole must be essential, but for any other "fatter" hole there is actually a freedom of choice in which of the corridors should be declared to be the essential one.
However, it will be also necessary to consider the entire area that lies over some fat corridor. This area will be called the "shadow of the hole". If k is the index of the corresponding hole, then its shadow is just the area of the corresponding $\mathcal{N}_k$-corridor that is emanating upward from the hole (and hence is much thicker as the $\mathcal{N}_i$-corridors that are usually considered in this proof). According to the construction-principles of our corridor network, shadows of different holes hence can just intersect that way that some shadow is contained in the other. Hence shadows are partially ordered by inclusion, and we can talk of "innermost shadows". In that context note that the finest holes of consideration (i.e. those of index i) are not associated with a shadow, only those holes are that are at least one step fatter.

The above construction gives that the essential corridors give a slice-system as described in 9.2.

9.4 Reduction of the proof of Theorem 9.1 to a planar graph-theoretical problem:
When we ignore in our word ω_i all letters that are not associated with essential corridors, we get a word that describes the movements of our path according to the classical concept of determining the homotopy class of a path in a finitely punctured plane by associating it to a concrete word in a concrete free basis of the corresponding fundamental group. In the following we will definitely look at these "thinned" words that arise from ignoring all acts of crossing with inessential corridors; if η_i denotes a word that is according to B.1.4 be read from some path, then

(1) $\overline{\eta_i}$ denotes the corresponding thinned word.

Recall that for this proof we had to consider some concrete non-exhaustible word ω_i, and the crucial question now is, whether this implies that $\overline{\omega_i}$ is or is not exhaustible. In that context note that all processes of B.2.2(i)–B.2.2(v), if we apply them to ω_i and ask in how far they do affect $\overline{\omega_i}$, apart from truncations, just induce either the identity or some classical cancellation of adjacent inverse variables. Truncations, however, can be neglected, since by 2.4/1.1(iii)/2.17 they do never give more than a readjusting of our reading. In addition without loss if generality, one can assume that ω_i is adjusted in such a way that the first entry and the last entry of the corresponding triple are identical (recall that ω_i was assumed to describe a closed path), and if that is assumed, then the exhaustibility of a word in the free and the fixed equivalence relation coincide (recall that truncatable letters do never obstruct other operations, since they are placed at the ends of our words, and if it should be possible to cancel all letters except the truncatable ones down to the empty word, then C.1.6 implies that the truncatable letters have to appear in such an order, that they could also be removed by cancellations). That way we get rid of truncations by skipping to a fixed equivalence relation, and hence we may now assume that any legitimate process that could be applied to ω_i can only affect the induced thinned word by classical cancellations and insertions, i.e. by such processes that are also considered as legitimate equivalence transformation if the thinned word is used to represent the corresponding homotopy class in $\pi_1(X_i)$. The proof of this direction 9.1("$\Leftarrow$") will be essentially based on showing that our non-exhaustible ω_i induces in that sense a non-exhaustible thinned word, which according to the classical calculus of a free group must describe a non-trivial homotopy class in $\pi_1(X_i)$. Since $X \subset X_i$, this gives that the associated path of ω must be non-nullhomotopic in X, also. Hence just one gap need to be closed in this proof, namely it has to be shown that "non-exhaustible" is a property that is passed on from words to thinned words. We will show this fact below in Propositions 9.5–9.10 by proving the contraposition. As already indicated by the introductory remarks of this proof-direction, we will not perform the corresponding conclusions on an algebraic level, but we will think in the geometrical terms of an edge-path ρ that moves around in the dual graph of the corresponding lattice. By analyzing

how the complete and the thinned readings of such a path are related to each other, we will be able to fill in the remaining gap in the proof of 9.1("$\Leftarrow$"), namely

9.5 Proposition: *Given a corridor lattice $\mathcal{N}_i$ that satisfies all conditions of B.1.1. A subsystem of corridors that gives slice-system as in 9.3 is marked as essential. Let ρ be a path that behaves like a dual edge path to the $\mathcal{N}_i$-lattice (cf. C.1.6) and let ω_i be the full reading from ρ in the sense of B.1.4 while $\overline{\omega_i}$ is the thinned reading that only reports the crossings of the essential corridors. Then, if $\overline{\omega_i}$ is exhaustible, ω_i is exhaustible (cf. 9.1("$\Leftarrow$")).*

Proof: Unfortunately, even though this is just a supplementary proposition to 9.1("$\Leftarrow$"), the proof is lengthy, and is hence split up over the subsequent discussions in 9.5–9.10. Hence this paragraph is only devoted to explain the basic set-ups and main assumptions:
By construction (cf. 9.4) the exhaustibility of a thinned word $\overline{\omega}$ is merely due to iterated cancellations. In the following we want to describe how, based on this assumption, the edge-path can be homotoped around in X_i till it is the empty path. Since the corresponding terminology of homotopy is a combinatorial one and each of the used elementary processes corresponds by 1.1/3.9 to some valid transformation of ω_i, this will yield the desired result. Observe that, whenever ρ has a thorn, it can be simplified by the corresponding cancelling process. Since this process can never be obstructed by a hole, we will always immediately perform such thorn-cancellations, whenever some other process should have generated such a thorn. Having made this observation enables us, without loss of generality, to restrict the discussion in the remaining items 9.6–9.10 to thorn-free edge-paths ρ.

9.6 Proof of Proposition 9.5 in case of a non-empty thinned reading $\overline{\omega_i}$ in the first special case:
If $\overline{\omega_i}$ is exhaustible and not (yet) empty, there has to exist an innermost cancelling pair "ll^{-1}". We first look at the corresponding segment of ρ that is bounded by the acts of crossing the corridor that are associated to our innermost cancelling pair. If the corresponding corridors lie inside the shadows of other corridors we ask whether our path manages to leave each of these shadows before is has to reenter them again in the final part of its segment of consideration. *This item is only devoted to discuss the case where our path does manage to leave all these shadows and in the corresponding "middle part" of the segment of consideration it again passes through these or through other shadows.* Now we look at the innermost shadow through which our segment of ρ passes. Since the upper and the lower boundary of each shadow are on some hole, our path has to enter and to leave the shadow on one of its sides. In addition, it has to be the same side, since otherwise our path would also have crossed the essential corridor that belongs the same hole as the corresponding shadow, and this would yield a contradiction since we are discussing some segment of our path which lies between two neighbouring letters and hence must not cross essential corridors. Hence some subsegment of our path has to exist which looks like "$vhh\ldots hhv^{-1}$", where v stands for a crossing of an inessential vertical corridor, v^{-1} is the act of crossing the same line of corridors in the inverse

direction, and all "h"-letters stand for crossing horizontal corridors. Since without loss of generality (cf. the proof of 9.5) we may assume that our path is free of thorns, all h have the same exponent. That way a "$vhh\dots hv^{-1}$"-segment just reflects the fact that within such a shadow (depending on whether the entrance and exit is from the left or from the right) our path must take on a maximal or minimal x-coordinate, and hence it must avoid thorns while doing this. We now hope that a far-distance cancellation can be performed between v and v^{-1}. It can be precisely be performed, if in our dual graph the area between the v-edge and the v^{-1}-edge is free of $\mathcal{N}_i$-holes. But if there actually were holes, then either the v-edge or the v^{-1}-edge would either cross the associated essential corridor of one of these holes, or at least would enter/exit a corresponding shadow. Both assumptions would give a contradiction: we already saw the argument why we cannot have crossings of essential corridors in this segment, and any shadow that our path would see that way would lie in the interior of the shadow that contains the entire segment "$vh\dots hv^{-1}$" and has been assumed to be an innermost one already. That way our path can be simplified by reducing its number of edges. This is a basic observation that allows us to iterate this conclusion, however we have to argue that we can find similar reducing processes, if the assumptions of this case are not (all) given. This is the purpose of the following items 9.7–9.9:

9.7 Getting rid of some of the assumptions of the special case 9.6:
Now assume that ρ in its segment bounded by the l- and the l^{-1}-crossings manages to exit from all shadows that contain the corridor l, but does not enter any shadows in the corresponding middle part (that is defined as in 9.6). It is not hard to see, that we can play in principle the same arguments as before, as long as we do not interchange the role of "v" and "h", since a segment such as "$h^{-1}v\dots vh$" might well run around a hole without getting a contradiction by having entered a shadow or performed an essential crossing. However, without such a reversion the arguments of 9.6 can still be applied: The exit and re-entrance from the outermost shadow that contains l (it has to be on the same side of this corridor, if no essential crossings are to be performed) can be used in a similar way as the entering ad re-exit from an innermost shadow: It is some vertical line to which our path returns, and, unless our path is entirely vertical (the next item will be devoted to that case), there has to be some maximum or minimum of its x-coordinate (other than those at the endpoint of our segment) and here we see some configuration "$vh\dots hv^{-1}$", to which the same arguments as in 9.6 can be applied.

9.8 Getting rid of even more assumptions of the special case 9.6:
As mentioned in the last sentence of 9.7, we also have to consider the situation that the crucial segment of our path might be entirely horizontal, i.e. that we only see "$h\dots h$". However, if we add the corresponding segments that give the exit and the re-entrance to the outermost shadow that contains l, we again have "$vh\dots hv^{-1}$". We use λ to denote the boundary line of the shadow that is intersected be these ρ-segments "v" and "v^{-1}". It plays an essential role in the following because we know that the segment "$vh\dots hv^{-1}$" does only contain the two acts of crossing λ that correspond to v and v^{-1}, and in addition we know that our path on its way from l to v and on its way from v^{-1} back to l does not cross λ anymore, since otherwise it would have left or re-entered the

outermost shadow that surrounds l earlier or later than assumed, respectively. Knowing this about our segment of ρ makes us conclude that, if v would lie in a different set of essential corridors or shadows than v^{-1}, then the path-segment of consideration would encircle some hole, and then some crossing of an additional essential corridor could not be avoided. This would give the same contradiction as seen in 9.6. Hence we conclude that the area between the v- edge and the v^{-1}-edge must not contain a hole. Hence this case also does not provide an obstruction for pulling v into a shorter position by cancelling v against v^{-1}.

9.9 Completion of the proof of Proposition 9.5 in the case of a non-empty word $\overline{\omega_i}$:
By 9.6–9.8 we are only left with the situation that our path ρ between its l-crossing and between the l^{-1}-crossing should not manage to exit from all surrounding shadows. However, essentially the same arguments as above may be applied: In that case we just ignore all those shadows from which the corresponding segment of ρ does not exit, i.e. we treat the outermost shadow for which we still see an exit as "the outermost" one. Then apart from the single exceptional case as discussed in the following sentences, the same line of arguments as above in 9.7–9.9 can be applied. The one exceptional case which deserves special attention is the one, where, similarly as in 9.8, we only see a segment "$vh \dots hv^{-1}$", but where v and v^{-1} are not an exit and the re-entrance of some shadow that surrounds l, but from l itself. In that case the combinatorial information that l^{-1} is the inverse letter with respect to l provides the essential information that even in this case the corresponding shortening process is not obstructed by a hole. In that case, other than in the one discussed before, the shortening process affects the thinned word and the cancelling pair "ll^{-1}" disappears from the it. Hence it has become visible that by iteratively applying our shortening processes in 9.5–9.9 together, our path ρ may be homotoped into such a position where its associated thinned word is empty. □

9.10 Proof of Proposition 9.5 in case of an empty thinned reading $\overline{\omega_i}$:
By this step the proof of 9.5 will be completed. Having successfully handled the case of non-empty $\overline{\omega_i}$ in 9.5–9.9, all it takes now is merely noting that an edge-path with an empty thinned reading can be simplified by the same processes that have been used in 9.5–9.9 as above: Other than in the cases before the entire ρ has to be considered, not just some subsegment of it. However, since ρ is a closed path we can play the same arguments as before: instead of knowing that the relevant segment starts and ends on the same boundary line, in this case we can even use the stronger assumption that it actually starts and ends at the same point. This makes us again find reducing processes, and these processes can only stop if our edge-path has become empty. This implies that even the complete reading ω has become empty, i.e. that ω_i has been exhaustible. Hence Prop.9.5 is proved by now, and with it Theorem 9.1 follows from 9.4. □

Having proven 9.1 it might appear that as if it would not take long any more before we are ready to prove, as several times mentioned already (cf. B.2.0(i)–(ii)/B.2.10), that the weak equivalence relation precisely corresponds to the relative homotopy of paths. However, one crucial ingredient is still missing: In order to perform the combinatorial

equivalent for reducing ordinary homotopies to nullhomotopies we need to analyze what type of operations one could see if a product of word sequences $\omega \cdot \omega'$ is to be transformed into a reduced word sequence, where ω and ω' have already been reduced. This will have to be discussed before in Section §11. However, before we can enter that discussion, another topic has to be raised in the subsequent section:

§10. Combinatorilly describing corridor entrances and exits

It is the purpose of the first remark of this section to explain why the concepts as developed so far do not yet give the desired combinatorial description:

10.1 Motivating Remark: Recall that in the proof of C.4.1 already the problem arose, to find a suitable concept for describing the start- and endpoint of reading domain by combinatorial objects that should accordingly be related to the associated word sequence. In C.5.3–C.5.4 we already introduced the corresponding combinatorial concept, namely the "position sequences". However, we postponed characterizing those position sequences, that describe start- or endpoints of our reading domains, since it is to some extent delicate, and it requires a dual approach: For obtaining the corresponding combinatorial characterization of the relevant position sequences (they have to exist by C.5.18(iii)), we firstly define them by an arbitrary definition as "corridor entrance- and exit points" (10.5) and later in 10.8 we will prove that these combinatorially defined position sequences actually describe the start and end-points of the corresponding reading domains, at least up to to an unimportant uncertainty that had already been visible in C.2.7 and in the course of the proof of C.4.1(ii). Up to the moment we could do well without that dual approach, but we will not in the remainder of this paper. Further, in the remainder of this paper the phrases "corridor entrance- and exit points" will be exclusively used for what is combinatorially defined below, up to the moment we had sometimes, when it was clear that the geometrically defined start- and endpoints of reading domains where referred to, used this phrase as a synonym for the latter. It is the purpose of Lemma 10.4 to prepare the definition of these corridor exit- and entrance points; before that we take the opportunity to mention some other point of interest that fits into that context:

10.2 Remark: Recall that in C.5.17–C.5.18 we managed to find for each parameter value of a path and each associated word sequence at least one position sequence (cf. C.5.3–C.5.4) that gave a corresponding combinatorial description of the parameter value. This result might seem natural with respect to the what the corresponding concepts had been invented for, nevertheless we should regard it is as a delicate result. For if we just had asked for a tiny bit more, instead of a proof we would have only got a counter-example, namely the one presented below:

10.3 Example: Assume that the $(i+1)^{\text{st}}$ word of a word sequence contains the letters "$\ldots a_1 b_1 b_2 a_2 \ldots$" where a_j and b_j are commuting split-up letters of two commuting letters A and B. Further assume that our path does not go through crossing regions of corridors on the $(i+1)^{\text{st}}$ level, but, as it is clear from those words, it goes through

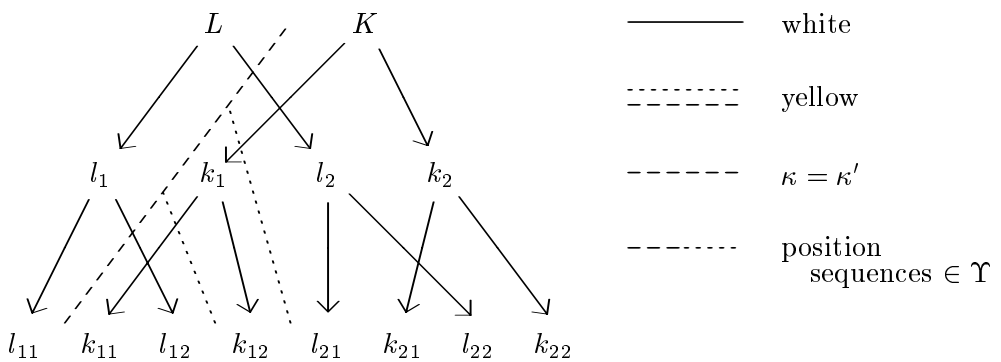


Fig.o (belonging to Remark 10.4(1.)):
Essentially this figure shows a subgraph of a dual split-up picture in a some case, that is of interest in one of the non-standard cases of Lemma 10.4, where κ and κ' coincide. For this drawings the adjustment of variables in the definitions of κ and of κ' is so that $\kappa = \sup\{\lambda \in \mathbb{P}(\omega) \mid \lambda_{i+m}$ is to left of all iterated left split-up-letters of K for almost all $m\}$, and $\kappa' = \inf(\Upsilon)$, where $\Upsilon = \{\lambda \in \mathbb{P}(\omega) \mid \lambda_{i+m}$ is to the right of all iterated left split-up-letters of K for almost all $m\}$.

the crossing region of A and of B. Hence the ith word of our word sequence could read "$\ldots BA \ldots$". If now we look for a position sequence that describes such a segmentation of our path that the a_1- and b_1-crossings fully belong to the first and to the second segment respectively, by C.5.18(iii) such a position sequence exists: On the $(i+1)^{\text{st}}$ level the split-up position is uniquely determined by our demands, and since it would contradict to C.1.5 to have both b_j-letters in the second half and B in the first one, we have on the ith level to define the split-up position before B. But this implies that we are *not* given the choice whether to put A into the first or into the second half, even if the segmentation point lies inside the corridor A. I.e. asking for a position sequence κ that describes such a split-up at a given parameter value where the associated path just crosses some corridor A and makes sure that corresponding letter A is on a particular side of κ might already be to much to ask for. By using the fact that according to C.2.4(3) the use of the freedom C.1.3(ii) can always be extended to higher levels and combining this fact with our key results C.2.1/C.3.1 and 2.17, it follows that a strongly equivalent word sequence with such a κ always exits, but this skipping to a strongly equivalent word sequence might actually be necessary.

10.4 Lemma: *Let L_0 be the jth letter of the ith word of ω. Let $\rho = (\rho_m)_{m\in\mathbb{N}}$ an infinite sequence whose entries are either "l" or "r". Let L_m be the $(\nu_i^{\rho_1} \circ \nu_{i+1}^{\rho_2} \circ \nu_{i+2}^{\rho_3} \circ \ldots \ldots \circ \nu_{i+m-1}^{\rho_m})(j)^{\text{th}}$ letter of ω_{i+m} so that $(L_0, L_1, L_2, \ldots)$ give a series of iterated split-up letters. Let*

(1) $\kappa := (k_n)_{n\in\mathbb{N}} := \sup\{\lambda \in \mathbb{P}(\omega) \mid \lambda_{i+m}$ *is to the left of* L_m *for almost all* $m \in \mathbb{N}\}$

and

(2) $\kappa' := (k'_n)_{n\in\mathbb{N}} := \inf\{\lambda \in \mathbb{P}(\omega) \mid \lambda_{i+m}$ *is to the right of* L_m *for almost all* $m \in \mathbb{N}\}$,

where the infima and suprema of position sequences have been introduced in C.5.16. Then

(i) $[\kappa] \leq [\kappa']$, *if* $[\cdot]$ *is used to denote the equivalence classes as introduced in C.5.15(i)/ C.5.18(iii)*

(ii) If $[\kappa] < [\kappa']$, *then* $\omega_{|[\kappa,\kappa']}$ *is a word sequence that can at most contain the letters of the above series* $(L_0, L_1, L_2, \ldots)$ *as letters of the corresponding basic direction and must contain infinitely many letters of the other basic direction.*

(iii) If almost all entries of ρ *are identical, then Case (ii) cannot be given.*

Remarks:

(1.) Note that the assumptions of this lemma do not imply that $\omega_{|[\kappa,\kappa']}$ contains any of the letters $L_0, L_1, L_2, \ldots$. Fig.o shows a picture of a split-up graph where $\kappa = \kappa'$ so that $\omega_{|[\kappa,\kappa']}$ is empty and hence all the letters L_i are contained outside.

(2.) This lemma gives the essential information for defining what is the goal of this section, namely a combinatorial characterization of corridor exit and -entrance points. Hence we postpone the proof of this lemma to 10.6 for presenting the corresponding definition below:

10.5 Definition of corridor exit and entrance points:

Recall that by B.1.4 a letter had to be put into a word if the associated path crosses a corridor. Further recall that in some situations there was more than one possibility how to read word sequence from a path, in particular if the path is tangent to the boundary of a corridor at some point. In case of such tangencies we might have been given the choice which of the points precisely should be regarded as the one where our path has started or finished to cross a corridor or whether to regard a certain behaviour of our path as an act of crossing at all. 10.4 now gives us the opportunity *to associate with each non-truncatable letter* L *of a word sequence* ω *an "entrance point" and an "exit point" of the corresponding corridor as follows:*

(i) Let $(L{=}L_0, L_1, L_2, \ldots)$ be the iterated left split letters of L. By putting this sequence of letters into Formulae 10.4(1) and 10.4(2) construct two position sequences $\in \mathbb{P}(\omega)$. Now 10.4(iii) gives that these position sequences are equivalent, hence by C.5.18(iii) they describe the same parameter value of the parametrization of the path associated with ω. This is the entrance point associated with L.

(ii) The exit point of L is analogously obtained by considering the sequence of iterated right split-up letters of L.

Recall that the above definition made use of Lemma 10.4 whose proof shall be approached next:

10.6 Proof of Lemma 10.4:

Ad 10.4(i): Assume the contrary, i.e. that $[\kappa] > [\kappa']$. By C.5.15/C.5.18(iii) this implies that there exists an index m such that $k_{m+n} > k'_{m+n} + 5$ for all $n \in \mathbb{N}$. Recall that by C.5.16 our sup- and inf-position sequence κ and κ' need not to be contained in the corresponding set of maximized or minimized position sequences, respectively, but according to the way in which sup and inf are constructed we can assume that there exist position sequences $\kappa_0 \in \{\lambda \in \mathbb{P}(\omega) \mid \lambda_{i+m}$ is to the left of L_m for almost all $m \in \mathbb{N}\}$

and $\kappa_0' \in \{\lambda \in \mathbb{P}(\omega) \mid \lambda_{i+m}$ is to the right of L_m for almost all $m \in \mathbb{N}\}$ that coincide with κ and κ' for the index m, respectively. Since $k_m > k_m' + 5$ for these sequences also, by using the argument of C.5.14 a similar inequality has to hold for the entries of those sequences for all bigger indices as well, and this contradicts to the left/right-relation that the k_0 and k_0'-places should satisfy by definition.

Ad 10.4(ii): Since $[\kappa] \neq [\kappa']$, the word sequence $\omega_{\restriction[\kappa,\kappa']}$ describes a non-constant path and has to be an unbounded word sequence by C.5.15(i)/C.4.8. Assume that $\omega_{\restriction[\kappa,\kappa']}$ would on the $(i+m)^{\text{th}}$ level contain some letter L_m' with $BD(L_m') = BD(L_m)$. We assume that L_m' is a different letter from L_m, and, without loss of generality, that L_m is within ω placed to the left of L_m'. Recall (10.3) that L_m may be found inside or outside of $\omega_{\restriction[\kappa,\kappa']}$. Let k''_{i+m} be the position number to the left of L_m'. By C.5.5/C.5.6 k''_{i+m} can be extended to a finite sequence $(k_1'', k_2'', k_3'', \ldots, k''_{i+m})$ satisfying C.5.4(1) and that $k_n'' \leq k_n$ for all those indices n. By C.5.4(ii) we can extend $(k_1'', k_2'', k_3'', \ldots, k''_{i+m})$ to a complete position sequence $\kappa'' = (k_n'')_{n\in\mathbb{N}}$ such that k_n'' stands for all $n \geq i+m$ to the right of the corresponding iterated right split-up letter of L_m and to the left of the corresponding iterated left split-up letter of L_m'. This makes sure that κ'' satisfies all the minimality properties of κ' despite being a smaller position sequence. Hence we can find no such letter like L_m'. This implies that, since on the other hand $\omega_{\restriction[\kappa,\kappa']}$ is unbounded and hence has to contain an increasing number of letters on the corresponding levels, the letters other than those from the sequence $(L_1, L_2, L_3, \ldots)$ have to have a basic direction different from that of L_m.

Ad 10.4(iii): We take a more careful look at the situation of Case 10.4(ii): Let $i+m$ be a level where $\omega_{\restriction[\kappa,\kappa']}$ contains at least five letters of the opposite basic direction as L_0, so that at least the middle three of them have to contain their full set of split-up letters within $\omega_{\restriction[\kappa,\kappa']}$. Denote these three letters by L', L'', L''', and let k''_{i+m} and k'''_{i+m} be the place numbers (cf. C.5.2) to the left and to the right of L'', respectively. As in the proof of 10.4(ii), these position numbers can be extended to word sequences κ'' and κ''' such that $\kappa < \kappa'' < \kappa''' < \kappa'$ and that $\omega_{\restriction[0^{\mathbb{N}},\kappa'']}$, $\omega_{\restriction[\kappa'',\kappa''']}$ and $\omega_{\restriction[\kappa''',\ell^{\mathbb{N}}]}$ contain at least the full set of split-up letters of L', L'' and L''' respectively. We conclude that infinitely many letters of our series $(L_0, L_1, L_2, \ldots)$ have to be contained in $\omega_{\restriction[0^{\mathbb{N}},\kappa'']}$, since otherwise κ'' would give a bigger position sequence than κ that satisfies the same maximality property as κ. From similar arguments using κ', κ''' and the corresponding minimality property, we get that on the other hand infinitely many letters of $\{L_0, L_1, L_2, \ldots\}$ have to be contained in $\omega_{\restriction[\kappa''',\ell^{\mathbb{N}}]}$ nevertheless. Now think of the diagram as constructed in C.5.5 and trace the monotonous edge path of white arrows that connects L_0 and the series of split-up letters $L_1, L_2, L_3, \ldots$. When translating the above condition into geometry, we get that this edge path has to be intersected from the left to the right infinitely many times by white arrows coming from the iterated split-up letters of L'', and infinitely many such intersections from the right to the left can be found, also. Now only such white arrows pointing at the left split-up letter of some letter can be intersected from the right to the left, and vice versa; hence the situation described above cannot be given, if our assumption on ρ implies that the sequence $(L_0, L_1, L_2, \ldots)$ stabilizes as a sequence

of iterated right or iterated left split-up letters. □

However, note that it has neither been claimed nor proved so far that *for a path crossing a corridor the entrance point and exit point have to lie on the corresponding boundary,* and that this fact is not totally immediate from our definitions. However this will be an immediate consequence of our proposition 10.8.

10.7 Remark and Corollaries:

(i) It is not that surprising that a word sequence as a whole uniquely determines corridor exit and entrance points despite the phenomena recalled above, since the higher indexed words of a word sequence also report the fine movements of a paths, and since in the reading process of a word sequence by B.1.4(vi)–(vii) a certain consistency had been required in whether to treat a certain tangency as if an intersection had taken place or as if not. Recall that up to the moment corridor exit and entrance points are are just abstractly defined concepts. In addition note that until 10.8 it will not be clear that corridor exit- and entrance points actually lie where they should by intuition of their definition.

(ii) *The corridor entrance points of the iterated left split-up letters of some letter L coincide with the corridor-entrance point for L, and so does the exit point of L with the iterated right split-up letters for L.* This in an easy consequence of the fact that the supremum and the infimum of Formulae 10.4(1)–10.4(2) clearly is not affected if the first finitely many letters are truncated from the sequence $(L_i)_{i\in\mathbb{N}}$. This result corresponds to the fact that by B.1.4(vii)) the boundaries of each corridor had to coincide with the outer boundaries of the according split-up corridors.

(iii) Note that by 3.2/3.3 a letter which does not have its full set of split-up letters is truncatable. Hence for letters which are truncatable to the beginning or to the end the corresponding corridor entrance or exit point may not be defined, respectively. This is consistent with the result of 2.3: Truncatable letters correspond to such situations when the start point of a path or the end point of a path falls into the region described as a corridor, and where hence the according letter may only describe a partial act of crossing. However by C.1.3(iii) this partial act of crossing may in fact by complete, i.e. some truncatable letters might also give rise to both, their entrance and their exit point.

However, even if the concepts of reading domains and of corridor entrance- & exit points have been set up for giving the same segmentation of a path, they actually need not to coincide in any case. However their non-coincidence is restricted to the exceptional case that a path moves along a corridor boundary for an entire subsegment, and the following lemma gives what can precisely be stated:

10.8 Lemma:

(i) Let L be a letter of a word sequence ω that is read from a path u. Let $[s_1, s_2]$ be the reading domain for L and let t_1 and t_2 be the parameter values that are associated with the corridor entrance- and exit point of L, respectively. Then $s_1 \le t_1 < t_2 \le s_2$.

(ii) If $s_1 \neq t_1$, the $u|_{[s_1,t_1]}$ is entirely contained on the boundary segment of L, and the analogous statement holds if $t_2 \neq s_2$. In addition, the corresponding boundary

segments cannot contain any outer boundary segment of holes that might crop up inside L for higher indices.

Proof: For symmetry reasons we can restrict the following proof to considering the corridor entrance point and the start point of the reading domain of L.

Ad (i): By recycling the proof-techniques of C.4.5/C.2.8 the following line of arguments can be applied to prove the desired statement:
We assume that the parameter value of the corridor entrance point of L is smaller than the start point of its reading domain. By our assumption that the path is not constant, for a sufficiently fine level, there are at least three acts of crossing parallel corridors which have to take place between the two parameter values. We restrict our discussion in the following to the more tricky case where these parallel corridors l_1, l_2 and l_3 are of different basic direction than L is, and hence the corresponding split-up letters could mix. We then define a position sequence κ' that separates all iterated split-up letters of l_2 from those of l_3 (cf. C.5.9). Since the l_3-crossing is finished before the L-crossing starts it would in this case give a contradiction if our split-up letters should actually mix, and hence κ' is still be well-ahead of all split-up letters of L on the sufficiently fine levels. On the other hand κ' must come after the sequence given by the corresponding corridor entrance points, since even if that sequence should for higher indices go between the iterated split-up letters of l_1, the iterated split-up letters of l_2 still give a valid buffer that makes sure that, even in terms of equivalence classes of position sequences (cf. C.5.18(iii)), κ' is a bigger position sequence than our assumed corridor entrance point. And, since on the other hand κ' satisfies the condition 10.5(i) itself, the existence of such a κ' would contradict to the maximality property of a corridor entrance point according to 10.4(1). By symmetry arguments only one other case needs a discussion, namely, if we assume that the parameter value that is associated to the corridor entrance point comes after the start-point of the corresponding reading domain.
Ad (ii): The arguments of (i) can only be recycled in that case, where within the segment $u_{\restriction[s_1,t_1]}$ the moves of our path u have some non-trivial component in the coordinate that is perpendicular to L. In that case on a sufficiently fine level our path u would have to complete the crossing of iterated split-up corridors between the start-point of the reading domain and before the corresponding exit point, and this would then give a contradiction. However, if $u_{\restriction[s_1,t_1]}$ is just a segment on ∂L and if it crosses a lot of fine corridors that are perpendicular to L and are commuting with L, and if we should decide to put all these corridors inside ω into such an order that all the letters of the perpendicular corridors come before the first split-up letter of L, then our definition of a corridor entrance point will adjust the corresponding parameter values to that place where our path actually goes into the interior of our corridor. On the other hand, we will not violate any reading condition if we include our ∂L-segment also into the corresponding reading domain of L and of all its corresponding split-up letters. We could not do this, if a part the corresponding ∂L-segment is the outer boundary of some fine hole inside L, since if such a segment should be treated as a part of the corresponding split-up letters of L, then the conditions B.1.4(vi)–(vii) could not be fulfilled. Note with such

an adjustment of reading domains we are in severe danger of getting a non-extendible reading either, (cf. C.2.5).
According to our two concepts of identifying the start- and endpoints of an act of crossing, the above case (ii) remains the only one where the corresponding points may be said to lie at different places (cf. B.1.4(v), 10.5 and 10.1). □

10.9 Remark on tangencies and in how far they do (not) correspond to cancellable pairs:
Consider a segment of a path $u : [0,1] \to X$. Assume that $u(t)$ is a point on the boundary of some corridor L such that in a neighbourhood of t our path u only takes on values in L. Hence u has a tangency to the boundary of L at t, and in addition we want to assume that the behaviour of u in a larger neighbourhood is so that t could be regarded as marking the endpoint of an act of crossing of L and the beginning of a new act of crossing in the reversed direction. For the purposes of this paragraph we want to introduce a different notation for the area that is described by a corridor on the one hand and for the corresponding letter on the other, and accordingly we will use l for denoting the letter of $\mathcal{A}$ that is associated with L. According to our assumptions we are by C.1.3/B.1.4 given the freedom either to put the letters ll^{-1} into the word of the according level or none of them. In this context observe the following:
(i) If the tangency at t is restricted to one point only, (i.e., if there exists a neighbourhood of t such that for all $s \in$ that neighbourhood with $s \neq t$ we have that $u(s)$ falls into the interior of L), then the pair ll^{-1} is cancellable in the word sequence ω.
(ii) In case that for a certain neighbourhood of t our path u takes on only values on the boundary of L, then reading "ll^{-1}" might be the only facility for reading that leads to a valid word sequence ω.
(iii) Note that we might have the situation of Case (i) and nevertheless we are not given the opportunity for a local homotopy at t that avoids such a tangency.
Hence the situation is not entirely straightforward.

10.10 Proof of the claims 10.9(i)–10.9(iii):
Ad 10.9(ii): Consider the following situation: A path $u : [0,1] \to X$ performs an entire act of crossing of L, and after having completed this act of crossing by reaching the opposite boundary it runs along this boundary for a while. The corresponding segment is $u_{[t,t']}$. Then it enters the interior of the corridor again and performs a second act of crossing of our corridor L in the opposite direction. Now assume that $L \in \mathcal{A}_i$, and that the corresponding split-up line of $\mathcal{N}_{i+1}$-corridors that reuses the boundary line that contains the points $u([t,t'])$ has a hole in the region that is covered by L. Further assume that a subsegment of $u([t,t'])$ is the outer boundary of that hole. Hence the performance of u as described before that could according to C.1.3(i)/B.1.4 on the ith level as well be acknowledged by "ll^{-1}" or by putting none of these two letters into our word, has on the $(i+1)^{\text{st}}$ level to be described by letters $l_1 l_2 {l'_2}^{-1} {l'_1}^{-1}$ which are all split-up letters of L. The assumption that $u_{|[t,t']}$ goes along the outside boundary of some hole implies that $l_2 \neq l'_2$, hence $l_2 {l'_2}^{-1}$ is not a cancellable pair, and all these four letters have to

appear in ω_{i+1}. Hence C.1.7 implies that also "ll^{-1}", despite looking like a cancellable pair, have to appear in ω_i (cf. C.2.4).

Ad 10.9(i): By assumption of our geometrical set-up it seems most likely that for all level indices beyond i the iterated right split-up letter of l and the iterated left split-up letter of l^{-1} follow immediately after each other on each higher level of our word sequence, since the corresponding acts of crossing follow immediately after each other. However, C.1.3(ii) also provides the possibility that nevertheless the corresponding letters need not to follow immediately after each other; on the other hand C.1.3(ii) guarantees that if this happens, then the only letters that can be found between our corresponding split-up letters do commute with them. This guarantees that each of the iterated right split-up letters of l can be cancelled against the corresponding iterated left split-up letter of l^{-1}, provided that on each level index $j > i$ the corresponding letters are inverse to each other. Now this condition is guaranteed by our geometrical assumptions. The situation as described in "Ad 10.9(ii)" gives the only possibility for how such iterated split-up letters could be not inverse to each other; but if this situation should be given for any level index j, it would imply that an entire segment of our path u that is at least as long a the mesh-width of the lattice $\mathcal{N}_j$ has to lie on the boundary line of L; and this would contradict to our geometrical assumption. Hence the corresponding split-up letters all form definitely cancelling pairs, and this makes sure that a cancellation of l against l^{-1} can be extended throughout the entire word sequence (as can be straightforwardly shown with the techniques of 3.2–3.5).

Ad 10.9(iii): It is not hard to construct within the area of L a sequence of holes, at most one taken from one of the higher indexed lattices, such that none of these holes is tangent to the boundary of L, but such that they do accumulate to some point $P \in \partial L$. If then we assume that u is a path which surrounds all of these hole on the side towards the boundary line of L that contains P, then any path with this property (and this includes any path homotopic to u) either has to intersect the corresponding boundary line or it has to meet this boundary line in a tangent point at P. However, this is the situation of (i), not of (ii), since none of the corresponding holes has been tangent to ∂L. Hence none of the lines of corridors containing these holes has had a coinciding boundary line with L. Thus from the arguments of (i) the cancelling pair "ll^{-1}" need not appear in the word sequence that describes u. We just have not note that the non-appearance of such a cancelling pair does not imply that a touching of the boundary can be avoided. □

§11. Analyzing the cancellations when multiplying (reduced) word sequences

Before in the next and last section of this part we will be able to complete the proof of Theorem B.2.10(i), it takes another preparatory step to which this section now is devoted, namely to analyze in how far the concatenations of paths and of word sequences are related to each other, and what the corresponding rules are if we want to restrict

our considerations to reduced word sequences. This in particular will be the basis for extending the result of Theorem 9.1 from closed paths to non-closed paths, and that way this section will provide the final ingredient to complete the proof of Theorem B.2.10(i).

11.1 Lemma: *Let ω and ω' be two word sequences such that by using the notation of C.1.9 for the extension of ω to triple sequences we have that $\vec{\omega}=\overleftarrow{\omega}'$. Let u and u' be paths realizing ω and ω', respectively. Then the concatenation of ω and ω' satisfies that if those truncatable letters at the end of ω and those truncatable letters at the beginning of ω' which within $\omega\omega'$ form a cancellable pair are taken out (this can at most give two cancellations on each level of the word sequence), the resulting word sequence ω'' satisfies all properties of C.1.5(i)–C.1.7(iii) and describes the concatenations of the paths u and u'. In addition we have that $\overleftarrow{\omega}=\overleftarrow{\omega}''$ and $\vec{\omega}'=\vec{\omega}''$.*

Proof: Taking into account that ω and ω' did satisfy all conditions C.1.5(i)–C.1.7(iii), ω'' must fulfill them outside a neighbourhood of the attaching point of ω and ω', also. Near to this attaching point there remain essentially four possibilities how ω'' could fail to satisfy these conditions:

(i) The internal compatibility of some words ω_i'' of ω'' might not be satisfied.

(ii) There might be letters of ω_i'' which do not have enough split-up-letters in ω_{i+1}''.

(iii) There might be split-up pairs in ω_i'' which do not correspond to some letter in ω_{i-1}''.

(iv) Some letters of ω_i'' might show up in inconsistent order to their split-up letters in ω_{i+1}''.

Each of the following four propositions 11.2–11.5 is devoted to rule out one of the possibilities. Hence these propositions together will imply Lemma 11.1.

11.2 Proposition: *With ω, ω' and ω'' as defined in 11.1, the internal compatibility of all the words ω_i'' of ω'' is guaranteed.*

Proof: This is a straightforward consequence of the internal compatibility of ω_i and ω_i' and of the fact that $\vec{\omega}_i=\overleftarrow{\omega}_i$. This gives the internal compatibility for $\omega_i\omega_i'$ and hence for ω_i'', since the internal compatibility can not be overturned if ω_i'' differs by an elementary cancellation from $\omega_i\omega_i'$. □

11.3 Proposition: *With ω, ω' and ω'' as defined in 11.1, all the letters of ω_i'' do have a sufficient number of split-up-letters in ω_{i+1}'' with respect to the demands in C.1.7/C.1.8(i).*

Proof: This proposition needs a more careful discussion, since the exceptional rules C.1.7(ii)–(iii)/C.1.8(i) that permit letters near the beginning or the end of a word to have less than two split-up-letters can be applied to the beginning of ω' and to the end of ω, but not to the interior of ω''. So, let us assume that only the first split-up letter l_1 of some letter L that is placed at the end of ω_i can be found in ω_{i+1}. Then due to the internal compatibility of $\omega_{i+1}\omega_{i+1}'$, there must be some partner letter for l_1 which is contributed by one of the first letters of ω_{i+1}'. It is either taken from the same line of corridors as l_1, or from the line of the split-up companions. Let us discuss the latter

situation at first place:
In this case we may assume that l_2 is the first letter with $BD(L) = BD(l_2)$, and hence it is a possible split-up companion for l_1. We now have to discuss whether within ω' this l_2 is a split-up letter or whether it is not. If it was, then the first letter of ω_i with the same basic direction as L would have to be the letter L again, and this would contradict to the internal compatibility of $\omega_i\omega_i'$ according to 11.1(i). Hence l_2 cannot be a split-up letter in ω' and that way it can be regarded as the second split-up letter of L after having concatenated ω with ω'.
We are left with the case where the first letter of ω_{i+1} with the same basic direction as L is some letter ${l_1'}^{-1}$ that is taken from the same line of corridors as l_1. In this case the internal compatibility of $\omega_i\omega_i'$ gives that ${l_1'}^{-1}$ must be a single split-up letter, i.e. that the letter L^{-1} has to occur near the beginning of ω'. Since letters with only one split-up letter are always truncatable, the two letters L and L^{-1} are to be cancelled when ω is concatenated with ω'. That way the problem of L having not enough split-up-letters in ω'' disappears.
In case where L is within ω a letter with zero split-up letters only, essentially the same arguments can be applied; in this case $\overleftarrow{\omega}_{i+1}$ has to be used for the corresponding compatibility information that could in the above case be read from the letter l_1. Hence we are done in this case, also. □

11.4 Proposition: *With ω, ω' and ω'' as defined in 11.1, all the split-up pairs in ω_i'' do correspond to some letter in ω_{i-1}'' in the sense of C.1.7.*

Proof: Let l_1, l_2 be a pair of letters that occur as split-up companions in ω_i'' and let $L \in \overline{\mathcal{A}_{i-1}}$ be the letter which describes the same act of crossing of an $\mathcal{N}_{i-1}$-corridor as l_1 and l_2 together. If these letters are both contributed either by ω or by ω', then ω_{i-1} or ω_{i-1}' has to contain the letter L at a corresponding place, respectively. If this letter L is truncatable in $\omega^{(\prime)}{}_{i-1}$, then l_1 or l_2 must also be truncatable. Hence we have to ask whether it might happen that L disappears in the course of constructing ω'' (since it is contained in a cancelling pair of truncatable letters), but the corresponding letter l_1 or l_2 does not. If such phenomenon should happen, then (restricting our language to one of the cases now) we read "LL^{-1}" in $\omega_{i-1}\omega_{i-1}'$, but in $\omega_i\omega_i'$ we see "$l_1l_2{l_2'}^{-1}$" with l_2' being a different corridor as l_2 that is from the same line of corridors as l_2. However, a string of letters such as "$l_2{l_2'}^{-1}$" violates C.1.6, since any path on its way from the exit point of l_2 to the entrance point of l_2^{-1} has to cross a corridor p that is perpendicular to $\mathbb{L}(l_2) = \mathbb{L}(l_2')$ and ends at the hole that separates l_2 from l_2'. Hence actually we could only read "$l_1l_2p{l_2'}^{-1}$" where p does not commute with l_2 and with l_2'. Now, p has to be either contained at the end of ω_i or at the beginning of ω_i', and hence it prevents l_2 or ${l_2'}^{-1}$ from being truncatable. But according to 3.2/3.3 this would prevent L or L^{-1} from being truncatable, also, such that the cancellation of "LL^{-1}" could not be performed. Consequently we conclude that, when constructing ω'', L cannot disappear because of being contained in a cancellable pair of truncatable letters without l_1 or l_2 disappearing also. Thus, if l_1 and l_2 occur in ω_i'', so does their root-letter in the standard case discussed above. The exceptional case when the two letters l_1, l_2

have been contributed by ω_i, ω_i' respectively needs extra discussion:
However, here with similar arguments as in 11.3 the internal compatibility of $\omega_{i-1}\omega_{i-1}'$ gives that L must be contained in one of the places, either at the end of ω_{i-1} or the beginning of ω_{i-1}' only. This guarantees that L is also contained in ω_{i-1}''. Thus even such pairs as $l_1 l_2$ satisfy C.1.7(i), and hence the proposition. □

11.5 Proposition: *With ω, ω' and ω'' as defined in 11.1, all the letters of ω_i'' do show up in a consistent order to their split-up letters in ω_{i+1}'' in the sense of C.1.7/C.1.8(ii).*

Proof: By making use of the internal compatibility of ω_i'' and ω_{i+1}'', C.1.8(ii) gives that there is only one possibility left for how two letters L, L' of ω_i'' may have pairs of split-up letters (l_1, l_2) and (l_1', l_2'), respectively, that do not show up in an appropriate order within ω_{i+1}''. That is, if "LL'" gives the order of the ω_i''-letters, that we see "$l_1' l_2' l_1 l_2$" within ω_{i+1}''. It is not hard to verify that, regardless, whether we assume that the borderline between ω and ω' is before, between or after L and L', and no matter where we assume the borderline to be on the $(i+1)^{\text{st}}$ level, in each of the $3 \cdot 5 = 15$ cases the internal compatibility cannot be given either for ω or for ω'. Similar arguments as in the last line of 11.3 hold in case where we do not see the full set of split-up letters. Hence this proposition and with it Lemma 11.1 are proved (cf. 11.1(i)–(v)). □

11.6 Lemma: *Let ω and ω' be two reduced word sequences with $\vec{\omega}=\overleftarrow{\omega}'$. Assume that ω'' results from concatenating ω and ω' and from performing the operations of 11.1 to turn this sequence of words into a word sequence again, and then in addition performing all necessary shortening operations to turn the latter word sequence into a reduced one again. Then in the course of transforming $\omega\omega'$ into ω'' we never have to cancel two letters against each other which belong to the same factor ω or ω'.*

Proof: Let L and L^{-1} be two letters of ω such that L comes before L^{-1}. The proof is based on assuming that L is cancelled against L^{-1} in the course of reducing $\omega\omega'$ to ω''. Without loss of generality we may also assume that these two letters form an innermost cancelling pair of letters with respect to their associated line $\mathbb{L}(L)$ on the corresponding level, since otherwise we can skip to considering such a pair. The following discussion will give that these assumptions yield a contradiction:
$\mathbb{L}(L)$ divides $\mathbb{R}^2$ into two half planes which will be denoted by h_1 and by h_2. The area of $\mathbb{L}(L)$ is considered as belonging to neither of these, it is a separate third region. The index convention for the $h_?$ is hereby set up such that the corridor exit point of L (cf. 10.4) lies on the common boundary of h_1 and of $\mathbb{L}(L)$.
For all level indices i that are bigger than the index to which cancelling pair "$L \ldots L^{-1}$" is associated we look at the subwords η_i of ω_i that are bounded by the iterated right or left split-up letter of L and L^{-1}, respectively. In the following we also define "thinned" variants of our words η_i just by deleting a lot of letters within the η_i. The corresponding rules for deleting those letters are the following:
(i) Each letter that belongs to the same basic direction as L is to be deleted.

(ii) The letters of the other basic direction are only partially deleted, according to the following rules. The most important of these rules states that the remaining corridors shall give a slice-system (cf. 9.2), as it had also been used in the course of the proof of Lemma 9.1 in Proposition 9.5. It is not hard to see that such a slice-system exists and the the subsequent additional conditions give a guide for how to construct such a slice-system:

(iii) In each of the lines of corridors that is perpendicular to $\mathbb{L}(L)$ at least one of the corridors must be taken out (in order to make sure that the complement of all remaining corridors is still path-connected). In the standard case we choose this missing corridor so that the corridor that intersects the boundary line ∂h_1 is the missing one. One of the non-standard cases is given when the corresponding line of perpendicular corridors contains in the region of intersection with $\mathbb{L}(L)$ a very fat hole, i.e. a hole with a diameter that is bigger as the width of L. In that case we take out the first corridor on the h_2-side. The second exceptional situation is given, when the perpendicular line of corridors contains a fine hole that is tangent to ∂h_1 and has a smaller width the L. In that case we do not remove the corridor that by emanating from that hole to h_1 may be also thought of intersecting ∂h_1, but the corridors that emanate on the other side of the corresponding hole to the interior of $\mathbb{L}(L)$ (and perhaps goes even into h_2).

(iv) Apart from the above to we need take out additional corridors in each perpendicular line, when the corresponding line goes through a hole that has a bigger width then the corresponding mesh-width of $\mathcal{N}_i$. As a rule of thumb, for each of those bigger holes we take out all except one corridors, that are emanating to the outside. Here "outside" means: that move away from $\mathbb{L}(L)$. Our course, for those holes that sit inside $\mathbb{L}(L)$, "outside" has to be arbitrarily defined.

By constructing our words η_i as above and by then deleting all those letters within those word that belong to one of the corridors that had to be "taken out" in order a get a slice-picture, we obtain our "thinned words" $\overline{\eta_i}$. The following two propositions 11.7 and 11.8 will both have conclusions that contradict to the assumption as set at the beginning of the proof of Lemma 11.6. Since the assumptions of 11.7 and 11.8 are complementary, by proving these propositions below the proof of this lemma will be completed after 11.8:

11.7 Proposition: *Let η be the sequence of thinned words as it is set up according to the notations and the rules of 11.6(i)–(iv). If each thinned word can according to the calculus of free groups be reduced to such a word that does not contain letters of h_1-corridors any more, then the word sequence ω admits a compatible cancellation pattern that includes the cancellations of L against L^{-1}.*

Proof: Recall that in the course of proving 9.1 we had developed some technique for showing that if a cancellation can be performed to a thinned word, then the original word also permits to be cancelled to such an extend that finally the same letters as of the thinned word may be cancelled. By applying this technique we can for each of our levels i get that far that at the end also the bounding iterated right split-up letter of L

and the iterated left split-up letter of l^{-1} are in a position where it may be discussed whether these letters may also cancel. Assume that on some level i these letters cannot be cancelled against each other. However, with the techniques of 9.1 it is clear that there are not any obstructions for performing such an amount of cancellations that a realizing path of the correspondingly cancelled path could within X_i be homotoped into such a position that instead of having associated with the letter "L" some exit point and some re-entrance that corresponds to L^{-1} the path just remains inside the area of L. We are also definitely able to rule out the C.2.5/10.9–case, since the letters of the corresponding corridors have not been deleted in our thinned word. We also know that the path cannot have gone anywhere else: It had not been in h_2 before, so that cancellations and interchanges of variables cannot have brought it there. Of course, it might cross some corridors that start inside L and emanate to h_2. Our path could also not have gone to regions of $\mathbb{L}(L)$ that are different from L. For that purposes our path would have to pass along the corresponding holes either via h_1 our via h_2. And, as stressed before, it could not have taken the h_2-way, and due our construction any h_1-way would be visible, even in the thinned word, by the appropriated letters. From these considerations we get that on none of our levels there is an obstruction for cancelling L against L^{-1}. By using the techniques of 7.18 (cf. also the proof of 11.10) the cancelling patterns as found on each of our levels separately can be made compatible by Königslemma-type of conclusions so that a reducing pattern in the sense of 1.17 arises that manages to cancel L against L^{-1}. This gives a contradiction to our assumption that ω is a reduced word sequence, already. □

11.8 Proposition: *Let η be the sequence of thinned words as it is set up according to the notations and the rules 11.6(i)–(iv). If the assumption of 11.7 is not satisfied, then neither ω nor $\omega\omega'$ admit a compatible cancellation pattern in which L occurs as the cancelling partner for L^{-1}.*

Proof: Since any of our equivalence operations B.2.2(i)–B.2.2(v) can only affect a word of our word sequence in such a way that the corresponding thinned word either remains unaffected or has been changed by some process as in B.2.2(i), the assumption of 11.8 implies that there does not exist, even on the level of words, a facility to get rid of all h_1-letters in the sense as described above, at least not, as long as we consider the words η_i separately. As negation of the assumption of 11.7 we have that for some i the ith level provides an obstruction for cancelling L against L^{-1} that cannot be overcome as long as the corresponding h_1-letters (that are by construction all non-commuting with any iterated right split-up letter of L) have not disappeared. This result appears quite consistent with our assumption that L and L^{-1} still appear in the reduced word sequence ω. On the other hand this obstruction cannot be overturned by replacing ω by $\omega\omega'$. For, we had assumed that L was to be cancelled against L^{-1} in this pattern, and this rules out that L may (have) disappear(ed) in a different way (i.e. by being cancelled against some ω'-letter). Moreover, we know that up to a certain state L, L^{-1} and a sufficient number of split-up letters must have survived, and that the obstruction that according to our assumption prevents L from being cancelled against L^{-1} must

have gone before the cancellation process can be performed. But how could they have gone? – According to the way this proposition is stated it must have been the influence of some letter from outside of η_i. On the other hand, since η_i is bounded by some (still surviving) split-up letters, any such influence could have only been due to far-distance truncations, far distance cancellations and transpositions. All these processes can have only affected those letters of η_i, that commute with some iterated right split-up letter of L. That implies, that the thinned word $\overline{\eta_i}$ remains unaffected by all these processes, since all its letters have not been commuting with any right split-up letter of $\mathbb{L}(L)$. More precisely, only the reduced representative of the thinned word is known to remain unaffected, since, if other letters then those of the thinned word have disappeared due to the processes as listed before, this might have given way to some partial reductions of the thinned word. In addition, we have to note the "bounding iterated split-up letters of η_i" are not necessarily invariant: Even if we are assuming that L and L^{-1} have not been cancelled, as discussed in 3.15 we might see a process such as $l_1 l_2 \ldots l_2^{-1} l_2' \mapsto l_1 l_2'$ where the old iterated split-up letter l_2 of L disappears and some other letter l_2' takes over the role of a split-up letter instead. In our context such a phenomenon would in particular imply that the "barrier" against further cancellations within η_i had been moved to the inside and hence is less effective than before. However, even that phenomenon does not overturn our claim that the reduced representative of $\overline{\eta_i}$ is not affected by all those cancellations: For, before l_2 could have been cancelled against l_2^{-1}, the part of η_i that is marked by "..." in the above formula has, with barriers as originally given, to be cancelled down up to such letters that are commuting with l_2. In particular this implies that the corresponding part of $\overline{\eta_i}$ has to be cancelled to an empty word, since all its letters had been non-commuting with l_2. And this gives that the reduced representative of $\overline{\eta_i}$ is the same representative that one gets as if one cuts of the corresponding "..."-part of $\overline{\eta_i}$. Hence the obstruction that is given by some $\overline{\eta_i}$ that has a non-empty reduced representative could not have been overturned at the time where L is supposed to be cancelled against L^{-1}, and this is a contradiction to the assumption of 11.6. □

11.9 Lemma: *With the same notations and assumptions as in 11.6, an interchange between two letters that belong to the same factor ω or ω' does not occur when $\omega\omega'$ is transformed into ω''.*

Proof : Now we assume that L_1 and L_2 are two commuting letters of ω such that L_1 occurs before L_2 within ω, and such that in the course of transforming $\omega\omega'$ to its reduced representative ω'' these letters are at some state transposed. Of course, our assumption implies that it is according to B.2.3 a shortening transposition to interchange L_1 and L_2, which due to some curious compatibility obstruction can not performed inside ω unless the additional factor ω' is appended. In order to show that such curios compatibility obstructions do actually not exist, we use the same line of arguments as in the case of 11.6, and hence it will now suffice to give an outline below:
By skipping to an according innermost pair we may assume that our path on its way from the exit of L_1 to the entrance of L_2 does return into neither of our corridors that far that it would complete an act of crossing in the inverse direction. $\mathbb{L}(L_1)$ and $\mathbb{L}(L_2)$ must

be two perpendicular lines of corridors that intersect each other in a crossing region, and hence the relevant segment of our path lies essentially in one quadrant of $\mathbb{R}^2$ that is bounded by L_1 and L_2. This quadrant and the corresponding segment of our path will be the h_1 and the η_i that have to be considered in this case of the proof, respectively. h_1 must be an upper left or upper right quadrant, as follows by B.2.3 from the assumption that the interchange of L_1 and L_2 is assumed to be of shortening type. As in the case of 11.6, the set of corridors in h_1 has to be thinned in order to give a slice system of corridors again. However, the geometrical phenomena that we get into our slice picture are a bit richer as in 11.6–11.8, h_1 is now a quadrant and not just a half-plane. In particular this gives that not all slice-corridors can be chosen to be perpendicular to ∂h_1 any more. Instead we will choose our slice-corridors in such a way that they are all perpendicular to $\mathbb{L}(L_1)$, and with the same technique as before, we avoid to get slice-corridors that commute with the corresponding iterated right split-up letter of L_1. According to that all slice-corridors will be parallel to L_2, and we also might have to cope with such corridors that entirely lie inside L_2. However, since $L_1 \cap L_2$ is free of holes, we can also keep it free of slice-corridors. Observe that this construction guarantees, as in the proof of 11.6, that none of the slice-letters commutes with the iterated right split-up letter of the L_1 or with the iterated left split-up letter of L_2. In the L_1-case this is due to the same arguments as in the proof of 11.6, in the L_2-case this follows from using the simple fact that letters of the same basic direction do not commute. This makes the reduced representative of the thinned word $\overline{\eta_i}$ a similar useful invariant as in 11.6–11.8: Namely, it can be told from the reduced representative whether η_i can be realized by a path that entirely remains inside $L_1 \cup L_2$ or whether there are homotopical obstructions inside X_i for finding such a position for our path. And then the same lines of arguments can be used as we had used before in 11.8 and 11.7: If this obstruction appears on some level, it turns out to be an obstruction that holds the same for ω as for $\omega\omega'$, and if we assume this obstruction to be absent for all levels, then we can firstly extend the reduction pattern from the $\overline{\eta_i}$ to the η_i, then make them compatible and finally conclude, that ω would have to permit a reduction scheme that is able to interchange L_1 with L_2. In that context note that here the corresponding arguments need some refinement with respect to what had been considered in 11.6/9.1, since h_1 is now just a quadrant and not a half-plane: The crucial technique of 9.1 that had been quoted in 11.6–11.8 was based on tracing the essential action of our path (according to the combinatorial description by the relevant word, i.e. in principle we are just considering the edge-path as described in C.1.6) and finding some place where the path could either be pulled into some simpler position, or where the obstruction for doing this would have already been visible in the thinned word. The search mechanism was essentially been based on finding half-turns of our path, and, as long a h_1 was a half-plane, any path that comes in and goes out again has to perform some half-turn. However, our path could perfectly go through our quadrant h_1 without performing any half-turn. However, it is not hard to see that our path either has half-turns (so that one can argue as in 9.5–9.10), or it goes monotonously in both coordinates through our quadrant. In that case our geometry is set up in such a way that in order to push our path into a position where it makes its

way inside $L_1 \cup L_2$, it only takes such kind of transpositions that, if they should be valid, are of shortening type. Since the non-valid transpositions can be traced in the thinned word, the case of the quadrant h_1 can be analogously treated as the half-space case; all it takes is an adding of the case as just described to the list of cases as discussed in 9.6–9.10 and some noting that the arguments as used in 9.4/9.5 suffice to treat this case as the other ones.
Having observed that the situation of this Lemma 11.9 can be analogously treated as the situation of Lemma 11.6 before, 11.9 is proved. □

11.10 Lemma: *Assumptions as in 11.6. Then the transformations that turn $\omega\omega'$ into ω'' can be performed while sticking to the following three phases each of which may be empty:*

Phase 1: Some end segment of ω is completely cancelled against some initial segment of ω'.

Phase 2: Far distance cancellations between letters of ω and letters of ω' are performed. All letters affected by this phase can only be of the same basic direction.

Phase 3: Now the transpositions between some of the letters of ω and of ω' are to be performed.

Proof: The same scheme of proof can be applied as for 7.18, and this essentially means that we have a two-step argument:

(i) We look at each of the words separately and make sure that that for each level there exists at least one possibility to put the corresponding processes into the desired order. This is clear, since by 11.6 all cancellations to be performed are somehow nested around the attaching point of ω and of ω', and hence there is a natural order to perform these cancellations. Our demands (1.)–(2.) do not contradict to using this natural order. On the other hand, since only commuting variables may be interchanged, by performing transpositions we can never obstruct or give way to cancelling variables. Hence there is not an obstruction for putting all transpositions into some final Phase (3.).

(ii) After having made sure that there all transformation can be put on each level in the desired order, we apply Königslemma-type of conclusions to find a compatible pattern with this property, also. As in 7.18, here this kind of conclusions will essentially be performed by finding stabilizing patterns within the nested graph. Two essential conditions have to be given to make the corresponding scheme of conclusions work, namely:

- ▷ The relevant parts of the nested graph have to be finite on each level — and
- ▷ We have to be given the "upward-extension property", i.e. by some other source we must know that each transformation pattern on some level can be extended in such a way to the less fine level of our word sequence, that the desired also hold for these extensions.

Similarly as in 7.18 both of these properties are given, since here we have to deal with shortening transformations, only. □

When we will be discussing a situation as described in 11.10, we will be given the task of comparing the positions of $\omega\omega'$ and of ω'' with each other. In that context the following observation will be essential:

11.11 Proposition: *Let η be a word sequence and let η' be a word sequence that arises from η by applying a compatible sequence of cancellations and truncations (in the sense of B.2.1(ii)/B.2.8) to η. Then any $\kappa \in \mathbb{P}(\eta)$ naturally extends to some element of $\mathbb{P}(\eta')$.*

Proof: The natural extension in this situation shall be given by looking at the dual split-up picture as defined in C.5.5: When white vertices are deleted, the yellow vertices should persist, and in the course of these identifications some of the yellow edges might be identified, too; however in any case such processes can be regarded as passing from the graph of position sequences of η to some quotient graph which hence can be regarded as the corresponding graph of η'. Thus this quotient map is what allows us to define the "natural extension" of κ to $\mathbb{P}(\eta')$: Recall that by C.5.5 a position sequence is pictured by an infinite branch in the yellow graph of η, and hence the image under the quotient graph-map defines that position sequence of $\mathbb{P}(\eta')$ that according to the above convention will also be denoted by κ.
There has to be a postscript to these proof-arguments, since we missed to note one kind of phenomenon that, apart from the mentioned ones, might also be found on the white level: For, truncations and deletions on the level of word sequences can cause more phenomena than just deletion of vertices and white edges, namely as already seen in the proof of 3.15, they can also cause readjustments of some of the white edges (or arrows). As already described in 3.15: If we see as adjacent letters of one basic direction "$l_1\, l_2\, l_2^{-1}\, l_2'$", (our use of indices corresponds to 5.11–5.12) then l_1 and l_2 are split-up companions of the same root-letter L. If now l_2 is cancelled against l_2^{-1}, then l_2' steps in and takes over the role of the split-up companion for l_1. Hence one white arrow in the dual split-up picture is to be readjusted, and, unless we have not made sure that this readjustment does not cause new obstructions for putting yellow edges into our dual graph picture, we cannot be sure that the proof arguments as indicated in the first paragraph of this proof are actually applicable. However, in the following paragraph we will give the corresponding reasons. In that context in particular note that according to 1.3/8.4/8.6–8.9 the phenomenon as described above is the only one that makes white arrows appear in positions in which they had not been before. The arguments as given below will enable us to finally agree that what as been said about the yellow level in the first paragraph of this proof is true.
Any new obstruction for yellow edges would have to come from intersecting both white arrows of our pair "$L \to l_1$" and "$L \to l_2'$". However, regardless whether we assume that the start vertex for the yellow edge is to the left or to the right of L, in any case any edge intersecting these both arrows would have to intersect "$L \to l_2$" also, and hence such an edge could not have existed in the yellow η-graph as well. Hence our readjustments of white arrows can never give new obstructions to the existence of yellow edges, and

hence κ extends to $\mathbb{P}(\eta')$ by the arguments as given in the first paragraph of this proof.□

11.12 On notational conventions and problems that come with comparing positions in word sequences to which transformation processes are to be applied.
Let us assume that ω and ω' are word sequences as in 11.10, and that, in particular, some initial subword-sequence of ω had not been cancelled against ω', so that we can expect that this part of ω is turned into ω'' just by taking some of the letters of ω out and bringing in some other letters of ω' instead according to 11.10(2.) and 11.10(3.), respectively. The latter could have happened by transpositions according to 11.10(3.), and in this context in particular note that the results 11.6/11.9 permit to treat all letters of ω (that are not cancelled) as fixed coordinates, and to regard the letters of ω' as the only ones that are moved when transpositions are to be performed. We now seek for a facility to compare in the initial part of ω that had been unaffected by 11.10(1.)-cancellations the $\mathbb{P}(\omega)$-position sequences with those of ω''.
At this point we should mention that we will use for position sequences a similar convention as we had in B.1.7 introduced for letters of words: I.e. if we have some $\kappa \in \mathbb{P}(\omega)$, and if ω'' is a word sequence that is has enough letters in common with ω so that there is a natural way for defining "the analogous position of ω''", this position shall be also denoted by κ. However, this convention of notation cannot replace a definition of what can be regarded as a "natural extension" and a clarification how many letters ω and ω'' need to have in common before such an extension exists. Such a concept of extension had been defined at beginning of the proof of 11.11, however there is some other aspect which should be pointed out:
On the one hand our way of extending positions from one word sequence to another that is obtained by only deleting letters and our decision of using the same variables for denoting the corresponding positions in the other word sequences might seem natural and is analogous to our use of letter-variables (cf. B.1.7), from another point of view it is fairly inconsistent: Recall the in C.5.3(i) the corresponding positions had been identified essentially by counting the numbers of preceding letters on each level, and accordingly our elements of the $\mathbb{P}$-sets are defined as such integer sequences. If now we skip from some word sequences ω by deleting letters to another word sequence $\widehat{\omega}$ and decide to also use the variable κ that originally that been defined in $\mathbb{P}(\omega)$ for denoting the corresponding $\widehat{\omega}$-position, then due to some deletion of the corresponding preceding letters the integer entries as originally defined for κ might not be give the correct number of preceding $\widehat{\omega}$-letter any longer. Despite this somehow inconsistent notation we stick to our decision for how to extend ω-positions to $\widehat{\omega}$-positions and how to use the corresponding variables. However, to weaken the inconsistent aspect of this natural extension a bit, we will avoid to call κ "position sequence" if κ is used to identify an $\widehat{\omega}$-position and was originally introduced as a position sequence for ω. In that context we will prefer the phrases "positions" or *"position variables"*.

11.13 Remark: *(On how to compare the position sequence with those word sequences that arise in the course of transformation processes).*
This remark is devoted to discuss in how far Proposition 11.11 and Convention 11.12

can be applied to the situation of 11.10 where ω is to be turned into ω'':
According to 11.10 one can think of performing the corresponding transformations in three steps, and hence by applying the classical Königslemma (cf. C.5.10) to the transformation graph (cf. 1.18), we get a word sequence $\widehat{\omega}$ that is obtained from ω by applying the Phase-1 and Phase-2 operations to $\omega \cdot \omega'$ only, so that $\widehat{\omega}$ can then be turned into ω'' by performing the transpositions of Phase 3, only. The above criterion permits to extend all position sequences of the initial part of ω to $\widehat{\omega}$, i.e. of that part that is not cancelled in the course of Phase 1. In order to see that there is also some facility to compare $\widehat{\omega}$-position sequences with those from ω'', we regard the situation as follows. By 11.10 the effect of the Phase-1 operation is to turn $\omega \cdot \omega'$ into $\omega|_{[0^{\mathbb{N}},\lambda]} \cdot \omega'|_{[\lambda^{-1},\ell^{\mathbb{N}}]}$ (cf. C.5.24). Phase 2 then imposes some further changes to these two factors by taking out some additional letters — we will not introduce names for the word sequences that we see then. However, we then turn our two-factor product into a three-factor product but simply attaching the inverse of the second factor as the new third factor. We then impose the changes of Phase 3, i.e. we bring some of the ω'-letters into the the remains of ω. Before performing these processes it has clearly been possible to cancel our third factor against the second; and since, as already pointed out in the course of 7.18, transpositions can never obstruct cancellations, the corresponding cancellations can still be performed in the present state of our product of word sequences. I.e. in principle all the newly inserted letters that give the difference between the initial substring of $\widehat{\omega}$ and the analogous substring of ω'' can be made disappear again by accordingly set up cancellations. And that way we see that our criterion can be applied a second time: this time it tells us that each position sequence in the corresponding initial substring of ω'' extends to some position sequence of $\widehat{\omega}$.
However, the above facility for how to compare the position sequences of ω with those of ω'' will be the only one that can be used in subsequent proofs: Both types of position sequences can be naturally be extended to the set of position sequences of some third word sequence $\widehat{\omega}$, and as $\widehat{\omega}$-position sequences they can be compared according to all facilities of C.5.15.

In particular the results 11.6/11.9/11.10 will enable us to extend Theorem 9.1 and to consider in the following section two arbitrary relatively homotopic paths instead of one nullhomotopic one.

§12. Combinatorial characterisation of relative homotopy and summary of Part D

This section is devoted to give the final steps of the proof and to summarize the results that describe the geometrical interpretation of the combinatorially defined equivalence relations. Note that by these results the proof of our Theorem E as listed in the introduction and of Theorem B.2.10(i) will also be completed.

12.1 Theorem: *Two word sequences are weakly equivalent if and only if they represent relatively homotopic paths.*

Proof:

"$\Longrightarrow$" This follows from the same arguments as the corresponding direction of 9.1, observe that none of the results as quoted there was set up so as to deal with the nullhomotopic case only, they were all set up so as to hold in the general case as well. Hence the same line of arguments as in 9.1 can be used.

"$\Longleftarrow$" This is the only claim of 12.1–12.3 which still needs some proof, and for which the results of Section §11 provided the essential preparatory steps. We will show this direction by proving the contraposition. So, let us now assume that ω and ω' are two word sequences which are not weakly equivalent. By 7.9 there exist reduced representatives ω_0 and ω_0' which cannot be firmly equivalent. We will be done after having proven that $\omega_0^{-1} \cdot \omega_0'$ cannot be equivalent to the empty word sequence, since by 9.1 this implies that the corresponding concatenation of paths is not nullhomotopic and hence that the paths that are associated with ω and ω' cannot be relatively homotopic either. Note that by "$\Longrightarrow$" it is already clear that the path associated with ω is relatively homotopic to the path associated with ω_0, and similar for ω' and ω_0'.

So we are now left with proving that $\omega_0^{-1} \cdot \omega_0'$ which must describe a non-nullhomotopic path is not-exhaustible in the sense of 9.1("$\Longleftarrow$"). This proof will be given in 12.4 below. However, before entering this more technical discussion, we give a summary of similar results:

12.2 Theorem: *A path is constant, if and only if its associated word sequence is bounded.*

Proof: By 3.2/3.3 all letters of a bounded word sequence are truncatable, and hence by B.2.9(ii) such a word sequence is even strongly equivalent to the empty one. According to C.4.8(iv) any non-constant path gives an unbounded reading. Conversely, in C.4.8(ii) it had been proven that only bounded word sequences can be read from a constant path. □

12.3 Theorem: *Two parametrizations describe the same abstract path, if any only if any two word sequences that can be read from them are strongly equivalent.*

Proof: This is a essentially a rephrasing of 2.17. □

12.4 Proposition: *Let ω_0 and ω_0' be reduced word sequences that describe paths which are not relatively homotopic and have coinciding start points and coinciding end points. Then $\omega_0^{-1} \cdot \omega_0'$ is not weakly equivalent to the empty word sequence.*

Proof: Assume the contrary and discuss the possible cancellation patterns that a word sequence $\omega_0^{-1} \cdot \omega_0$ could have taking into account that ω_0 and ω_0' are reduced word sequences already. By 11.6 there cannot be any cancellations within the letters of ω_0 or within those of ω_0'. By using similar techniques as in the proof of 9.1 (in particular cf.

9.4) we can by a suitable norming of our endpoints get rid of truncations, and, besides this, we can avoid transpositions by 7.18. Thus, all what is left are cancellations of pairs of letters one of which belongs to ω_0 and the other to ω_0'. Hence, when thinking of an arch-system as pictured in A.4.8(Fig.d), we get a system of arches which are mostly centred around the attaching position of $\omega_0^{-1}\&\omega_0'$, some of the neighbouring arches of which might be intersecting. Due to the particular restrictions on our transformation process, the transformation diagram (cf. B.2.5/1.3) on each level actually gives such an arch-picture. We cut the bottom bar of this arch-picture into two different segments so that one corresponds to the letters of ω_0^{-1} and the other to those of ω_0'. We then apply a homeomorphism of the plane to this picture, so that the ω_0'-bar remains in its position, and that the ω_0^{-1}-bar is moved to some position above the ω_0-bar while a 180°-rotation is performed to it. Due the the 180°-rotation the lines of our transformation diagram (which had been upward directed at their start-points on the the ω_0-bar, as long as this bar had been in their old position) are now downward directed at their new start points. The 180°-rotation also causes that the lettering of our bar should be readjusted from the corresponding word of ω_0^{-1} to that of ω_0, and hence we will call the corresponding bar in its new position the "ω_0-bar". Since according to our assumptions we had a partially intersecting arch-system before, after these line have been moved by our homeomorphisms into their new positions, we can adjust them in such a way that they are all monotonous with respect to the y-coordinate. That way our homeomorphism of the plane has transformed our cancellation pattern into a transposition pattern. This transposition pattern gives a valid sequence of transformations on each level, since all intersecting lines belong to commuting variables (according to the old picture intersecting arches could be given in case of far-distance cancellations, and hence the associated variables had to be commuting by B.2.2(iv)). Even if one might seriously wonder whether our method of turning a compatible cancellation pattern of $\omega_0^{-1}\omega_0'$ into a sequence of transposition processes that turn the words of ω_0 into the words of ω_0' has any chance to fail giving a compatible pattern again, the techniques of 11.10/7.18 provide at least one facility to reconstruct compatibility again (note that even 1.9 guarantees that it never takes more than transpositions to get an upward extension for transpositions on the finer level, and, apart from that, the finiteness conditions of the relevant segments of the nested graphs are given, since we did not introduce new variables in the course of our transformation processes).
Thus there exits a compatible transformation pattern that transforms ω_0 into ω_0'. Hence ω_0 and ω_0' are proven to be weakly equivalent, and due to the transitivity of weak equivalence this implies that ω must be weakly equivalent to ω', also. Since this would contradict to the basic assumption of this proof direction, we at least found some contradiction to the inner assumption that we made at the end of the first paragraph. I.e. we now proved that $\omega_0 \cdot \omega_0'$ is not exhaustible, and hence by 9.1 it cannot describe a nullhomotopic path. As exposed in 12.1, this implies that ω and ω' cannot describe relatively homotopic paths, either, and completes the proof of 12.1("$\Leftarrow$"). □

12.5 The postponed Proof that the homotopic distance as introduced in A.5.2(2) actually gives a metric for $\widehat{X_x}$:

Symmetry, the triangle inequality, and most parts of the zero-compatibility are straightforward; there is just one aspect of the zero-compatibility which needs some consideration: Namely, it has to be proved that the homotopic distance of two pairs (P_1, α_1) and (P_2, α_2) with $\alpha_1 \neq \alpha_2$ but $P_1 = P_2 =: P$ cannot be zero. Proving this property means that we have to prove that the homotopy class $\alpha_1^{-1} \cdot \alpha_2$ cannot be represented by a sequence of paths $(u_i)_{i \in \mathbb{N}}$ such that the diameter of the paths of this sequence tends to zero. Although there might be direct elementary proofs for this fact, we now prefer to deduce it as an immediate corollary from our theory of word sequences:
Assume there exist such paths u_i and denote the associated word sequences by ω_i. Since the u_i are assumed to be homotopic to each other, by 12.1 all these word sequences have to be weakly equivalent. I.e. there exists some reduced word sequence ω which is unique up to firm equivalence and equivalent to all the ω_i. Since $\alpha_1^{-1} \cdot \alpha_2 \neq 1$, by 12.2 ω cannot be empty. In addition it has to contain on some level at least five letters so that at least one of them cannot be the first or the last letter of the corresponding basic direction in ω or in any word sequence that is firmly equivalent to ω, and such a letter l must report a complete act of crossing some corridor. Since ω and its firmly equivalent representatives were assumed to be reduced representative for all the ω_i, this letter must occur in a similar position in all the ω_i. Consequently, all the paths u_i perform a completed act of crossing of the corridor that is associated with l, and this rules out that the diameter of these paths can tend to zero.
Since this was the only non-straightforward aspect of the zero-compatibility, we now see that the homotopic distance indeed gives a metric on $\widehat{X_x}$. □

Abschnitt E: Proofs of Section §B.3

§1. Motivation and initial definitions and lemmata

1.1 Motivating Remark: Recall that the purpose of this part is closing the main gap that has been left in the proof of Theorem A so far, and this means proving the lifting component B.3.2 of the existence theorem A.5.6(iii) of a continuous homotopy combing. As already mentioned in B.3.3, this essentially requires constructing a synchronous parametrization for the combing paths so that those paths that have been defined in Section D/Formula B.3.1(1) and are homotopically near get parametrizations which are also close to each other with respect to the supremum metric. Recall that according to A.6.1 there is not any natural compatibility between the homotopic distance and the supremum metric between paths.
As already stressed in A.4.10, our synchronous parametrization will have to individually respond to the behavior of each combing path, speeding up in the wild segments of the

path and calming down in the tame segments. Hence there appears hardly any alternative apart from considering the associated word sequences of the combing paths and to use the information given by them in order to synchronously parametrize our paths. However, this is a fairly delicate task, since the parametrizations to be defined have to satisfy an overall continuity with respect to the entire system, while the information as given by a word sequence is an essentially discrete information. If a path is just moved a tiny bit, this may suffice to turn a non-complete act of crossing a corridor into a complete one (and vice versa), and cause that in the associated word sequence an additional letter appears or some level has to be deleted. Hence, if subsequently we will use the data given by word sequences to synchronously parametrize the paths of our homotopy combing, we have to make sure do that this does not cause a non-continuous behaviour of the parametrization. It should be pointed out that the definitions that are given below do not achieve this overall continuity if they are just applied to an arbitrary system of paths. However, for our purposes it will suffice to use them since by for the proof of our Lifting Theorem B.3.2 we need only consider a system of paths that are associated with reduced word sequences (cf. 1.3).

1.2 Motivating Example: Consider some open set of X which is parametrized by some individual local coordinate system, and assume that within this (x, y)-coordinate system $\{(x, y) \mid 0 \leq y \leq 1\}$ is the area of some corridor. A system of paths is passing through this coordinate system, so that the corresponding segments passing through this neighbourhood might be parametrized by $u_b : t \mapsto t^2 + b$ for some $b \in [-\frac{1}{2}, \frac{1}{2}]$. This is the situation explained above in 1.1: Those paths with $b < 0$ do cross the corridor twice in reverse direction, and hence their word sequences have to contain the corresponding cancelling pair of letters, and those with $b > 0$ do not. Consequently we have to expect that the synchronous parametrizations respond quite differently according to whether the corresponding path has a b-parameter that is bigger or that is smaller than 0. However, for the following reasons even such a considerably different respond will not mess up our continuity:

(i) If we assume that some neighbourhood of the origin in our coordinate neighbourhood is free of punctures (i.e. if it entirely belongs to the space X), then the given system of paths could not have been reduced, since all of them could have been homotopied into a position where none of them crosses the corridor of consideration.

(ii) If there should have been a hole in this region and if some of the paths have passed along this hole on the upper side and others on the lower, and if the region containing the endpoints of our system of paths should have been far away from the origin of our local coordinate system, the corresponding paths could not have been homotopically near according to A.5.2(2). Hence the different responses of our scheme of synchronous parametrization do not matter in such a case.

(iii) If we assume that there is a hole as in (i), but if all of our paths reach their endpoint inside or near to the local coordinate system, the different responses of our synchronous parametrization will also not obstruct to the desired continuity, since it will only affect those segments of our path where we know them to be near the region containing the endpoint, either.

By having considered the Cases (i)–(iii) we have seen an outline of the main continuity arguments that we will expose in complete detail below in 3.2–5.11.

After these introductory remarks and examples it is now time to start with the proof of the lifting theorem. As stressed in the remark following the statement of Theorem A.5.6, it is one step to verify that having obtained almost unique reduced representatives of weak equivalence classes of word sequences gives us at least a valid homotopy combing. This is the task of the following:

1.3 Lemma-Definition: Let X be a space in Normal Form, (cf. A.3.3 and A.3.6) so that by B.1.1 and B.1.3 a corridor network $\mathcal{N}$ and the alphabet $\mathcal{A}$ have been associated with X. Then we define the following system C_X of paths on this space, which, as will be proved below, gives us a homotopy combing for X:
Take the alphabet $\overline{\mathcal{A}}$ and construct all freely reduced word sequences that describe paths with the start point (0,0). The homotopy combing C_X contains precisely all paths that are associated to the firm equivalence class of those word sequences (cf. B.2.9(iii)).

Proof that C_X is a homotopy combing: Let $P \in X$. Since X is by assumption pathwise connected and $(0,0) \in X$, as a consequence of the adaptation process, there exists at least one path running from (0,0) to P. By C.2.1 the path can be described by a word sequence, and by D.7.9/D.2.1 this word sequence can be reduced without affecting the endpoints of the associated path. Hence our homotopy combing contains at least one path ending at P. In addition, if $\pi_1(X)$ should be non-trivial, there exist so many relative homotopy classes of paths running from (0,0) to P as $\pi_1(X)$ has elements, and since by D.12.1 the reduction of word sequences does not affect the corresponding homotopy class, our homotopy combing contains for each point P at least one representative in each relative homotopy class. Now, by D.12.1 the free weak equivalence classes correspond precisely to the relative homotopy classes, and, in addition there is only one reduced firm equivalence class within each of these equivalence classes (cf. D.7.9). Since, on the other hand, firm equivalence classes are by definition B.2.9(iii) subclasses of strong equivalence classes, and since by D.12.3 all representatives of them are associated to the same abstract path, we in addition get that the system of paths as defined above does not contain more paths as required for a homotopy combing.

1.4 Remarks:

(i) Recall that the requirement to use the free weak equivalence relation in 1.3 means that any possible truncation should be performed (cf. D.3.6). In this context note that the start point of all paths, i.e. (0,0), lies by definition outside of all corridors (cf. B.1.1), and hence by D.2.3 none of the relevant word sequences can contain any letters that are truncatable to the beginning.

(ii) *Note that within the homotopy combing C_X we do not have the phenomenon of paths that are not uniquely determined by their word sequences.* By C.1.9 this could happen if paths are constant, entirely horizontal or entirely vertical, since in that case the word sequence only contains letters of at most one of the basic directions, and those letters do not necessarily describe the constant coordinate. Of course, even C_X does contain some of those word sequences; however, since by definition all

relevant paths start at (0,0), there is not any uncertainty any more about the constant coordinate. Since for the beginning of our word the list of truncatable variables is empty according to (i), there also is not any uncertainty about the first and third entries of the corresponding triple sequences either: Recall that according to (i) the start point of our paths lies outside of all corridor boundaries so that the phenomenon of appending a letter at the beginning and truncating it to the end again as described in D.7.17(2.ii) cannot be applied to some path of our homotopy combing X.

1.5 Remark: The purpose of this remark is to briefly discuss why we are not going to use the parametrization as constructed in the course of our construction C.3.1 as the synchronous parametrization as described in 0.1/B.3.3.
The main obstruction that will force us to define in 1.7–Sect.§2 the new synchronous parametrization for our homotopy combing C_X is not due to the fact, that in D.7.9 we did not obtain unique minimal representatives (but only minimal firm equivalence classes of them), since our C.3.1-parametrization does respond the same to each firmly equivalent representatives. We will not include a rigid proof of this fact (since we will not need it), but it should be pointed out here, that the techniques of C.3.1–C.3.17 for how to prove that our approximating paths converge to the same limit could easy be adapted to give a proof that approximating paths as constructed for firmly equivalent representatives converge to the same (parametrized) limit path. In this context note that the main problem of C.3.1–C.3.17 was to control the possible order of the split-up letters, and the possible order of the letters of another firm representative is bounded by a very similar set of conditions.
So, having to cope with firm equivalence classes can hardly be considered as the crucial obstruction in our context. Moreover, this obstruction will be related to a very bad behaviour of the C.3.1-parametrization with respect to the phenomena as described in 1.1/1.2. Consider in particular the following situation:
A system of paths is given by a sequence of polygons with two segments, here to be denoted as $\overline{P_i\,Q_i} \cup \overline{Q_i\,R_i}$. Assume that all the segments $\overline{P_i\,Q_i}$ are horizontal and are oriented to the right, and that all the segments $\overline{Q_i\,R_i}$ are vertical and upward oriented. Due to the effect as described in D.3.10 such a behaviour of paths might easily occur even in our homotopy combing C_X. Thanks to our considering of vertical and horizontal segments only, the effect of the C.3.1-parametrization can easily be traced: Each intersection with a perpendicular boundary line gives a node, and the values that are taken on by the approximating paths at the same node have to lie on the same corridor boundary (since in our example we do not see any funny order of split-up letters, we do not have to cope with effects as discussed in C.3.4). If then we in particular consider a sequence of points Q_i such that all of the $\overline{P_i\,Q_i}$-segment complete a certain act of crossing a corridor, but the limit paths just ends on the outer boundary (so that it does not complete this act of crossing on its final vertical segment, also), then in a reduced word sequence the corresponding letter might not appear. Consequently, if the C.3.1-parametrization would have directly been computed for the limit path, (at least in a certain segment) a different number of nodes is defined and different values adjusted to them. Since these nodes are according to C.3.1 used as "essential nodes" in the remainder of the construction and

their values are used for all approximating paths, the effect of having the values of these nodes differently adjusted will also be visible in the final parametrizations as provided by C.3.1. Hence the C.3.1-parametrization definitely behaves in a non-continuous way to the overall system of paths, and hence it cannot be used for our purposes. Hence we will have to construct a new "synchronous" parametrization below in 1.7-Sect.§2. However, some other bad feature of our old parametrization should be observed before:

1.6 Remark on the effects that *can make a parametrization as implicitly constructed in C.3.1 to come up with constant segments:*
Similar methods as in 1.5 can be used: We should start with some path which according to a valid reading has two different start- or endpoints of reading domains at the same parameter value. These phenomena may occur precisely in a tangency case as described in C.1.3(i), or if a path goes through the corner point of a crossing region of corridors or a hole, or if a path starts or ends on a boundary line of a corridor so that the exceptional nodes that are fixed at the parameter values 0 and 1 coincide with the ordinary start- or endpoint of the reading domain of the act of crossing. The easiest way of constructing corresponding examples is probably given, if one uses, as in 1.5, a path, where split-up letters do not show up in reversed order. In that case our having adjusted different nodes to the corresponding start- and endpoint of reading domains and our keeping them fix during the approximation process, even if they correspond to to the same point in the limit path, will result in a path which has a corresponding constant segment.

1.7 Proposition: *Any path can be parametrized in such a way that it has not any constant segments.*

Proof: Let $u : [0,1] \to X$ be a path which may have constant segments. We want to get rid of these by constructing a monotonously increasing function $\varphi : [0,1] \to [0,1]$ which takes on constant segments on precisely those intervals on each of u also has a constant value. For such a φ the composition $u \circ \varphi^{-1}$ is a well-defined function that gives a reparametrization of u that cannot have constant segments any more.
We proceed by constructing functions φ_m so that φ finally will be obtained as the limit function of the φ_m. All the φ_m arise by linearly interpolating the values $\varphi(a_{k/2^n}) = \frac{k}{2^n} = \varphi(b_{k/2^n})$ for k odd, $n \in \mathbb{N}$, $m \leq n$, $\frac{k}{2^n} \in [0,1]$ for the $a_?$-nodes or $b_?$-nodes, respectively. The corresponding nodes are defined as follows:

- $a_0 = 0$ and b_0 is the maximal value so that $u_{\restriction [0,b_0]}$ is constant. I.e. if u does not start with a constant segment, then b_0 is zero.
- Analogously a_1 is defined as the minimal value so that $u_{\restriction [a_1,1]}$ is constant and $b_1 = 1$.

This scheme of definition is iterated by letting

- $b_{k/2^n}$ be the maximal value so that $u_{\restriction [(a_{\frac{k+1}{2^n}} + b_{\frac{k-1}{2^n}})/2\,,\, b_{k/2^n}\,,\,]}$ is constant, and by finally letting
- $a_{k/2^n}$ be the minimal value so that $u_{\restriction [a_{k/2^n}\,,\,(a_{\frac{k+1}{2^n}} + b_{\frac{k-1}{2^n}})/2]}$ is constant.

The following properties are immediate from this definition:

- $0 = a_0 \leq b_0 < a_{1/2^n} \leq b_{1/2^n} < a_{2/2^n} \leq b_{2/2^n} < \ldots \leq b_{\frac{2^n-1}{2^n}} < a_1 \leq b_1 = 1$ for all n.
- φ_m is monotonously increasing and precisely takes on constant segments on those intervals $[a_{k/2^n}, b_{k/2^m}]$ that are not degenerated to one point.

$\triangleright$ $\varphi_n|_{[a_{k/2^n}, b_{k/2^n}]} = \varphi_m|_{[a_{k/2^n}, b_{k/2^n}]}$ for all $m \geq n$.
$\triangleright$ $|\varphi_m(x) - \varphi_{m-1}(x)| \leq \frac{1}{2^m}$ for all $m \in \mathbb{N}$, $x \in [0,1]$.

From the above construction it is immediate that the φ_m uniformly converge giving a limit-function φ such that $\varphi|_{[a_{k/2^n}, b_{k/2^n}]}$ is constant for all binary fractions $\frac{k}{2^n} \in [0,1]$. This construction also gives that the node-system $\{a_{k/2^n}, b_{k/2^n} \mid n \in \mathbb{N}$, k odd, $\frac{k}{2^n} \in (0,1)\}$ is dense everywhere apart from the interior points of the intervals $[a_{k/2^n}, b_{k/2^n}]$. From that we conclude that φ cannot have other intervals on which it is constant apart from the above, and, in addition, that u cannot have any constant intervals that had been integrated into our node-system as one of our intervals $[a_{k/2^n}, b_{k/2^n}]$. Hence φ is the desired function that by $u \circ \varphi^{-1}$ provides a parametrization without constant segments for our path u, as claimed above. □

§2. Construction of the synchronous parametrization of our homotopy combing

2.1 Introductory Remark: The task of defining our synchronous parametrization is lengthy and will take the entire section. Some auxiliary objects have to be defined before, such as

(i) the function φ_x and
(ii) a system of nodes on which φ_x is defined before it is extended to the entire unit interval.

Hence several paragraphs of this section will be devoted to extend the definitions of these auxiliary objects and to prove properties of them that are prerequisites for the next step of definition. The remainder of this introductory remark is devoted to give an overview over the most important steps that it will take to define our synchronous parametrization:

By 1.7/1.3 we may assume that our path is given with some parametrization $u : [0,1] \to X$ that is free of constant segments and that the associated word sequence ω is also given and is reduced (for uniqueness cf. D.7.9/2.16). In 2.15(1) the desired synchronous parametrization v will be defined as $u \circ \varphi^{-1}$, hence the transition function φ is what has to be defined essentially in the following. φ will be defined as a sum $\varphi_x + \varphi_y$, where the summands φ_x and φ_y will be defined by essentially considering the behaviour of u with respect to the x-coordinate and to the y-coordinate, only. Recall that X was defined as a subset of $\mathbb{R}^2$, hence we can write $u(t) = (u_1(t), u_2(t))$ for denoting the according coordinate decomposition of u. Thus we are left with the task of defining φ_x and φ_y. This will be done by interpolating some original setting on a discontinuous subset of nodes, and these nodes will be drawn from the corridor exit points as they are defined by D.10.5(ii) from the associated word sequence ω of u. See 3.6 for the reasons why we do not consider the corridor entrance points in this context. Hence we will start the following construction by an accordingly indexing of the nodes. Of course will the functions $\varphi_x, \varphi_y, \varphi$ and the nodes depend on the given path u and the associated word sequence ω; however for keeping our notation simple we will not permanently recall this

fact by attaching an appropriate index, other than we will do for the corresponding length of the parameter interval of the synchronous parametrization ("b_v", cf. 2.15(2)/ A.5.5). In the following paragraphs we are performing this programme step by step:

2.2 Definition of our node system (cf. 2.1(ii)):
Let k_i and m_i be derived by taking the number of vertical and of horizontal letters of the ith word of ω, respectively, and by adding 1 to each of these numbers. Then we let $a_{i,j}$, $b_{i,j}$ be the parameter value that by D.10.5(ii)/C.5.17 is associated with the corridor exit point of u that corresponds to the jth vertical or horizontal letter of the ith corridor, respectively. Hence on the unit interval [0,1] which is identified with the parameter domain of u for our purpose, two systems of nodes $(a_{i,j})_{i\in\mathbb{N},1\le j\le k_i-1}$ and $(b_{i,j})_{i\in\mathbb{N},1\le j\le k'_i-1}$ are defined.
We always want 0 and 1 to be nodes (even if they are not corridor exit points), hence we in addition let $a_{i,0} = b_{i,0} = 0$ for all i and $a_{i,k_i} = b_{i,m_i} = 1$ for all i.

2.3 Properties of the node-system as defined in 2.2:
The purpose of this item is to note that in any case we have for all $i \in \mathbb{N}$ that $a_{i,0} < a_{i,1} < a_{i,2} < \ldots < a_{i,k_i-1} < a_{i,k_i}$ and that $b_{i,0} < b_{i,1} < b_{i,2} < \ldots < b_{i,m_i-1} < b_{i,m_i}$, since

- ▷ $a_{i,1} > 0$ and $b_{i,1} > 0$ by 1.4(i).
- ▷ The rightmost "<"-relations follows, since if a_{i,k_i-1} or b_{i,m_i-1} should be 1, they would be associated to a truncatable letter of ω (cf. D.10.7(iii)). This would contradict to ω being reduced.
- ▷ The remaining "<"-relations follow since by geometry it is impossible that at the same point a path can exit two corridors of the same basic direction simultaneously, since vertical corridors of a fixed corridor lattice are always disjoint and since, if the same corridor should be involved in adjacent acts of crossing, these acts of crossing have to be performed in inverse direction so that the exit points lie on opposite boundaries. On the other hand, each node $a_{i,j}$ reappears for any higher first index according to D.10.7(ii), and this fact will be used in the iterative definition of φ_x in 2.4.

2.4 Definition of φ_x (cf. 2.1(i)):
The purpose of this item is to define φ_x on the node-system $(a_{i,j})_{i\in\mathbb{N},1\le j\le k_i}$. The zero-step of the induction is given by requiring that $a_{i,0} = 0$ for all i and that $s_{0,j} = 1$ for all relevant j. For the inductive step of this definition we may assume that for given indices i and j the values $\varphi_x(a_{k,m})$ are already defined for all m and $k < i$, and for $k = i$ and $m < j$. Additional assumptions of the induction hypothesis are that for the same index pairs (k, m) we have that

(1) $\qquad s_{k,m} > 1$ and that, if $m < m'$, then $\varphi_x(a_{k,m}) < \varphi_x(a_{k,m'})$.

As the first step of our now defining $\varphi_x(a_{i,j})$, let j', j'' be the biggest/smallest index which is smaller/bigger than j and for which $\varphi_x(a_{i,j'(')})$ has already been defined by the

mechanism described by the last sentence of 2.3, respectively. Then we let

$$\begin{aligned}\varphi_x(a_{i,j}) = \varphi_x(a_{i,j-1}) - \tfrac{1}{s_{i,j'}} \cdot \max\{u_1(a_{i,j-1}), u_1(a_{i,j})\} + \\ \tfrac{1}{s_{i,j'}} \cdot \min\{u_1(a_{i,j-1}), u_1(a_{i,j})\} + \tfrac{2}{s_{i,j'}} \cdot \max\{u_1(t) \mid a_{i,j-1} \le t \le a_{i,j}\} \\ -\tfrac{2}{s_{i,j'}} \cdot \min\{u_1(t) \mid a_{i,j-1} \le t \le a_{i,j}\}\end{aligned} \tag{2}$$

where the $s_{i,j'}$-value is uniquely determined by the requirement that Formula (2) should also give a true relation for $j = j''$, as it will become clear by the following explanations: In this context observe that $s_{i,j'}$ is the same, no matter which of the $j \in \{j'+1, j'+2, \ldots, j''-1\}$ we put in (2). For these j-values (2) gives the definition for $\varphi_x(a_{i,j})$ modulo $s_{i,j'}$. Now take these equations obtained by substituting all $j \in \{j'+1, j'+2, \ldots, j''-1, j''\}$ into Formula (2) and add them. The unknown values $\varphi_x(a_{i,j})$ $(j \in \{j'+1, j'+2, \ldots, j''-1\})$ cancel and only the known values $\varphi_x(a_{i,j'})$ and $\varphi_x(a_{i,j''})$ remain. Hence what is obtained by having added up all these equations contains only one unknown, namely $s_{i,j'}$. A closer look at this equation gives that it is of type

$$\varphi_x(a_{i,j''}) = \varphi_x(a_{i,j'}) + \frac{\text{some positive value}}{s_{i,j'}},$$

and hence, since by induction hypothesis (1) $\varphi_x(a_{i,j'}) \le \varphi_x(a_{i,j''})$, it can always be fulfilled by a unique positive value of $s_{i,j'}$. By having noted that our additional requirement on Formula (2) uniquely determines $s_{i,j'}$, the inductive definition of $\varphi_x(a_{i,j})$ is complete. For future use we take the opportunity and let

$$s_{i,j'} =: s_{i,j'+1} =: s_{i,j'+2} =: \ldots =: s_{i,j''-1}. \tag{3}$$

2.5 Analysis of the essential summand $\Delta_{i,j}$ in the definition of φ_x (cf. 2.4(2)): This summand was:

$$\begin{aligned}\Delta_{i,j} := -\max\{u_1(a_{i,j-1}), u_1(a_{i,j})\} + \\ \min\{u_1(a_{i,j-1}), u_1(a_{i,j})\} + 2 \cdot \max\{u_1(t) \mid a_{i,j-1} \le t \le a_{i,j}\} \\ -2 \cdot \min\{u_1(t) \mid a_{i,j-1} \le t \le a_{i,j}\}.\end{aligned} \tag{1}$$

For that purpose we hereby agree that exceptionally within this Paragraph 2.5 we want to argue as if all corridor exit and entrance points would have in 2.2 given rise to nodes, since in this case simpler geometric pictures can be obtained. In order to understand the behaviour of the $\Delta_{i,j}$ we start by considering the following simple geometrical situations:

(1.) u performs a smooth monotonous crossing of a corridor or of a gap-line between two adjacent corridors. In this case $\Delta_{i,j}$ just gives the width of the corridor which is $\frac{1}{3^i}$ by B.1.1. However, we would have got the same result if u_1 had been far from monotonous, since the above result only depends on the facts that $u(a_{i,j})$ and $u(a_{i,j-1})$ lie on certain adjacent boundary lines, that $a_{i,j}$ and $a_{i,j-1}$ do correspond to letters of $\mathcal{N}_i$ (i.e. that u in a sufficiently big neighbourhood of $a_{i,j}$ and $a_{i,j-1}$ performs a completed act of crossing a corridor and so gives rise to a letter of ω_i), and that u does not intersect other boundary lines of vertical corridors of $\mathcal{N}_i$ on the interval $[a_{i,j-1}, a_{i,j}]$. Hence, even if u_1 was far from monotonous on $[a_{i,j-1}, a_{i,j}]$, we would

have obtained the same value for φ_x. This is what later causes the desired speed-up phenomenon (cf. A.4.11): v will have a parameter interval of the same length for the corresponding act of crossing, no matter whether it is monotonous or whether its first coordinate goes over various maxima or minima.

(2.) Let us now consider the situation where u performs a "simple return", i.e. $u(a_{i,j})$ and $u(a_{i,j-1})$ lie in the same boundary line of a corridor, and u firstly goes into the adjacent region, but then it turns around and comes back before having touched some other parallel boundary line. In this case $\Delta_{i,j}$ is twice the width of the range of x-values over which u has passed in this segment. Having obtained the factor "two" may be understood as if in some sense our Formula 2.4(2) acknowledges the fact that u had to pass at least twice over the corresponding segment. This does not contradict to the speeding-up phenomena described before in (1.), we just have to be aware that certain phenomena of non-monotonous behaviour, even if they do not immediately give rise to corridor entrance and exit point in the sense of (1.), are detected by our Formula. However, this effect is bounded, since by the four summands of (1) an arbitrary number of maxima and minima can certainly not be detected.

(3.) Indeed, Formula (1) should be understood as a continuous extrapolation of the two special cases described before, giving a sensible value in any case and being sufficient to cause the desired speeding-up phenomena. Note that maxima and minima can lie above or below the corridor exit- and entrance points. This can happen if the range of x-values spanned by $[a_{i,j-1}, a_{i,j}]$ is that of a gap-line between two corridors and if there should exist some subsegment of $u_1|_{[a_{i,j-1},a_{i,j}]}$ that goes into some of the corridors adjacent to that gap-line without performing a completed act of crossing. In that case one of the last two summands of Formula (1) will find such a maximum or a minimum above or below that x-value that describes the corridor exit- or entrance-point. However in any case $\Delta_{i,j} \geq 0$, since we have that

$$\begin{aligned} &\max\{u_1(t) \mid a_{i,j-1} \leq t \leq a_{i,j}\} \leq \\ &\qquad \max\{u_1(a_{i,j-1}), u_1(a_{i,j})\} \leq \min\{u_1(a_{i,j-1}), u_1(a_{i,j})\} \\ &\qquad\qquad \leq \min\{u_1(t) \mid a_{i,j-1} \leq t \leq a_{i,j}\}. \end{aligned} \tag{2}$$

(4.) Another way for how to understand Formula (1) is that it gives a lower estimate of the length of the path $u_1|_{[a_{i,j-1},a_{i,j}]}$, namely the best estimate that someone could deduce who is only told the u_1-values at the beginning and at the end of the interval of consideration, and the maximal and minimal values of that segment.

2.6 Proposition: *As a consequence of 2.5(4.) which is independent of the assumption as set at the beginning of 2.5, we get that* $1 \leq s_{i-1,m} \leq s_{i,j}$ *if* $[a_{i-1,m-1}, a_{i-1,m}] \supset [a_{i,j-1}, a_{i,j}]$.

Proof: Let j', j'' be defined as in 2.4 and recall that in the course of the inductive definition 2.4 the sum $\Delta_{i,j'} + \Delta_{i,j'+1} + \Delta_{i,j'+2} + \ldots + \Delta_{i,j''-1}$ has to be computed. This sum is some lower estimate for the length of $u_1|_{[a_{i,j'},a_{i,j''}]}$ that is deduced from knowing the u_1-values at various intermediate nodes and at various local maxima and minima. This information in particular includes $u_1(a_{i,j'})$, $u_1(a_{i,j''})$ and the u_1-values

at the absolute maxima and minima. $a_{i,j'}$ and $a_{i,j''}$ have been adjacent nodes for the first index $i-1$, and the corresponding $s_{i-1,m}$-value had been adjusted on the lower estimate for the length of $u_1{\restriction}_{[a_{i,j'},a_{i,j''}]}$ based on the latter information only. Since for the first index i we have more information, we get an estimate which is as least as high as the one before, and hence we need a denominator $s_{i,j'}$ which cannot be smaller as the corresponding $s_{i-1,m}$-value we had before. If then we recall that at the beginning of 2.4 we had defined all $s_{1,?}$-values to be one, we inductively get that all $s_{i,j}$ are bigger or equal than one. □

2.7 Getting rid of the assumption of 2.5:
If now we undo our incorrect assumption as set for our Analysis 2.5 and skip to considering the geometry as it is really considered in the course of our construction of the synchronous parametrization in 2.2–2.3/2.7–2.15(iv), the situation is not much more complicated than described above: We just have to be aware that the phenomenon of a "simple return" is not possible, since two adjacent corridor exit points can never be on the same boundary line and a "smooth passage" has always to involve one gap-line between corridors and one of the adjacent lines of corridors. In fact, apart from the first and the last segment, any possible behaviour of our path falls in one of the two following categories:

(i) The "(smooth or generalized) passage", where this phrase is reserved for the following situation: The corridor exit point $u(a_{i,j})$ is found on one boundary line of a line of corridors and the preceding one $u(a_{i,j-1})$ found on the neighbouring line of corridors, so that u has crossed at least the latter corridor line and the gap-line in between. The passage is considered to be "smooth" if the boundary line that contains $u(a_{i,j-1})$ has not been intersected by the corresponding segment of u and "generalized" otherwise.

(ii) The remaining possibility is called "return" and is given when $u(a_{i,j})$ and $u(a_{i,j-1})$ are contained on opposite boundary lines of the same line of corridors. This happens if u, after having completed an act of crossing at the exit point $u(a_{i,j-1})$ turns around, enters the corresponding line of corridors again and performs an act of crossing in the reversed direction. Again we might (but will not) attach additional attributes to make some distinction whether u_1 has only two monotony intervals, or whether it has more, and whether u, before turning around, enters the neighbouring line of corridors, or whether, if $a'_{i,j}$ denotes the parameter value of the entrance point that corresponds to the exit point $a_{i,j}$, the segment $u{\restriction}_{[a_{i,j},a'_{i,j}]}$ does or does not cross the corridor boundary that contains the points $u(a_{i,j-1})$ and $u(a'_{i,j})$.

By simply asking how far a path can get after exiting a certain corridor without having to perform a completed act of crossing a (i.e. the same or another) parallel corridor, we get that the possibilities described above are the only ones.

2.8 Estimates for the $\Delta_{i,j}$-summands as defined in 2.5(1):
They are all now immediate corollaries from 2.5(1.)–2.7, and they all follow by adding up the width of corridors or of gap-lines (cf. 2.5(4.)) that our path u in the according segment may cross or must cross. The factors "2" of Formula 2.5(1) have, of course, to

be taken into account, and the fact that the span of x-values $|u_1(a_{i,j}) - u_1(a_{i,j-1})|$ is to be deducted again. Then we get:

(i) If according to 2.7(i) $u_{|[a_{i,j-1},a_{i,j}]}$ is a (possibly generalized) smooth passage, we have $\frac{2}{3^i} \leq \Delta_{i,j} \leq \frac{4}{3^i}$.

(ii) If $u_{|[a_{i,j-1},a_{i,j}]}$ is a return, we have $\frac{1}{3^i} \leq \Delta_{i,j} \leq \frac{5}{3^i}$.

(iii) If $u_{|[a_{i,j-1},a_{i,j}]}$ is a segment where either $a_{i,j-1} = 0$ or $a_{i,j} = 1$, then we have $0 \leq \Delta_{i,j} \leq \frac{6}{3^i}$.

We can actually get the nominator $6 = 2 \cdot 3$, since the factor "2" occurs in Formula 2.5(1), and the range of x-values as taken on by our segment of u can comprise three lines, namely two adjacent lines of corridors and the intermediate gap line; and in case that the according segment of u contains the start- or the endpoint u we cannot rule out that $u_1(a_{i,j}) = u_1(a_{i,j-1})$, and hence in that case the difference between that start- and the end value of the u_1-segment would not be deducted in Formula 2.5(1). On the other hand we can also get the nominator zero, namely in case of a constant path or of a path that is constant with respect to some coordinate.

By Definition 2.4 and 2.5(1)/2.5(2) a monotone function φ_x is defined on some possibly but not necessarily dense set of nodes. We now proceed with the next extension step:

2.9 Extension of the definition of φ_x to an everywhere defined but possibly multivalued function:

We let

$$(1) \qquad \varphi_x(\{t\}) := \begin{cases} \{\varphi_x(a_{i,j})\} & \text{if } t = a_{i,j} \text{ for suitable } i \text{ and } j \\ [\sup\{\varphi_x(a_{i,j}) \mid a_{i,j} < t\}, \inf\{\varphi_x(a_{i,j}) \mid a_{i,j} > t\}] & \\ & \text{in case } t \text{ coincides with none of the } a_{i,j}. \end{cases}$$

Of course, it must be the next step of the proof-programme of this section to show that this definition actually gives an ordinary and continuous function. The following proposition which is an adapted version of the ε-δ-criterion for that monotone and possibly multi-valued situation will be of help for that purpose:

2.10 Proposition: *For φ_x as defined in 2.9 and each $s \in \varphi_x(\{t\})$ and each $\varepsilon > 0$ there exists $\delta > 0$ such that*

(a) *each of the nodes $a_{i,j}$ with $a_{i,j} \in (t - \delta, t]$ satisfies $\varphi_x(a_{i,j}) > s - \varepsilon$ — and such that*

(b) *there exist nodes $\varphi_x(a_{i,j})$ with $a_{i,j} \leq t$ and $\varphi_x(a_{i,j}) \in (s - \varepsilon, s]$ — and such that*

(c) *for each of the nodes $a_{i,j}$ with $a_{i,j} \in [t, t + \delta)$ we have that $\varphi_x(a_{i,j}) < s + \varepsilon$ — and such that*

(d) *there exist nodes $\varphi_x(a_{i,j})$ with $a_{i,j} \geq t$ and $\varphi_x(a_{i,j}) \in [s, s + \varepsilon)$.*

Proof: Even though for symmetry reasons only the Claims (a) and (b) need to be shown, the proof of this criterion will require some discussion in separate cases and thus be not that immediate. Hence we will postpone it to 2.12–2.14 and now demonstrate how this proposition does help us with the main proof-programme of this section:

2.11 Proposition: *φ_x, as defined in 2.4(2), is a well-defined univalent continuous and monotone function $[0, 1] \to [0, b_u]$. In addition, the values that φ_x takes on the node system as originally defined in 2.2 are dense.*

Proof: Monotony follows from the monotony on the interpolation nodes (cf. 2.5(1)&(2)) and from Definition 2.9. The univalence-, continuity-, and density claims are then immediate consequences of the ε-δ-criterion 2.10. In this context observe that even if there should exist subintervals of $[0,1]$ that do no intersect $\{a_{i,j} \mid i \in \mathbb{N},\ 0 \leq j \leq k_i\}$, nevertheless the monotony demand uniquely determines the value of φ_x. Of course, such an interval will then become an interval where φ_x is constant. Having proven that φ is univalent, it is also well-defined as a function. □

2.12 Proof of Proposition 2.10(b) in the case where t is none of the nodes and $u_{1\restriction[0,t]}$ is not constant:
Let $t' < t$ with $u_1(t') \neq u_1(t)$. Since our corridor-network $\mathcal{N}$ is arbitrarily fine, u has to cross some vertical corridors on its way from $u(t')$ to $u(t)$, and hence there also exists some node $a_{i_1,j_1} \in [t',t)$ with $u_1(a_{i_1,j_1}) \neq u_1(t)$. Assume that a_{i_1,j_1} is that last node before t with its first index smaller than or equal to i_1. By monotony of φ we have that $\varphi_x(a_{i_1,j_1}) \leq s$. However, we cannot have "=", since from $u_1(a_{i_1,j_1}) \neq u_1(t)$, we know that $u_{1\restriction[a_{i_1,j_1},t]}$ is not constant so that some node $a_{i',j'} \in (a_{i_1,j_1},t)$ exists that by monotony of φ_x and 2.8 permits the following estimate:

$$\varphi_x(a_{i_1,j_1}) < \varphi_x(a_{i',j'}) \leq \sup_{\tau \leq t}(\varphi_x(\tau)) \leq s.$$

In the following we want to construct a sequence $(a_{i_1,j_1}, a_{i_2,j_2}, a_{i_3,j_3}, a_{i_4,j_4}, \ldots)$ with strictly monotonously increasing first indices by a systematic choice of such nodes and to show that in the φ_x-values of these nodes converge to s.
Now a_{i_2,j_2} can be found by using the same arguments as for the construction of a_{i_1,j_1}: our path u on its way from $u(a_{i_1,j_1})$ to $u(t)$ has to cross some of our corridors. Hence we can define i_2 to be the first lattice index for which this happens and we let a_{i_2,j_2} be the parameter value of the last exit point before t. We now seek an estimate for the ratio

$$\frac{s-\varphi_x(a_{i_2,j_2})}{s-\varphi_x(a_{i_1,j_1})} = \frac{s-\varphi_x(a_{i_2,j_2})}{s-\varphi_x(a_{i_2,j_2})+\varphi_x(a_{i_2,j_2})-\varphi_x(a_{i_1,j_1})} \tag{1}$$

that permits us to conclude that $\varphi_x(a_{i_\nu,j_\nu}) \overset{\nu\to\infty}{\longrightarrow} \varphi_x(t)$. For that purpose we recall that $a_{i_1,j_1} = a_{i_2,j_2-k} < a_{i_2,j_2-k+1} < a_{i_2,j_2-k+2} < \ldots < a_{i_2,j_2-1} < a_{i_2,j_2} < t$ and that none of the nodes $a_{i_2,j_2-k+1}, a_{i_2,j_2-k+2}, \ldots, a_{i_2,j_2-1}, a_{i_2,j_2}$ belongs to a system of nodes with a smaller first index than i_2. Hence

$$\varphi_x(a_{i_2,j_2}) - \varphi_x(a_{i_1,j_1}) = \frac{\Delta_{i_2,j_2-k} + \Delta_{i_2,j_2-k+1} + \ldots + \Delta_{i_2,j_2-1}}{s_{i_2,j_2-k}}. \tag{2}$$

Thus by 2.8 we have

$$\varphi_x(a_{i_2,j_2}) - \varphi_x(a_{i_1,j_1}) \geq \frac{k \cdot \frac{2}{3^i}}{s_{i_2,j_2-k}}. \tag{3}$$

and

$$s - \varphi_x(a_{i_2,j_2}) \leq \varphi_x(a_{i_2,j_2+1}) - \varphi_x(a_{i_2,j_2}) \leq \frac{\frac{6}{3^i}}{s_{i_2,j_2-k}}. \tag{4}$$

Now elementary calculus gives that the maximal value of the ratio (1) is achieved if the difference $s - \varphi_x(a_{i_2,j_2})$ is maximal and if the difference $\varphi_x(a_{i_2,j_2}) - \varphi_x(a_{i_1,j_1})$ is minimal; and hence we get by using $k \geq 1$ as a trivial estimate that $\frac{6}{8} = \frac{3}{4}$ is an upper estimate of the value of Formula (1). Since $(\frac{3}{4})^m \to 0$ for $m \to \infty$ we see that there have to exist nodes $a_{i,j}$ with $a_{i,j} < t$ such that their φ_x-values lie arbitrarily close to s. □

2.13 Proof of Proposition 2.10(b) in the remaining cases:
Note that $\{a_{i,j} \mid a_{i,j} \leq t\}$ is not empty, since it always contains $0 = a_{i,0}$. Further observe that the claim is evident in the case that one of our nodes $a_{i,j} = t$, since then by monotony and definition $s = \varphi_x(a_{i,j}) = \max\{\varphi_x(a_{i',j'}) \mid a_{i',j'} \leq t\}$. Hence we may now assume that t is none of the nodes and by 2.12 that $u_1|_{[0,t]}$ is constant:
In order to get rid of the trivial case right now, assume that u_1 is constant on its whole domain of definition. Then there exist at most four nodes (two by Definition 2.2 and thanks to phenomena as described in C.4.8 there may exist two more); but since u_1 is constant, Formula 2.4(2) gives the same φ_x-value for all these nodes. This makes sure that 2.10(b) holds in the special case.
The same arguments can also be applied if u_1 is not entirely constant, but some additional segments $u_1|_{[0,t']}$ is, and if t lies between zero and the last of the at most three nodes within such an interval of constancy. Hence for the remainder of the discussion of this case we need only to consider the situation that $0 < a_{i,j} \leq t \leq t' < 1$ such that $u_1|_{[0,t']}$ is constant and that $a_{i,j}$ is that last of the nodes within this interval of constancy. In that situation it is of help to observe that it does not need more arguments to prove 2.10(d) than to obtain 2.10(b), and that in particular it takes only the same arguments as Case 2.12 of 2.10(b) to obtain the results 2.10(d) for t. Hence the φ_x-values at the nodes bigger than t accumulate to any $s \in \varphi_x(\{t\})$ and hence $\varphi_x(\{t\})$ can only contain one point, namely $\varphi_x(a_{i,j})$. This gives the result 2.10(b) at t, also. □

2.14 Proof of Proposition 2.10(a):
In principle this is just is a corollary from 2.10(b): If in some neighbourhood of t some node with a too small value of φ_x should be found, then the methods of 2.10(b) suffice to construct a better node $a_{i',j'}$ such that $\varphi_x(a_{i',j'}) > \varphi_x(t) - \varepsilon$. The monotony of φ_x then gives the same estimate for all nodes of $[a_{i',j'}, t]$ (in this context observe that our construction scheme of additional nodes can also be applied if t should be some node, and that, other than for the proof of 2.10(b), this case cannot be ruled out here).
Since according to the above construction any $\delta' \leq t - a_{i',j'}$ satifies the condition of 2.10(a) we let $\delta_1 := t - a_{i',j'}$ and define a δ_2 with the analogous properties in Step 2.10(c) of our proof: Finally, $\delta := \min\{\delta_1, \delta_2\}$ satisfies all desired properties of our criterion 2.11 which hence by now is proved. □

2.15 The remaining steps of the definition of a synchronous parametrization:
(i) As an immediate consequence of the construction of φ_x in 2.1–2.14 (in particular see 2.4(2) and 2.9(1)), we get that φ_x constant precisely on the same segments as u_1 is constant.

(ii) In analogy to φ_x we define a monotonously increasing continuous function on [0,1] named "φ_y" satisfying $\varphi_y(0) = 0$ and that it has constant segments precisely on those intervals where u_2 is constant. The crucial definition formula for φ_y can be obtained from 2.4(2) by replacing all nodes of the system $(a_{i,j})_{i\in\mathbb{N},1\leq j\leq k_i}$ by the corresponding nodes of the system $(b_{i,j})_{i\in\mathbb{N},1\leq j\leq m_i}$, and by substituting u_1 by u_2. The arguments that this definition extends to a continuous function are the same that have been given in 2.11 for φ_x.

(iii) By now letting $\varphi := \varphi_x + \varphi_y$ we obtain a real-valued, monotonous and continuous function defined on $[0, 1]$. This function φ can only have constant intervals where the constant intervals of φ_x and of φ_y intersect. Such intervals would have to be constant intervals of u_1 and of u_2 and hence of u, but since u was assumed to be free of constant intervals, we see that φ cannot have any. I.e. φ is strictly monotone, and hence it bijects [0,1] to its interval of image domains.

(iv) By now we are ready to define the synchronous parametrization for the homotopy combing C_X that was constructed in 1.3. In the preceding steps 2.2–(iii) we have constructed for each path of this homotopy combing represented by some almost arbitrary parametrization u a strictly monotonous continuous function φ such that we can now let

$$v : [0, \varphi(1)] \to X \quad , \quad v := u \circ \varphi^{-1}. \tag{1}$$

Recall that in 2.1 we agreed to resist during the construction of φ to remind ourselves by an additional index that, of course, φ_x depends on u. Accordingly the domain of definition $[0, \varphi(1)]$ may be some different domain for each path in our homotopy combing. However, we now let

$$b_v := \varphi(1), \tag{2}$$

i.e. we attach some name to this parameter value reminding us by a corresponding index that this may be a different value for each path $v \in C_X$.
Since u and φ are free of constant segments, so is each of the v.
We should point out at this place that this is the main reason why we could not get around considering parameter intervals of different length: Of course our homotopy combing has also to contain the trivial path as the path that is associated with the empty word sequence and gives the trivial connection of $(0, 0)$ to itself, and this path cannot be parametrized without constant intervals, except if we use the interval degenerated to one point.

(v) We should mention at this place already, that in the course of future arguments we will have to compute distances such as "$|v_1(t) - v_2(t)|$ for all parameter values t", and for that purpose we consider both paths as being parametrized on $[0, \max\{b_{v_1}, b_{v_2}\}]$, and we have to define an extension for the path with the shorter parameter interval. If $b_{v_1} < b_{v_2}$, we let $v_1{\restriction}_{[b_{v_1}, b_{v_2}]}$ be the constant path taking on the value $v_1(b_{v_1})$. Hence, if this extension has to be performed, the parametrizations v_i cannot be considered as free of constant segments any more.

2.16 Remark: We remark that the synchronous parametrization as defined above in 2.2–2.15(iv) does not depend on the arbitrary parametrization of the the corresponding path as denoted by u in Paragraphs 2.2–2.15(iv). It just depends on the abstract path as described by the associated word sequence ω. This can be easily seen if we recall that essentially the functional values of φ have become the parameter values of our new parametrization v, and these functional values only depend on the fact what corridors our path is going to cross in what order, i.e. on the data given by the word sequence. However, note that for our parametrizations in general also the future behaviour of the path may have its impact (i.e. the question whether the act of crossing some corridors will or will not be completed), and this is why in 3.7 we will not obtain straight away that our homotopy combing is prefix-closed. In addition note that, as it will be discussed in 3.5, the skipping to another word sequence in the same strong equivalence class may also have an impact on the precise parametrization. In this context we will only obtain uniqueness since we have unique reduced firm equivalence classes that are to be used when defining the synchronous parametrization (cf. D.7.9), and since in 2.3–2.15(i)/ 2.15(ii) only the vertical/horizontal letters had been considered, respectively, *it does not matter which representative of the firm equivalence class we had been used in 2.2–2.15(iv).*

§3. The Basic Set-up for the Continuity proof of the Lifting Theorem

Having defined the synchronous parametrization for our homotopy combing, as motivated in B.3.3 the corresponding task to be done is to prove an appropriate ε-δ-criterion for any t for two paths v_1, v_2 that represent relative homotopy classes α_1, α_2 that are homotopically near in the sense of A.5.2(2). As in A.5.2(i) we use P_i to denote the endpoint of a relative homotopy class α_i, and hence we will prefer to talk about pairs "(P_i, α_i)" instead of α_i.

3.1 Remark on the uniformity of the desired continuity:
As it is standard in proofs of continuity, we may regard one of our pairs (P_1, α_1) as "fix", i.e. as being the centre of an according η-neighbourhood, and the other pair (P_2, α_2) correspondingly as "free".
Since we do not want to give a proof for uniform continuity, δ may depend on various properties of the fixed pair (P_1, α_1), in particular also on properties of the parametrization v_1. We will have to make use of such dependence in following.
Of course, since our codomain for the continuity proof, i.e. the sphere S^k is compact, indeed the continuity of discussion must be a uniform one. However, since on the other hand the homotopy combing as a whole is not compact (it has to contain relative homotopy classes of paths which encircle a hole once, twice, three times, four times,... — and such a sequence of paths cannot accumulate to a well-defined path), it turns out to be the most convenient if we proceed as follows: In the crucial continuity proof we argue as if a not necessarily uniform continuous behaviour is to be shown; and whenever we

need uniformity, we can just refer to the compactness of S^k and of $f(S^k)$ and quote the corresponding elementary results of calculus.

Having defined a synchronous parametrization for our homotopy combing and hence a candidate for the lifting $\widetilde{C_X}$ of C_X as defined in 1.3/Sect.§2 (cf. also Theorem A.5.6), it is now time to approach the technical final steps of this paper. We proceed by explaining

3.2 The principle set-up for the continuity proof of our Lifting Theorem B.3.2:

The purpose of the subsequent items 3.3–5.11 is to give the continuity proof for $\widetilde{C_X}$. The construction principle of $\widetilde{C_X}$ has been described in Sect.§B.3 and by having defined our homotopy combing in 1.3 and the corresponding parametrization of all paths in 2.1/2.15, $\widetilde{C_X}$ can now be considered as a well-defined parametrized homotopy combing, i.e. a well-defined function $\widehat{X_{x_0}} \to \widetilde{\mathcal{L}_{x_0}}(X)$. However, other than required in A.5.6(iii), the end-parameter projection may not be continuous according to our present state of knowledge, even not in $\mathrm{Im}(\widetilde{C_X})$ (see 3.9 for how we will handle that problem).

We want to prove continuity by proving a classical ε-δ-criterion. Hence within these paragraphs we consider ε as a given constant, and the same for $(x,\alpha) \in \widehat{X_{x_0}}$ which denotes the place at which the continuity of $\widetilde{C_X}$ shall be proven. In some of the discussions we might also want to include the preimage $z \in S^k$, i.e. a z-value with $f(z) = x$ and $\widehat{f}(z) = (x,\alpha)$, since by A.6.5 we need only to consider the continuity of $\widetilde{C_X}$ on $\widehat{f}(S^k)$. By continuity of f, for any given $\eta > 0$ there exists a δ such that f maps the δ-neighbourhood of z into the η-neighbourhood of x. By A.6.4/3.1 each image point $z' = f(x')$ can be associated with a pair (x',α'). Observe that all pairs that are associated with a z-value taken from our δ-neighbourhood are homotopically $4{\cdot}\eta$-close to each other, since the path giving a homotopically correct connection can always be constructed inside of our η-neighbourhood; we have to add a factor "4", since the diameter is always the double of the radius, and, in addition we have to take into account that the centre when measuring the homotopic distance between our pairs need not to coincide with the centre of the η-neighbourhood. Hence in the course of proving continuity for $\widetilde{C_X}$ we are now left with the task of showing that there exists some η such that any pair (x',α') that is homotopically near to (x,α) with respect to the chosen η-neighbourhood satisfies that $|v(t) - v'(t')| < \varepsilon$ for all $t \in [0,1]$, if v and v' are the paths that are associated with α and α' by 1.3 having parametrizations as they have been constructed in Sect.§2. The corresponding *definition for η will follow now in this paragraph without further motivation, and besides $\delta, \varepsilon, z, x, v, \omega$ and α and the corresponding dashed values it will be used as a standard notation in 3.3–5.11 ($\omega :=$ associated word sequence of v):*

For ε and x as above find some i_0 such that a (5×5)-block of regions of $\mathcal{N}_{i_0}$ still fits into $U(x, \frac{\varepsilon}{2})$, and in addition, that x is contained in or on the boundary of the middle one of our (5×5)-block of regions. The corresponding index i_0 will also become a fixed standard notation in 3.3-5.11. Now recall the notation that in Sect.§2 had to be introduced in order to find the appropriate parametrization $v = u \circ \varphi^{-1} = u \circ (\varphi_x + \varphi_y)^{-1}$. Let

$$s_{i_0} := \max\{s_{i_0,j} \mid 1 \le j \le k_{i_0}\} + \max\{s'_{i_0,j'} \mid 1 \le j' \le m_{i_0}\}. \tag{1}$$

The $s_{?,?}$ have been introduced in 2.4 in the course of defining φ_x and the $s'_{?,?}$ shall denote the corresponding quantities as they had to be computed when φ_y is to be defined. We now find another index $i' > i_0 + 1$ such that the mesh-width of that index, i.e.

$$(\frac{1}{3})^{i'} \text{ is smaller as } \frac{(1/3)^{i_0}}{17 \cdot s_{i_0}}. \tag{2}$$

We choose a (5×5)-block of regions of $\mathcal{N}_{i'}$ such that x falls into the middle one of these regions or on one of the twelve adjacent quadrants. In this context the expression "quadrants" is used if a region of one of our lattices is (2×2)-subdivided by using the vertical and the horizontal mid line, and for that purposes we think of the eight adjacent regions (i.e. the regions that have either a vertex or an edge in common with the middle region) as additionally subdivided into further quadrants. Further we assume that our (5×5)-block of regions is adjusted in such a way that the middle region is either a hole or a crossing region of corridors (and hence all regions that are adjacent to the outside are also holes or corridors). Since our corridor network is arbitrarily fine, for sufficiently high i' such a (5×5)-block of regions with that properties can be found as a neighbourhood of x. Finally we let

$$\eta := \frac{5}{2 \cdot 3^{i'}}, \tag{3}$$

i.e. half of the edge-length of our (5×5)-block. This (5×5)-block shall hence further be called Q_η ("the η-square").

3.3 Definition of the three-phase splitting up of ω that is to be used in the proof that $\widetilde{C_X}$ is continuous (cf. 3.2):
Let ω be as in 3.2 and let κ_2 be the leftmost position sequence of ω so that any letter of ω which is placed to the right of κ belongs to a corridor which has a non-trivial intersection which Q_η. See below for the arguments why κ_2 exists. Similarly we let κ_1^h and κ_1^v be the leftmost position sequences of ω such that each horizontal or vertical letter that is placed to the right of κ_1^h or κ_1^v has a non-trivial intersection with Q_η, respectively, and finally we let $\kappa_1 := \min\{\kappa_1^h, \kappa_1^v\}$. These position sequences define a split-up according to $\omega = \omega_{|[0^{\mathbb{N}},\kappa_1]} \cdot \omega_{|[\kappa_1,\kappa_2]} \cdot \omega_{|[\kappa_2,l^{\mathbb{N}}]}$ which by C.5.17 corresponds to the decomposition $v = v_{|[0,t_1]} \cdot v_{|[t_1,t_2]} \cdot v_{|[t_2,b_v]}$, if t_m denotes the parameter value associated with κ_m by C.5.17. Finally we introduce the abbreviations $v_1 := v_{|[0,t_1]}, v_2 := v_{|[t_1,t_2]}, v_3 := v_{|[t_2,b_v]}$ and regard v_m as that segment of v that corresponds to "Phase m". This subdivision of v into three phases will be essential for the subsequent proof that $\widetilde{C_X}$, as stated in 3.2 gives a continuous homotopy combing.

Well-definedness Proof: We have to use Königslemma-type of arguments (cf. C.5.10) in order to see that $\kappa_2, \kappa_1^h, \kappa_1^v$ and κ_1 are well-defined minima. For that purpose consider the dual split-up picture of ω as described in C.5.5. Mark those vertices in the white graph that satisfy that the associated letter of ω and all (horizontal/vertical) letters that are placed in the same line on the right hand side of the corresponding vertex belong to a corridor intersecting Q_η. Then also mark those yellow vertices which are placed immediately to the left of a marked white vertex. For each of the marked yellow

vertices ask whether they belong to an infinite monotonous edge-path which only leads over marked vertices. Remove the marks from those vertices which do not satisfy this condition, and that way one sees that the desired word sequence $\kappa_?^?$ exists: Construct the corresponding infinite edge-path inductively by in each step finding the leftmost downleading yellow edge-path to a marked edge.
We observe that the following properties are immediate from the definitions:
(i) $\kappa_1 \leq \kappa_2$
(ii) If $\kappa_2 = \ell^{\mathbb{N}}$, then $0^{\mathbb{N}} = \kappa_1 = \kappa_2 = \ell^{\mathbb{N}}$, i.e. we are in the case that ω is the empty word sequence. This property can most easily be seen from geometry: The paths to be considered in the proof that $\widetilde{C_X}$ is continuous are defined in such a way that they end inside Q_η, and hence, if they are not constant, some terminal segment has to be inside Q_η. By C.4.8/C.5.15 this translates to $\kappa_2 < \ell^{\mathbb{N}}$.
(iii) If Phase 2 should not be empty (i.e. if $\kappa_1 \neq \kappa_2$) and if they are not incomparable, then, with letters that are truncatable in $\omega_{|[\kappa_1,\kappa_2]}$ possibly excepted, for one of the basic directions $\omega_{|[\kappa_1,\kappa_2]}$ satisfies that all the corresponding letters are associated with corridors with a non-empty intersection with Q_η, while the $\omega_{|[\kappa_1,\kappa_2]}$-letters of the other basic direction do not satisfy this property. In future proofs we will call the basic direction with the first property the *"Q_η-direction"*, and the other basic direction the *"non-Q_η direction"*. We also introduce the symbol $\mathcal{A}_{Q_\eta}$ for the set of those letters of $\mathcal{A}$ that belong to corridors that intersect Q_η.
(iv) Our word sequence ω' (cf. 3.2) permits the definition of position sequences with similar properties as κ_1 and κ_2 to which parameter values can be associated by C.5.17. They will be denoted by κ_1', κ_2', t_1', t_2', respectively.
(v) We have that $v(t_2) \in \partial Q_\eta$ and that $v'(t_2') \in \partial Q_\eta$. (A proof of (v) can be obtained with similar techniques as the result D.10.8).
We will also use the abbreviations "Q_η-coordinate" and "non-Q_η coordinate" for the x-coordinate and the y-coordinate. The precise definition is given by the requirement that $\{x, y\} = \{Q_\eta,\ \text{non-}Q_\eta\}$ and that a boundary line of a Q_η-directed corridor has a constant non-Q_η coordinate and vice versa.

3.4 Remark on the well-definedness of the terms *"Q_η"- and "non-Q_η"-coordinates* as defined above:
From a mere combinatorial point of view it might seem that our concept of referring to the x- and y-coordinates as "Q_η-" and as "non-Q_η coordinates" and attaching similar attributes to our corridors might not be uniquely defined for a set-up of paths v and v' as described in 3.2, moreover it might appear as if there is some risk that when looking at the paths, v and v' might give different definitions of what "Q_η-" and "non-Q_η" are. However when taking a geometrical point of view it already seems unlikely that we might experience actually such non-uniqueness:
Let (a, b) be the x- and the y-coordinate of the mid-point of Q_η, such that $Q_\eta = [a-\eta, a+\eta]\times[b-\eta, b+\eta]$. Then the thickened lines $[0,1]\times[b-\eta, b+\eta]$ and $[a-\eta, a+\eta]\times[0,1]$ subdivide our basic square into four quadrants. A necessary but not sufficient condition for getting a non-empty Phase 2 is that our path v in its final phase of approaching Q_η does not enter Q_η from one of the corner points, but has spent a segment in one of

the thickened lines before. Of course, if our path does not satisfy this condition, then the attributes "non-Q_η" and "Q_η-" are not defined, but this is irrelevant since with a trivial Phase 2 the claims in the proofs 4.1–5.11 below in which these attributes play a role are then void, also. Hence, if a set-up as described in 3.2 should impose non-unique defintions of the attributes "non-Q_η" and "Q_η-", we will see a situation where v in its approaching of Q_η goes over the other of our thickened lines $[0,1]\times[b-\eta,b+\eta]$ and $[a-\eta,a+\eta]\times[0,1]$ as v' does. It seems very unlikely that homotopically η-near paths that are associated with reduced word sequences can behave like this, since either the corresponding separating quadrant is free of holes so that the reduction mechanism B.2.3/D.3.10 would then push one path into the position of the other, or, if there are obstructing holes, our paths would pass them on different sides, and this would prevent them from being homotopically η-near any more. A precise combinatorial argument that such a situation is ruled out and hence the attributes "non-Q_η" and "Q_η-" are indeed well-defined as long as we stick to considering one of our 3.2-set-ups is implicitly contained in the proof of the subsequent lemma 4.1: In this proof we will point out that ω' can be obtained from ω by firstly attaching to ω another reduced word sequence ω'' that only consists of $\overline{\mathcal{A}_{Q_\eta}}$-letters and then applying the reduction processes as described in D.11.10 to the set-up. Since these reduction processes can clearly not affect the non-Q_η letters of Phase 2 of v, we will that way see that v' cannot approach Q_η from another side than v does.

3.5 Discussion of how the synchronous parametrization responds to phenomena of strong equivalence:

The purpose of this remark is to point out that in principle the definition of our synchronous parametrization (cf. Sect.§2) is not defined invariantly with respect to the phenomena of strong equivalence. It does not respond differently if the associated word sequence is changed by transpositions; recall that this fact had to be used in Sect.§2 already and the corresponding reasons have been explained in 2.16.

However, the situation is entirely different for truncations and cancellations. The reasons why the corresponding different responses do not mess up the overall continuity of the desired homotopy combing $\widetilde{C_X}$ as constructed in 3.2/1.3 (and shown to be continuous in 3.2–5.11) have essentially already been listed in 1.2.

However in order to actually construct examples where a non-continuous response of the synchronous parametrization to strongly equivalent descriptions of the same paths is obtained, one should not start with too simple examples. E.g., one will hardly see a different response when just considering the quadratic parabolas as described in 1.2. The corresponding reasons have been mentioned in 2.5(2.), already: When considering quadratic parabolas only, then, according to the language as developed in 2.7, the additional acknowledgement of a tangency as a point where an act of crossing ends and a new act starts will just cause that what had been a "simple return" before has now changed into two "smooth passages". As it has been stressed in 2.5(2.), the corresponding formulae are strong enough to detect this phenomenon and to acknowledge a simple return by an additional factor "2". Hence in such a case one might not see a different response of our synchronous parametrization, no matter whether one regards

or disregards a certain tangency. On the other hand, if one sets up a similar picture as 1.2 with non-monotonous cubic parabolas and puts such a parabola in such a way into the plane that the local maximum and the local minimum are tangent points to the boundary lines of some horizontal corridor, then, when disregarding these tangencies, our synchronous parametrization will respond to that configuration as if it sees just a smooth passage. On the other hand, if both tangent points were acknowledged, then this parametrization responds as if it sees three smooth passages, and hence the corresponding adjustment of φ_x in the course of finding the synchronous parametrization will change in a non-continuous way.

3.6 Why did we ignore the corridor entrance points in Sect.§2?
Consider a path which ends up inside a corridor, but which has started outside. According to D.10.7(iii) the corresponding letter that acknowledges that partial act of crossing, is truncatable. Since in the course of our construction in Sect.§2 (cf. 2.2) the nodes for defining the functions φ_x and φ_y had been associated with the letters of the associated word sequence, if corridor entrance points would also give rise to nodes, a decision whether or whether not to truncate such a letter would cause that the parameter value that corresponds the entrance point of this corridor does not or does appear as a node. As pointed out in 3.5, the additional taking up or dropping of some nodes in the course of constructing the synchronous parametrization (cf. 2.2–2.15(iv)) might easily cause a non-continuous response of the parametrization.
The phenomenon described above might matter in the 3.5-sense, however the standard context they should be irrelevant for the paths of our homotopy combing, since the homotopy combing was based on considering reduced representatives, and in reduced representatives all possible truncations are to be performed. On the other hand, there are also exceptional situations, e.g. if one considers one path which after having crossed some corridor ends in an ε-neighbourhood of the exit point of this crossing. If now a second "shortened" version of this path is also considered so that the shortened path fails to complete the crossing, we clearly get two homotopically near paths (cf. A.5.2(2)). If corridor entrance points would give rise to nodes, then for the path that has completed the crossing, the entrance and the exit both would give nodes, while for the path that has failed to complete the crossing none of them would give a node. According to 1.2(iii) the node at the exit point does not cause any harm; we will see that we can deal with such kinds of nodes in 4.14–5.7 successfully, however the additional taking up of the node at the entrance point could cause all phenomena of a non-continuous response as described in 3.5. In the context of some attempt to prove continuity one might already be forced to treat the node at the entrance point as lying far away from the end-point of our two paths, in particular if the corresponding corridor belongs to a lattice with a small index. However, any adjustments which are made on that level are not to be undone by the forthcoming interpolation steps of Sect.§2, and hence the non-continuous response would affect the continuity of the synchronous parametrization. For that reasons we resisted to use the corridor-entrance points as nodes in Sect.§2.

3.7 Is our homotopy combing prefix-closed?
An ordinary combing (cf. A.5.5(i)) is called *"prefix-closed"* (cf. [ECHLPT].(Def.1.2.10)),

if it satisfies that wherever two combing paths meet (i.e. if there exists some t, s with $c_x(t) = c_y(s)$), then $c_x|_{[0,t]}$ and $c_y|_{[0,s]}$ are the same abstract path. A suitable adaptation of this property for a homotopy combing would only require such a coincide of the initial subsegments of two paths in those cases where we have that $c_x|_{[0,t]}$ and $c_y|_{[0,s]}$ are relatively homotopic. The following paragraph is devoted to show that *our homotopy combing as defined in 1.3 is prefix-closed in the latter sense.*

By using the properties that have been shown so far, this can be seen fairly quickly: Consider the word sequences ω and η that are associated to two combing paths c_x and c_y that meet as above, respectively, and use C.5.17/C.5.3(ii) to split up both of these word sequences into $\omega = \omega'\omega''$ and $\eta = \eta'\eta''$, so that we have in particular two word sequences ω' and η' that describe $c_x|_{[0,s]}$ and $c_y|_{[0,t]}$, respectively. Since we are assuming that these paths are relatively homotopic to each other, by D.12.1 the word sequence ω' and η' are weakly equivalent. If the paths should not agree, then ω' and η' can by D.12.3 not be strongly equivalent, i.e. in particular not be firmly equivalent. Hence by D.7.9 at most one of them can be reduced. Thus to the other of them, say to ω', reduction processes can still be applied that reduce ω' to a non-strongly equivalent representative. Since the existence of firmly equivalent reduced representatives was by D.7.9/D.7.14 given for all our equivalence classes of D.3.6 and hence we have the freedom to exclude truncation phenomena (they could only give strong equivalence relations, either) by considering ω' within its semi-free equivalence classes, only, the same shortening transformations as for ω' can also be applied to the entire word sequence ω (reduce ω' and keep the rest fix). Since it is only a question of the nesting depth (cf. B.2.6/B.2.9) whether the corresponding transformations give a weak or strong equivalence, we see that our transformations when imposed to the entire ω can also not fulfil the conditions of strong equivalence, but are shortening operations on the other hand. Hence ω could not have been reduced; and this is a contradiction since according to 1.3 our combing paths were always associated with reduced word sequences.

The arguments of the above paragraph give that our homotopy combing is prefix-closed — when view with respect to the abstract paths. *However, it is not prefix-closed with respect to the synchronous parametrization as constructed in Sect.*§2, and the remainder of 3.7 is devoted to explain why this happens:

In formulae, we claim that one is likely to find two combing paths c_x and c_y agree as abstract paths for some initial subsegment(s) (i.e. we have that $c_x|_{[0,s]} = c_y|_{[0,t]}$, but that for $s' < \min\{s, t\}$ we could perfectly have $c_x(s') \neq c_y(s')$. In order to see that this can actually happen, recall the interpolation process that in 2.4–2.15(iii) gave the φ_x-values for the corridor exit-points. By 2.15(iv) the φ-values gave rise to the parameter values of the synchronous parametrization. Now a linear interpolation gives a mixture of the effects that can be seen at either of the ends of the interval. Consequently, if our two paths are still in their initial phase where they agree as abstract paths (cf. 3.3) and if they have passed the last corridor exit of the first lattice in that phase already so that they will fall apart before they reach the next corridor exit on the first level, then the adjustment of φ_x- and φ_y-values is most likely to be different for both paths at their latter corridor exit points; and by linear interpolation it should then come out differently in the entire

interval as described in the preceding sentence, even before their falling-apart point. According to the relation between the φ-function and the synchronous parametrization as stressed before, it should now be evident that our homotopy cannot be considered as prefix-closed with respect to the parametrization that we have constructed in Sect.§2.

3.8 Are the lengths of the parameter domains bounded for a synchronous parametrization as defined according to Sect.§2?
Note that according to the present definition the range of b_v-values in a homotopy combing for one of our spaces X may be unbounded. For, if the single horizontal line and the single vertical line of corridors that we have on the first lattice actually intersect in the hole $[\frac{1}{3},\frac{2}{3}]\times[\frac{1}{3},\frac{2}{3}]$, then the b_{v_n}-values of a sequence of paths $(v_n)_{n\in\mathbb{N}}$ where the nth path goes around n times will linearly increase with n. However, a simple technical adjustment can actually give such a synchronous parametrization of our homotopy combing where the range of values is naturally bounded:
As already pointed out in A.3.4 and A.3.9(2), it might by suitable to think that our indices of corridor lattices actually start with "0", where the index "0" stands for the empty lattice. Recall that $b_v=\varphi_x(1)+\varphi_y(1)$ according to our inductive process 2.4 (cf. 2.15(2)/2.15(iii)), where this value is already adjusted in the first step of this inductive process. If this first step should already be performed for the level index "0" and hence should be based on an empty corridor lattice, we will not have any nodes apart from "0" and "1", and hence we do not iteratively have to compute sums according to Formula 2.5(1), but just once for each of the coordinates. Taking into account what in this case has to cancel in the alternate sum 2.5(1) and that we have to compute such a sum for φ_x and for φ_y, we see that "4" is the utmost value that could be taken on by any b_v, if they are adjusted in that way.

3.9 Remark: *On how to achieve continuity of $v\mapsto b_v$ on Im($\widetilde{C_X}$)* (cf. A.5.6(iii)):
In order to achieve this claim we offer in principle two possibilities:
(i) to imitate the continuity proof below in Sections §4 & §5. By in particular recalling that the bounding parameter values (cf. 2.15(2)/2.15(iii)) are defined as $\varphi(1)=\varphi_x(1)+\varphi_y(1)$ it should be clear that the desired ε-δ-estimate for a continuity of b_v is fairly similar to the estimates of 5.1 and can in principle be obtained by mixing these arguments in a new way.

However, this problem can be technically more easily be handled by just circumventing it:
(ii) Observe that there was only one place where we actually needed the continuity of $v\mapsto b_v$ and that was to conclude the boundedness of the b_v on $\mathrm{Im}(\widehat{f}\circ\widetilde{C_X})$. Now 3.8 already indicated that this boundedness can be achieved by a mere trick: By a slight change of the definitions of Sect.§2 we could have made the b_v-values overall bounded on our homotopy combing. However, actually there is not any need for such a change of definition, since, according to the discussion of 3.8 there is only one single effect which could make the $(v\mapsto b_v)$-function unbounded on $\mathrm{Im}(\widehat{f}\circ\widetilde{C_X})$, namely if f wraps its preimage S^k around $[\frac{1}{3},\frac{2}{3}]\times[\frac{1}{3},\frac{2}{3}]$ infinitely many times. However, a continuous f cannot do this, as can be easily seen by constructing an appropriate covering where

each covering set cannot contain a full wrap and by reducing such a covering to a finite one by using the Heine-Borel-property for the compact preimage S^k.

Hence we actually need not worry about this condition of our Combing Theorem A.5.6(iii)).

3.10 Outline of the continuity proof of our Lifting Theorem:

In the subsequent continuity proof of $\widetilde{C_X}$ which is covering both Sections §4 and §5, by 3.2 we may restrict our considerations to two relative homotopy classes that my be represented by paths that are homotopically near with respect to Q_η. Consequently these homotopy classes may be represented by two paths u and u' that coincide for some initial segment and only fall apart after they have reached Q_η and will remain in Q_η till the end. By having obtained almost unique reduced representatives for weak equivalence classes in D.7.9, the actual combing paths v and v' can be obtained by reading word sequences from u and u', reducing them in their weak equivalence classes, and the v and v' are obtained by realizing these reduced representatives by paths again. v and v' may fall apart outside Q_η already, but if this happens it is due to the above described reduction process, and this will give some restrictions to what behavior v and v' may have if they should fall apart before reaching Q_η, already. More precisely, we will think of subdividing v and v' in three phases as follows:

(1.) Phase One where v and v' still coincide,

(2.) Phase Two which starts when they fall apart and ends when they enter Q_η for not leaving it any more, and

(3.) Phase Three, the final segment that both path spend entirely in Q_η.

Of course, Phase 2 could just be empty, but since this is the less difficult case and may be treated as a special case of the situation with a non-empty Phase 2, we will hardly care about this case. While the above definition of three phases is just a definition, the non-trivial claim that comes with it will state that *in a non-empty Phase 2 our paths are bound by monotony conditions.* More precisely, in the Q_η-coordinate both paths go monotonously and in the same direction over the same interval of coordinates, while in the non-Q_η-coordinate they are bound to the interval of coordinates that is spanned by Q_η and in addition each of the paths is monotonous (cf. 3.3(iii)/4.13(ii)). With a coinciding Phase 1 and by having ruled out all extravagances by monotony conditions in Phase 2, it will by possible to verify that the possibilities for our synchronous parametrizations to respond differently to v and to v' are that limited that there is some ε-estimate for v and v' (5.8); Phase 3 cannot do any harm either, since we know our paths to be in Q_η, either.

Now, in order to verify these monotony conditions, we will have to analyze what could happen in a reduction process of word sequences in the given situation. The proof will be combinatorial, and, in addition, we will also work with a combinatorial definition of our three-phases splitting-up (as already defined in 3.3), and just verify at a later state (4.13/3.4) that this combinatorial definition gives the above described geometric picture. The heart of the monotony-argument (4.7) can be understood as follows: If the reduction process should have moved the falling apart-point outside Q_η, this must have been caused either by far-distance cancellations between letters that reported Phase-three

corridor-crossings with letters that reported Phase-two crossings, or letters of Phase 3 have been moved by iterated transpositions into Phase 2 of the in the word sequences. Such processes can only be applied to a word sequence, if a lot of letters are commuting. A non-empty Phase 2 can be understood as a part of our word sequence where such commutativity for applying these processes has been given. If now we would see non-monotony in Phase 2, we would also see some line of corridors that is crossed twice in opposite directions. If there should be some hole between these acts of crossing, this hole would cause too much non-commutativity for Phase 2. On the other hand, if there should be no hole in the entire region in question, then we would see a cancelling pair of letters and enough commutativity to perform this cancellation. In this case we would have a contradiction since our combing paths v and v' belong to reduced word sequences (cf. 1.3). In 4.7–4.12 we will discuss enough separate cases to make this vague geometric idea precise.

§4. Preparing the final Estimates for the proof of the Lifting Theorem

4.1 Lemma: *Let ω and ω' be as in 3.3. Then ω can be transformed into ω' by applying the following processes only:*

(i) $\omega|_{[0^{\mathbb{N}},\kappa_1]}$ *can be turned into* $\omega'|_{[0^{\mathbb{N}},\kappa'_1]}$ *by truncations and appendances that give a strong equivalence relation.*

(ii) $\omega|_{[\kappa_1,\kappa_2]}$ *can be turned into* $\omega'|_{[\kappa'_1,\kappa'_2]}$ *by truncations which are a strong equivalence relation, and by inserting and deleting letters of* $\overline{\mathcal{A}_{Q_\eta}}$ *with the latter transformations possibly affecting the equivalence class.*

(iii) $\omega|_{[\kappa_2,\ell^{\mathbb{N}}]}$ *and* $\omega'|_{[\kappa'_2,\ell^{\mathbb{N}}]}$ *both only consist of letters of* $\overline{\mathcal{A}_{Q_\eta}}$.

The Proof is lengthy and hence split up into 4.2–4.6:

4.2 Remark on how to understand the word *"phase"* in 4.3–4.6:
Having subdivided our path v in 3.3 so as to go through three "Phases" will play an important role in the remainder of our continuity proof 3.2–5.11. The purpose of this introductory paragraph is to recall that there is another meaning of "Phases" around. For, one of the main references of this proof will be D.11.10, and in D.11.10 also some "Phases" have been defined. These two concepts of phases must not get mixed up: the D.11.10-phases refer to a transformation process, while the 3.3-phases impose a subdivision of a path and its associated word sequence. We hereby agree that we will not use the word "Phase" in the D.11.10-sense in the remainder of this proof, unless when explicitly accompanied by the corresponding reference number. Instead "phases" are exclusively to be understood as defined in 3.3 and may refer to the segments of our path and to the associated word sequences.

Having clarified this aspect of the language to be used in the following, we proceed by explaining

4.3 The basic set-up for the proof of all three items of 4.1:

Since by Assumption 3.2 v' corresponds to a homotopically η-near path with respect to v, the path v' is homotopic to a path $v \cdot v''$, where v'' is a path running from $v(b_v)$ to $v'(b_{v'})$ inside $U(x,\eta)$ only. Let ω', ω'' be the reduced representative of the relative homotopy class of v', v'', respectively, where according to D.3.6/D.7.9 we can define the corresponding equivalence relation in such a way that the description of $\overrightarrow{\omega}'$ is free but the descriptions of $\overleftarrow{\omega}'$, $\overleftarrow{\omega}''$ and $\overrightarrow{\omega}''$ are fixed so as to coincide with with $\overleftarrow{\omega}$, $\overrightarrow{\omega}$ and $\overrightarrow{\omega}'$, respectively. Consequently by D.12.1 the product $\omega \cdot \omega''$ is weakly equivalent to ω'. Since v'' was entirely contained in $U(x,\eta)$, the associated word sequence only contains $\mathcal{A}_{Q_\eta}$-letters and this condition cannot have been overturned by having reduced this sequence to ω'', i.e.

(1) all letters of ω'' are contained in $\overline{\mathcal{A}_{Q_\eta}}$,

also. By uniqueness of reduced representatives (cf. D.7.9) we have that ω' results from reducing $\omega \cdot \omega''$ (without loss of generality we can think of ω' being accordingly adjusted in its firm equivalence class, cf. D.7.10/D.7.13 and 2.16). Lemmata D.11.6 and D.11.9 have dealt with this situation, i.e. the situation that the product of two reduced word sequences is to be reduced again. These results allow us to regard all letters of ω as fixed during the transformation process $\omega\omega'' \mapsto \omega'$, while the transpositions according to D.11.10(3.) are all performed by moving the letters of ω'' only. In order to compare position sequences of ω and of ω', we have to stick to the framework as described in D.11.13:
Recall that according to our convention D.11.12 a position κ inside ω can also be regarded as well-defined in each word sequence which is obtained by deleting some letters of ω, but not necessarily within a word sequence which is derived from ω by deleting and inserting letters. However, as pointed out in D.11.13 already, we can argue as follows: According to D.11.10/D.1.19 there is some intermediate state in the course of transforming ω into ω', namely after all operations of D.11.10(1.) and D.11.10(2.) are performed, and before the first operation of D.11.10(3.) can be performed, and let

(2) $\widehat{\omega}$ be the word sequence that corresponds to this intermediate state.

Since the initial substring of $\widehat{\omega}$ can be derived from ω by only deleting letters and from the initial substring of ω' also by deletions only (cf. D.11.13), any position variable (cf. D.11.12) of the initial part of $\omega\omega''$ is also factored through to some position of $\widehat{\omega}$. Recall that by 3.3(iv) we had defined κ_1' and κ_2' to be those position sequences of ω' that satisfy the same conditions as κ_1 and κ_2 do for ω according to 3.3. According to the above and to D.11.13 the position variables $\kappa_1, \kappa_2, \kappa_1', \kappa_2'$ are all well-defined inside $\mathbb{P}(\widehat{\omega})$, and in that sense they can be compared according to C.5.15. This will happen in the subsequent step of the proof of 4.1:

4.4 Proposition: *Assumptions and Notations as in Lemma 4.1. Then the position sequences κ_j and κ_j' $(j \in \{1,2\})$ are equivalent according to Definition C.5.18(iii) when regarded as position sequences of $\widehat{\omega}$, cf. 4.3(2).*

Proof: Assume that they were not equivalent, i.e. that for some fixed j the number of $\widehat{\omega}$-letters between κ_j and κ'_j is unbounded. Hence in particular there is some level-index i within $\widehat{\omega}$ where there are four letters "$l_1l_2l_3l_4$" of the same basic direction between κ_j and κ'_j. Since these letters can also be found within ω or ω'' we can also define a position between l_2 and l_3 in either of those word sequences. Without loss of generality, assume $\kappa_j > \kappa'_j$ inside $\widehat{\omega}$: In this case we choose to use some ω-position between l_2 and l_3 in the following arguments. By using the methods of C.5.5–C.5.6 this position can be extended to some position sequence $\kappa''_j \in \mathbb{P}(\omega)$. According to C.5.9 for each level of ω the position to which κ_j points is to the right of the iterated left split-up letters of l_4, the position to which κ'_j points is to the left of the iterated right split-up letter of l_1, and the position to which κ''_j points is between the iterated left split-up letters of l_2 and the iterated right split-up letters of l_3, and by C.5.7 κ''_j can be, without loss of generality, constructed in such a way so as to lie between κ_j and κ'_j for all level indices smaller than i. The same condition holds for the higher level indices, because the orders of the iterated split-up letters of the same basic direction cannot be reversed. Now κ_j has been defined to be the smallest position sequence of ω such that each letter of the corresponding basic direction(s) (according to j and to the Q_η-direction it depends whether this condition is applied to the letters of both basic directions or to the horizontal or vertical ones, only) which is left of κ_j belongs to $\overline{\mathcal{A}_{Q_\eta}}$. I.e. there cannot be a smaller position sequence than κ_j without having some non-$\overline{\mathcal{A}_{Q_\eta}}$-letter on its right hand side. Consequently, this has to apply to κ''_j also, which we had defined as a position sequence for ω. Now according to 4.3(1)/D.11.10 such a non-$\overline{\mathcal{A}_{Q_\eta}}$ letter cannot be made disappear by the operations that turn $\omega\omega''$ into $\widehat{\omega}$, and hence the corresponding letter can still be found inside $\widehat{\omega}$. Inside $\widehat{\omega}$ it is placed to the right of κ''_j and hence in particular to the right of κ'_j. The latter property persists if we skip from $\widehat{\omega}$ to ω' by inserting the corresponding letters. Now for ω' the position κ'_j had been defined by a similar minimality condition as κ_j had been for ω and hence the existence of such a letter yields a contradiction. Similar arguments hold if instead of $\kappa'_j < \kappa_j$ we assume that $\kappa'_j > \kappa_j$. Consequently we see that κ'_j and κ_j are position sequences of $\widehat{\omega}$ that point at places within a bounded distance apart and hence for $\widehat{\omega}$ these position sequences are equivalent in the sense of C.5.18(iii). □

We now proceed by stating and proving a five-items proposition which then will immediately imply our three-item lemma 4.1:

4.5 Proposition: *Assumptions and notations as in 4.1–4.3. Then the following holds:*

(i) It is just a matter of strong equivalence based on appendances and truncations only, to go from $\widehat{\omega}|_{[0^{\mathbb{N}},\kappa_j]}$ *to* $\widehat{\omega}|_{[0^{\mathbb{N}},\kappa'_j]}$.

(ii) Let κ be the bounding position of that block of $\omega_{|[\kappa,\ell^{\mathbb{N}}]}$ *that according to D.11.10(1.) cancels completely against some initial block of ω''. Then the bounding position sequence κ is bigger than or equal to κ_2, i.e. only Phase 3 of ω can cancel against ω''.*

(iii) If a letter of Phase 2 of ω should be affected by one of the far-distance cancellations as described in D.11.10(2.), it must have been a Q_η-directed letter or it must have been a letter that is truncatable to the end of $\omega_{[0^{\mathbb{N}},\kappa_2]}$.

(iv) If a letter of Phase 1 of ω should be affected by one of the far-distance cancellations as described in D.11.10(2.), it must have been a letter that is truncatable to the end of $\omega_{[0^{\mathbb{N}},\kappa_1]}$.

(v) If by processes D.11.10(3.) letters of ω'' should be moved inside ω, they can only be placed inside Phase 3, or, if they should be letters of the Q_η-direction, they may be also placed inside Phase 2 or at places of Phase 1 that are truncatable to the end of that word sequence.

Proof:

Ad (i) : As a consequence of 4.4, $\min\{\kappa_j,\kappa'_j\}$ and $\max\{\kappa_j,\kappa'_j\}$ are also position sequences that remain just a bounded distance apart, and consequently $\widehat{\omega}\big|_{[\min\{\kappa_j,\kappa'_j\},\max\{\kappa_j,\kappa'_j\}]}$ is a bounded position sequence. Hence none of its letters can contain its full set of split-up letters within that word sequence, and thus one can come from $\widehat{\omega}\big|_{[0^{\mathbb{N}},\max\{\kappa_j,\kappa'_j\}]}$ to $\widehat{\omega}\big|_{[0^{\mathbb{N}},\min\{\kappa_j,\kappa'_j\}]}$ by such a sequence of truncations that gives a sequence of strong equivalence transformations (cf. D.3.2/D.3.3). By going via $\widehat{\omega}\big|_{[0^{\mathbb{N}},\min\{\kappa_j,\kappa'_j\}]}$, Claim (i) follows.

Ad (ii) : A block of ω that could cancel against a block of ω'' has to consist of letters of $\mathcal{A}_{Q_\eta}$ only. Hence only Phase 3 of ω could have been affected by such a process so that $\kappa \geq \kappa_2$.

Ad (iii): Now assume that some letter L of Phase 2 of ω that belongs to the non-Q_η direction and is not truncatable to the end of $\omega_{|[\kappa_1,\kappa_2]}$ will be cancelled in the course of the process D.11.10(2.). By D.3.2/D.3.3, since l is assumed to be not truncatable to the end of ω_2, and since κ_2 is characterized by the corresponding minimality property (cf. 3.3), all the iterated right split-up letters have to appear and there have to be some of them so that between them and κ_2 some other letter that is not contained in $\overline{\mathcal{A}_{Q_\eta}}$ can be found. By 3.3(iii) this letter l' must be non-Q_η-directed, also. Use $l \in \overline{\mathcal{A}_i}$ to denote the corresponding split-up letter of L. Because $BD(l) = BD(l')$, (cf. C.5.9) there exists a position sequence κ'_2 separating all iterated left split-up letters of l from all iterated right split-up letters of l'. Without loss of generality, we can assume that $l' \notin \overline{\mathcal{A}_{Q_\eta}}$, since by construction $\kappa'_2 < \kappa_2$, and by the minimality property of κ_2 some letter with the according properties of l' has to exist. Since they have the same basic direction, l is not commuting with l' and, because of $l' \notin \overline{\mathcal{A}_{Q_\eta}}$, none of the processes as described in D.11.10 can make the letter l' disappear in the course of transforming $\omega\omega''$ into ω'. Hence there cannot exist a state within the corresponding transformation process of our word sequence where a far-distance cancellation between l and some ω''-letter could be performed. Hence (cf. D.3.4) even the letter L as originally considered cannot be deleted in the course of a compatible far-distance cancellation.

Ad (iv): This follows from a proof by contradiction similarly as for (iii). This time we may assume that the corresponding place of the iterated split-up letters of L is well-ahead of κ_1, the corresponding minimality property of κ_1 may be used in the proof, and hence the existence of such a non-truncatable affected letter can in both basic directions be ruled out.

Ad (v): Recall that in the course of our proof of (i) we had by quoting D.11.13 in the set-up 4.3 been arguing that the inverse of the process of bringing in ω''-letters

into ω according to D.11.10(3.) might be realized as a cancellation process with some appropriately defined extra factor, that only consists of $\overline{\mathcal{A}_{Q_\eta}}$-letters. This observation makes it possible to conclude that (v) follows by analogous arguments as given to prove (iii) and (iv). □

4.6 Proof of Lemma 4.1:
In order to deduce the claims 4.1(i)–4.1(iii) from 4.5(i)–4.5(v), subdivide the transformation process $\omega_{|[\kappa_1,\kappa_2]} \mapsto \omega'_{|[\kappa'_1,\kappa'_2]}$ into three steps according to

$$\omega_{|[\kappa_1,\kappa_2]} \mapsto \widehat{\omega}_{|[\kappa_1,\kappa_2]} \mapsto \widehat{\omega}_{|[\kappa'_1,\kappa'_2]} \mapsto \omega'_{|[\kappa'_1,\kappa'_2]}.$$

According to the splitting up of our transformation process D.11.10 into three phases as described there, the arguments as given above in 4.5(ii)–4.5(v) apply precisely either to the the first or to the last of these steps, and the middle step consists by 4.5(i) of truncations and appendances only. However, truncatable letters had been exempted in the claims of this lemma, and that way it is proved by now. □

4.7 Lemma: *All non-truncatable letters of $\omega_{|[\kappa_1,\kappa_2]}$ and of $\omega'_{|[\kappa'_1,\kappa'_2]}$ that belong to the same basic direction have the same exponent $+1$ or -1. This exponent coincides for the two word sequences for the letters of the non-Q_η direction, but it may not for the other one.*

Proof: The key-argument in both cases will be that $\omega_{|[\kappa_1,\kappa_2]}$ and $\omega'_{|[\kappa'_1,\kappa'_2]}$ are parts of reduced word sequences. The coincidence of exponents for the non-Q_η direction is an immediate corollary from the coincidence of letters (cf. 4.1(ii)). Hence we can in the remainder of this proof restrict ourselves to considering the undashed word sequence only:
By the minimality property of κ_2 and the existence of a non-$\overline{\mathcal{A}_{Q_\eta}}$-letter in any non-empty Phase 2 we have some parameter value $t \in [t_1, t_2]$ such that $v(t)$ lies outside Q_η. We will give a proof by contradiction, and hence we assume that we find within $\omega_{|[\kappa_1,\kappa_2]}$ two letters of the same basic direction with inverse indices. Without loss of generality, we can assume that the segment of v that is bounded by the corresponding corridor exit and entrance points of those letters lies outside Q_η, since $v(t)$ was outside Q_η, and if $\overline{v_{|[t_1,t]}}$ would return to Q_η, this would definitely imply inverse acts of crossing corridors outside Q_η which then could alternatively be used for the subsequent argument. Having assumed that there exist letters with different exponents, it is not hard to also find letters with different exponents from the same line of corridors. This can be most easily seen by geometry: Think of the corridor network and of acts of crossing in reversed order on different (and hence parallel) lines of corridors of the same basic direction. It is clear that a path cannot go from the exit point of one to the entrance point of the other act of crossing without having to cross one of the two lines of corridors of consideration for a second time. Having proven the existence of such pairs, we can assume without loss of generality that we skip to considering an innermost one. We now have to make a distinction whether the two letters in question are from the same corridor (and form a "cancelling pair" $l \ldots l^{-1}$) or whether they are not. This makes two cases, and by also

taking into account whether such pairs of letters are found for Q_η- or for non Q_η-directed letters we end up with four different cases. In 4.9–4.12 we will complete the proof of this lemma by essentially showing that none of these cases can occur and hence we always have coinciding exponents in Phase 2 of our word sequence. However, the distinction of cases is a bit more complicated than just indicated, since in the non-Q_η-directed case the distinction whether the two letters of our pair are drawn from the same corridor or only from the same line of corridors will not be made according to the level of l only, but we will also have to take the split-up letters of $l^{(-1)}$ into account. Hence the discussion of these cases in 4.9 and 4.10 is preceded in 4.8 introducing the questions that come with the precise distinction of cases and the corresponding notation. Proceeding that way will allow us to rule out any possibility for non-coinciding exponents and hence prove 4.7 in 4.9–4.12:

4.8 Notation that is needed for the proof of 4.7 in 4.9–4.10:
According with the assumptions of 4.9 and 4.10 we may assume that ω contains on some level a pair of non-Q_η-directed letters l and l^{-1} that denote adjacent acts of crossing of the same line of corridors. Assume that l is placed to the left of l^{-1}. Denote by l', l'' the iterated right/left split-up letters of l and l^{-1}, respectively, as they are found on some finer level i. Even though by construction l' and l'' belong to the same line of corridors, their notation takes care of the possibility that they need not belong to the same corridor any more. Having introduced for split-up letters of l and l^{-1} on a possibly finer level that notation, we can justify that the names l and l^{-1} have been chosen as if these letters belonged to the same corridor in any case, even though that could not be assumed a priori: The level i and with it l' and l'' are to be chosen in such a way that this level contains an obstruction for performing the cancellation between l and l^{-1}. Since $\omega_{|[\kappa_1,\kappa_2]}$ had been part of a reduced word sequence, there must be such an obstruction. According to D.3.4 it must be visible on some appropriately chosen level i either by the property that the subword that is bounded by l' and l'' contains letters which are not commuting with l, or by the property that l and l' themselves do not form a cancelling pair, i.e. belong to different corridors. One of this two cases must be given, perhaps for l and l^{-1} itself already, but at least for some finer level index i.
Denote by h_1 and h_2 the two half-planes into which $\mathbb{L}(l)$ separates our plane. Since l was assumed to be a non-Q_η-directed letter, $\mathbb{L}(l)$ cannot intersect Q_η. Hence Q_η lies entirely in one of our half-planes, say h_2, and we now want to argue that, without loss of generality, the segment of v, after its exit from having crossed l' to its entrance for crossing l'' lies in the other half plane h_1. For, if not, then v is at the exit point of its act of crossing labelled l^{-1} in the half-plane h_1, and hence it has to intersect the line of corridors containing l before it can reach $v(t_2)$ which lies on the boundary of Q_η by 3.3(v). Thus we could have easily found another cancelling pair which satisfies the corresponding property. So, without loss of generality, v is in h_1 at the beginning and at the end of the segment bounded by the acts l' and l''. Since we cannot be sure that it remains in h_1 for the entire segment, we have to introduce further notation: We let $l_1 := l'$ and then ask for the next letter to the right in ω_i that belongs to the same line of corridors. Since geometry implies that it represents an act of crossing in the reverse

direction, it must have the inverse exponent as l_1, and therefore we will denote it by l_2^{-1}. If l_2^{-1} is already the right split-up letter of l^{-1}, (i.e. if $l_2^{-1} := l''$), then no more $l_?$ are defined, otherwise we continue to attach names $(l_3, l_4^{-1}, l_5, l_6^{-1}, \ldots)$ to those letters drawn from the same line of corridors as l_1 till we reach l''. Then k is defined so that $l'' := l_{2\cdot k}^{-1}$. Observe that this letter must have an even index, according to direction of crossing to which it belongs, and according to the fact that these directions are alternating in our list of letters: We then ask the following questions for each $m \in \{1, 2, \ldots, k\}$:

(i) Is $l_{2\cdot m}$ the inverse letter with respect to $l_{2\cdot m-1}$?

(ii) Is each Q_η-directed letter that occurs between $l_{2\cdot m-1}$ and $l_{2\cdot m}^{-1}$ commuting with $l_{2\cdot m}$?

(iii) We then look whether non-Q_η directed letters can be found between $l_{2\cdot m-1}$ and $l_{2\cdot m}^{-1}$ that belong to the adjacent line of corridors as $l_{2\cdot m}$. Since $l_{2\cdot m-1}$ and $l_{2\cdot m}^{-1}$ are an innermost pair from $\mathbb{L}(l_1)$, the adjacent line can only be found in h_1. If we find such letters, we denote them by $l_{m,1}, l_{m,2}^{-1}, l_{m,3}, l_{m,4}^{-1}, \ldots, l_{m,2\cdot m_1}^{-1}$. After having set up the corresponding lists we ask the analogous two questions for each for the subwords bounded by $l_{m,2\cdot m'-1}$ and $l_{m,2\cdot m'}^{-1}$, namely whether $l_{m,2\cdot m'-1}$ and $l_{m,2\cdot m'}^{-1}$ are inverse to each other and whether all non-Q_η-directed letters between them are commuting.

(iv) For each index-pair m, m' we also list the letters that are taken from the line of corridors adjacent to the one containing the $l_{?,?}$-letters and are placed between $l_{m,2\cdot m'-1}$ and $l_{m,2\cdot m'}^{-1}$, and to the latter lists of letters $l_{?,?}$ we ask the same two questions (i) and (ii) again.

(v) If necessary, we iterate this principle of putting up lists of letters with larger index tuples a few more times. However note, that each iteration steps takes us deeper into h_1. On the other hand, we cannot get arbitrarily deep into it, since that would imply, that even on the index level of the original root letter l the neighbouring line of corridors would have to be crossed by v, and this would contradict to l any l^{-1} being an innermost cancelling pair on their index level. Hence after a finite number of iterative steps and considering list of $l_{?,\ldots,?}$ letters with longer and longer index-tuples, we find all the next lists to be empty.

Hence the number of questions to be asked according to (i)–(v) remains finite.

Note that l could perfectly be cancellable against l^{-1}, even if some of the above questions are to be answered by "No!". E.g. in the case $k = 2$ the corresponding string of $l_?$-letters may read "$L \ldots L'^{-1} \ldots L' \ldots L^{-1}$", and here, even if the first and second and the third and fourth letter do no cancel, L might be cancellable against L^{-1} after L' had been cancelled against L'^{-1}. Similarly, it might happen that non-commuting Q_η directed letters are found between our non-Q_η-directed ones, but if the Q_η-directed letters occur in cancelling pairs, it might be possible to firstly cancel these letters and then our Q_η-directed letters, also. However, the following proposition will only require the converse conclusion.

4.9 Proposition: *Assumptions and notations as in 4.7/4.8. Then $\omega_{\lfloor[\kappa_1,\kappa_2]}$ cannot contain a pair of non-Q_η-directed letters from different corridors with different exponents that are the iterated right and the iterated left split-up letters from some letters that denote adjacent crossings of the same line of corridors on a prior level.*

Proof: The proof of this proposition will rest on discussing two cases separately, namely whether all of the above questions 4.8(i)–4.8(ii) will be answered by "Yes!" or whether at least one will be answered by "No!".

CASE ONE ("Yes!-only-Case"). In this case we want to deduce a contradiction by showing the we have traced enough information about our word to see that l can be cancelled against l^{-1} which is a contradiction to the occurrence of these letters in a reduced word (cf. 4.8). Hence this will bring the desired negative answer in this case.

By what follows from having only "Yes!"-answers to the above questions 4.8(i)–4.8(ii), we conclude that at first those $l_{?,\ldots,?}$-letters with the biggest number of indices can be pairwise cancelled against each other, since only commuting letters are placed between each pair. The commuting letters between them remain, but thanks to the questions that had earlier been asked, we know that these letter also commute with all the remaining $l_{?,\ldots,?}$-letters, and hence they will not obstruct any of the desired cancellations that are still to be performed. Since by going step by step into our half-plane h_1, we have traced all non-Q_η directed letters, we now know that there do not exist obstructions for cancelling these $l_{?,\ldots,?}$-letters with one index less pairwise against each other. By iterating this principle, we finally come that far, that also the letters $l' = l_1, l_2^{-1}, l_3, l_4^{-1}, \ldots, l_{2\cdot k}^{-1}$ disappear from our word ω_i. The corresponding substring of ω_i need not have become empty because of the cancellations process, since we did never ask which letters are placed between some $l_{2\cdot m}^{-1}$ and l_{2m+1} for some m. According to the internal compatibility of our words we know that it could just be letters from corridors that intersect the corridor l, but even such phenomena as non-commuting letters might be found amongst them. However, since all these letters only report movements of our path inside l, we conclude that the corresponding level index i does not contain an obstruction for cancelling l against l^{-1}. On the other hand we had assumed that it does (this assumption had made us consider the corresponding split up letters l' and l'' on that index level!). From this contradiction we get that at least one of the questions as asked in 4.8(i)–4.8(v) must have had a negative answer, i.e. that we are only left with finding a contradiction in

CASE TWO: ("$\exists$-No!-case"). In this discussion we will restrict our considerations to the case that a question of Type 4.8(i) has a negative answer, since this assumption appears compatible with the basic set-up of this Proposition 4.9 (the Type-4.8(ii) case will then be discussed in the course of the proof of Proposition 4.10). And, for the sake of avoiding a multi-indexed language, we will proceed to argue as if the corresponding pair of adjacent letter for which the 4.8(i)-question had a negative answer had been the split-up letters l' and l'' directly, since it does not take more than straightforward substitutions to adapt the arguments given below to any other corresponding pair of letters. Since a negative answer to question 4.8(i) implies that l' and l'' stand for different corridors, there must be at least one hole between l' and l'', and, in order to come from the exit point of l' to the entrance point of l'' our path v has to cross a line of corridors that is emanating from that hole and is perpendicular to $\mathbb{L}(l)$, i.e. a Q_η-directed corridor. Since according to our previous assumptions l' and l'' are adjacent letters on their line of corridors, the relevant segment of v cannot intersect this line, and consequently v has

to be in the half-plane h_1 when crossing the Q_η-directed corridor. Hence, this corridor cannot intersect with Q_η because it ends at the hole between l and l', and thus it cannot enter h_2. This contradicts to our basic assumption of Phase 2 that all letters of the Q_η-direction stand for corridors intersecting Q_η (cf. 3.3).
Having obtained a contradiction in both cases proves the negative claim of this proposition. □

4.10 Proposition: *Assumptions and notations as in 4.7/4.8. Then $\omega_{\|[\kappa_1,\kappa_2]}$ cannot contain a pair (l, l^{-1}) of non-Q_η-directed letters with different exponents such that all iterated right split-up letters of the first and all iterated left split-up letters of the second entry of the pair belong for each level to the same corridor.*

Proof: Since it is our general assumption that l does not cancel against l^{-1}, according to D.3.4 there has to exist some index i where the subword bounded by the according split-up letters l' and $l'' = l'^{-1}$ contains some obstruction for cancelling l' with respect to l'^{-1}, i.e. some non-commuting letter L which has to exist if some of the questions of Type 4.8(ii) had a negative answer (cf. the analogous situation in 4.8(i)). Actually, this case can be treated by mixing up the arguments of 4.8/4.9 just in a slightly different way: As in 4.8/4.9 we consider the split-up letters l' and $l'' = l'^{-1}$. Without loss of generality we can assume that v between its two acts of crossing that are associated to l' and to l'^{-1} does not cross the corresponding line of corridors again (otherwise we have to play the sequence of questions 4.8(i)–4.8(v) and find a better pair of letters for our arguments, as in 4.9). Hence we know that the act of crossing L takes place in h_1. Since l' and L belong to different basic directions, the lines of corridors $\mathbb{L}(L)$ and $\mathbb{L}(l')$ have to intersect. Since, on the other hand, the corresponding variables have been assumed to be non-commuting, either $\mathbb{L}(l') \cap \mathbb{L}(L)$ must be a hole or at least one of the two corridors l' and L must end at some hole before reaching $\mathbb{L}(l') \cap \mathbb{L}(L)$. If this happens to L we are done, since this is the same situation that already gave a contradiction in Proposition 4.9.
In the other case the following argument holds: If L reaches the line $\mathbb{L}(l)$, since L and l do not commute it can only reach this line outside l. Consequently, the relevant segment of v has passed inside h_1 over the span of non-Q_η directed variables that correspond to one of the holes at which l ends. Hence it has crossed the Q_η-directed line of corridors that emanates from that hole, and by having found this act of crossing we get the same contradiction as two sentences ago. This completes the proof of this case; and with 4.8–4.9 the proof of 4.7 for a non-Q_η-directed letter l is also complete. □

In the following two propositions we will be discussing the case where letters of different exponents can be found amongst the Q_η-letters. Since we managed to complete the discussion of our first cases successfully, without loss of generality we can restrict our attention now to the case that all non-Q_η letters have the same exponent.

4.11 Proposition: *Assumptions and notations as in Lemma 4.7. Then $\omega_{\|[\kappa_1,\kappa_2]}$ cannot contain adjacent Q_η-directed letters l and l'^{-1} that are drawn from the same line of corridors, but not from the same corridor.*

Proof: Since l and l'^{-1} are by assumption adjacent letters from the same line of corridors, they must by the internal compatibility of words C.1.6 represent acts of crossing in the opposite direction. As a consequence of the general assumption for both Propositions 4.11 and 4.12 we get that our path has for the relevant segment between the acts of crossing l and l^{-1} and the end of Phase 2 a monotonous behaviour with respect to the Q_η-coordinate. Hence we get the following geometrical picture of the behaviour of the path v in Phase 2:
Let $Q_\eta =: [a,b] \times [c,d]$ and hence $\partial Q_\eta = \{a,b\} \times [c,d] \cup [a,b] \times \{c,d\}$. We cannot tell whether the x- or the y-coordinate is the Q_η- or the non-Q_η-coordinate, but we know that $v(t_2)$, which by 3.3(v) lies on ∂Q_η and must have its Q_η-coordinate on one of the corresponding interval-endpoints a, b, c or d. If we now trace $\overline{v}$, the inverse of v, when starting at the point $v(t_2)$, we see from the monotonous behaviour of the Q_η-coordinate, that $\overline{v}$ is constantly moving away from Q_η. During this act of moving away, the non-Q_η coordinate of $\overline{v}$ is at those points that correspond to the act of crossing l'^{-1} within a certain interval of non-Q_η coordinates, and it returns to the same interval of non-Q_η-coordinates when it comes to the act of crossing l. By the general assumption of Phase 2 we know that the corridors that correspond to the letters l and l' are both assumed to intersect Q_η. But according to the present geometrical picture it is impossible for l, since for that purpose those part of the according line of corridors that is filled by some other corridor l' would have to belong to l as well. And hence we get that this case yields the contradiction that proves the negative claim of this proposition. □

4.12 Proposition: *Assumptions and notations as in Lemma 4.7. Then* $\omega_{\restriction[\kappa_1,\kappa_2]}$ *cannot contain adjacent Q_η-directed letters that are drawn from the same corridor.*

Proof: We use the same notation as in 4.11, but now we have that $l = l'$. However, we can still assume that l and l' are an innermost pair of cancelling letters which cannot be cancelled due to some obstruction which is visible on the same level-index to which l and l' belong. Since by assumptions of Propositions 4.11 and 4.12 all non-Q_η letters between l and l^{-1} have the same exponent, we get that all obstructions for cancelling l against l^{-1} could only be given by some of those letters being not commuting with l, or if some of the Q_η-directed letters between l and l' do not form cancelling pairs. From our assumption that l and l^{-1} form an innermost cancelling pair that sees an obstruction, we get that the first possibility has to be given; use L to denote one of the corresponding non-commuting letters. In addition, we can assume that our path between its acts of crossing l and by l^{-1} does not return to the line of corridors spanned by l (at least, not for a complete act of crossing), since such a completed act of crossing would again be acknowledged by some letter l, and we would see several cancelling pairs "ll^{-1}" or "$l^{-1}l$", one of which would have to contain our letter L. Hence the corresponding pair would not be the innermost for the level index i which sees an obstruction for cancelling. Hence we know that our path v must behave as follows: After having performed the act of crossing that is labelled by "l" the path moves away from $\mathbb{L}(l)$, and, while the Q_η-coordinate is monotonously changing towards Q_η, our path performs the L-crossing. Then, while being strictly monotone with respect to the Q_η-coordinate, our path returns to the line of

corridors spanned by l. Let us now return to the region where the L-crossing took place. We try to follow this corridor directed towards l, but we know that we are unable to reach the corresponding line, since for the corresponding domain of Q_η-coordinates the $\mathbb{L}(l)$-line of corridors is filled by l, and l does not commute with L. Hence the L-corridor has to end at some hole E before it reaches the line l. Our geometrical assumptions give that the Q_η-directed line of corridors emanating from E must also have been crossed twice by v, and the corresponding letters cannot give a cancelling pair, since thanks to the hole E each of the acts of crossing which have to be performed between the l- and the L-crossing and between the L- and the l^{-1}-crossing the non-commuting letter do take place at different corridors of this line. I.e. we have found a non-cancelling pair of letters. Similar as in the above Proposition 4.11 we conclude that the first of these letters stands for an Q_η-directed corridor which cannot reach Q_η, and hence we get that same contradiction as in Proposition 4.11. □

Since by 4.9–4.12 all possibilities for non-coinciding exponents could be ruled out, by now the proof of 4.7 is complete. □

4.13 Corollaries: *With notations as in 3.2/3.3 we have:*
(i) $v|_{[0,t_1]}$ *and* $v'|_{[0,t'_1]}$ *describe the same abstract path.*
(ii) $v|_{[t_1,t_2]}$ *and* $v'|_{[t'_1,t'_2]}$ *are monotonous in each of its coordinate functions. The directions of monotony with respect to the* Q_η*-coordinate coincide.*
(iii) $v([t_2,1]) \subset Q_\eta \supset v'([t'_2,1])$.

Proof — Ad (i): This is an immediate consequence from 4.1(i), i.e. from the fact that the associated word sequences are strongly equivalent, and from D.12.3.
Ad (ii): This is a consequence of 4.7, i.e. from the fact that, if all possible truncations are performed, the corresponding exponents of the letters of the associated word sequences coincide. Since our corridor network is arbitrarily fine, it would be detected on some level of the word sequence if there were two parameter values t, t' such that $v(t) - v(t')$ had the wrong sign.
Ad (iii): This is an immediate consequence of 4.1(iii). With similar arguments as in (ii) it would on a sufficiently high level cause crossings of corridors that do not intersect with Q_η, if our claim (iii) should be not fulfilled. □

4.14 Lemma-Definition of exceptional letters:
With notations as in 3.2 analyze for each letter of ω *and of* ω' *whether the corresponding corridor exit point (cf. D.10.5(ii)) is bigger than* κ_j *or* κ'_j*, respectively, for* $j \in \{1,2\}$*. Call a non-*Q_η *directed and a* Q_η*-directed letter "exceptional", if this condition is satisfied for* $j = 2$ *or* $j = 1$*, respectively. Then the following assertions hold:*
(i) For i_0 *as defined in 3.2 and for all* $i < i_0$ *each of the words* ω_i *and* ω'_i *can have at most one exceptional letter in each of the two basic directions, and if such a letter occurs, then it is the last one.*
(ii) Use the upper indices "$()^{\mathrm{h}}$*" or "*$()^{\mathrm{v}}$*", if attached to a word* w*, to denote those words resulting from* w *by deleting all vertical or all horizontal letters, respectively, and use*

$\hbar$ exclusively as a variable for an element of $\{h,v\}$. Then for each $\hbar$ we have that if $\omega_i^{\hbar}$ and $\omega_i'^{\hbar}$ do or do not both contain exceptional letters, they have to coincide. In the other case, deleting the exceptional letters describes the transformation between the two words.

The proof of the Claims (i) and (ii) is lengthy and hence split up into the Propositions 4.15–4.19. It will rest on firstly proving the claim for some other word sequences that are strongly equivalent to ω and ω', and on later discussing that none of the possible operations of strong equivalence can overturn the properties (i) and (ii). The latter step will happen in separate propositions according to the types of possible equivalence operations:

4.15 Proposition: *With assumptions and notations as in 4.14 there exist word sequences η and η' that instead of ω and ω' satisfy the Claims 4.14(i) and 4.14(ii) and that are strongly equivalent to ω and ω', respectively.*

Proof: Recall (cf. 3.2) that Q_η is defined in such a way that three parallel horizontal and vertical lines of corridors of $\mathcal{N}_{i'}$ pass through Q_η, such that some segment of the outer boundaries of outer two of these parallel lines coincide with opposite boundaries of Q_η. Since by 3.2 i_0 has to be smaller than i' and cannot be the predecessor of i', it follows that at most one line of corridors and at most one of corresponding boundary lines can in each of the basic directions intersect Q_η. By using this information on the one hand and 4.13 on the other, it is not hard to see that the paths v and v' that are associated with ω and ω', respectively, have in principle to cross the same corridors in the same order. In this context recall (cf. 4.13(i)–(iii)) that $v_{\restriction[0,t_1]}$ and $v_{\restriction[0,t_1']}$ coincide up to reparametrization, that $v_{\restriction[t_1,t_2]}$ and $v_{\restriction[t_1',t_2']}$ go monotonously in the same direction over the same interval of Q_η-coordinates, while their non-Q_η-coordinates are restricted to the same interval that also spans Q_η and is of the length $2\cdot\eta$. Finally, $v_{\restriction[t_2,1]}$ and $v'_{\restriction[t_2',1]}$ both can just move around inside Q_η. Since the width of any corridor of $\mathcal{N}_i$ is bigger than the width of Q_η, in follows that even in Phases 2 and 3, where according to the above geometrical restrictions v and v' need not coincide, one of our path cannot perform an act of crossing that is not performed by the other. The differences as described above merely could permit one of the to paths to complete a certain act of crossing that has already started in Phase 1 for the non-Q_η-coordinate or in Phase 2 for the Q_η-coordinate, while it remains uncompleted for the other path. This is precisely what can cause the deletion or the appendance of an exceptional letter — as claimed by the assertion of this proposition. Please note that the above arguments do imply the proposition, but not Lemma 4.14, since they just imply that there exist possible readings from v and v', i.e. some word sequences η and η' that are strongly equivalent to ω and ω', respectively and that satisfy 4.14(i) & 4.14(ii). Since by D.2.5 different readings of the same path are strongly equivalent, the proposition follows. □

According to our proof-programme as described in 4.14, we proceed in 4.16–4.19 by discussing that the properties 4.14(i) and 4.14(ii) cannot get lost if $\omega^{(\prime)}$ is turned into $\eta^{(\prime)}$:

4.16 Proposition: *If $\eta^{(\prime)}$ as constructed in 4.15 should differ by truncations from $\omega^{(\prime)}$, this cannot cause that $\omega^{(\prime)}$ fails to satisfy 4.14(i) and 4.14(ii).*

Proof: According to 1.4, ω and ω' do not have equivalent representatives that differ by truncations to the beginning.
On the other hand, truncations to the end of ω and of ω' can only affect $\overline{\mathcal{A}_{Q_\eta}}$-letters, i.e. exceptional letters, and hence they can not result in ω and ω' being in a way different that does not correspond to what is described in Lemma 4.14. □

4.17 Proposition: *If $\eta^{(\prime)}$ as constructed in 4.15 should differ by transpositions from $\omega^{(\prime)}$, this cannot cause that $\omega^{(\prime)}$ fails to satisfy 4.14(i) and 4.14(ii).*

Proof: Transpositions between ${\omega_i}^{(\prime)}$ and its equivalent representative $\eta^{(\prime)}$ can occur, if incompatible readings have been chosen. However, since we had considered ${\omega_i}^{(\prime)^{\mathrm{h}}}$ and ${\omega_i}^{(\prime)^{\mathrm{v}}}$ separately when stating this theorem, such transpositions are irrelevant in this context. In addition note, that apart from that it is not very likely that we will see ω and ω' differing by transpositions, and the geometrical reason for this can be understood as follows: If one analyzes the possibilities for a different behaviour of v and v' according to 4.13 and further takes into account that the corresponding acts of crossing that correspond to the letters to be transposed must be completed (for not having truncated one of the letters), if we should actually see two different letters in different orders in a reduced word, then it is on the one hand possible to think that the paths v and v' behave in such a way that one of the paths avoids going through a crossing region of corridors (and hence the order of letters is clear), and the other path passes through this region. On the other hand, if such a situation should actually be given, it must involve a long Phase 2 (long enough to complete acts of crossing corridors of a level whose index is smaller than or equal to i_0); and for making the paths v and v' fall apart long before the region Q_η, one of the two paths must have been pushed into one of the pushing directions of D.3.10(Fig.k) so that it avoids going through the corresponding crossing region of corridors. Now this implies that for the other path which enters the crossing region of corridors the short order of letters is precisely the same order that occurs for that path where the corresponding acts of crossing did not overlap any more (that path might complete one of the act of crossing after having entered Q_η where a hole inside Q_η forces it into this less short position). However, since according to C.2.4(3) it is always possible to extend any chosen order of letters to the finer levels, if these levels correspond to corridors whose crossing region had been entered by our path, we will see for the path that has entered our crossing region of corridors the same order of letters as for the path which by the shortening mechanism of D.3.10 has been pushed outside.
However, due to the phenomena that gave the fact that we actually did not get unique reduced representatives but just firm equivalence classes of reduced representatives, transpositions might occur. E.g. the picture above might contain, if we take the additional behaviour of Phase 1 into account, a part of a picture as developed in D.6.8, and in that case we might actually see transpositions. However, as explained before, they are irrelevant. □

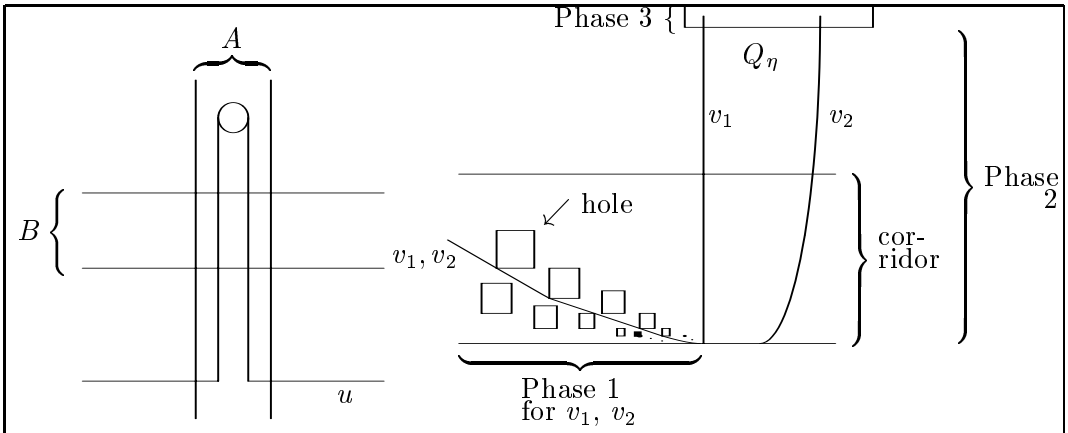


Fig.p (belonging to 4.19):

Fig.p$_1$ shows a path which cannot be represented by a unique reduced word sequence. This situation has also been discussed in D.6.8 already.

Fig.p$_2$ showns the falling-apart of two paths as according to our construction in 4.19 it happens at the end of Phase 1, and a particular nasty situation for where a corridor-boundary could lie with respect to the falling-apart point. In particular observe that a boundary-behavior like in Fig.p$_2$ could in some situations be combined with the non-uniqueness of the sequence of letters as we see it in Fig.p$_1$.

4.18 Proposition: *With assumptions and notations as in 4.14/4.15, if $\omega^{(\prime)}$ should differ from $\eta^{(\prime)}$ by cancellations that only affect non-exceptional letters, this cannot cause that $\omega^{(\prime)}$ fails to satisfy 4.14(i) and 4.14(ii).*

Proof: By B.1.4/D.2.17 a cancellation that is a strongly equivalent process can only occur if the path performs a certain tangency to a corridor boundary-line without intersecting this boundary. According to the assumption of this proposition, the corresponding tangency would have to be at a place or segment that is completely inside Phase 1, or, for a non-Q_η-directed letter, it could also be in Phase 2. However, we postpone the discussion of the latter possibility to Proposition 4.19, and hence we can assume in this case that the corresponding segment of path that contains the tangency has to be the same for ω and for ω'. Consequently the obstructions that could prevent one from extending the corresponding cancellation process throughout the entire word sequence (i.e. whether the corresponding segment that our path spends on the boundary line of a corridor is the outer boundary of a hole of a higher indexed corridor lattice or whether it is not cf. C.2.5/D.10.9) are the same for ω and for ω', also. Since ω and ω' are both assumed to be reduced, the corresponding cancellations have either to be performed in both cases, or in neither of them. Hence ω and ω' cannot differ due to such a strong equivalence process, and this gives the proposition. □

4.19 Proposition: *With assumptions and notations as in 4.14/4.15, if $\omega^{(\prime)}$ should differ from $\eta^{(\prime)}$ by a cancellation of two letters the latter of which is exceptional (at least for one of our paths), this cannot cause that $\omega^{(\prime)}$ fails to satisfy 4.14(i) and 4.14(ii).*

Proof: In difference to 4.18 we now assume that the exit point of the second letter of the cancelling pair is not contained in Phase 1 anymore. We now ask, where this exit point could lie. Since after the corresponding tangency at least one of our two paths has to perform a full act of crossing and since by assumption the width of Q_η is much smaller than the width of any of the corridors in question, there remains just one possibility: the corresponding cancelling pair must belong to the non-Q_η-direction, and the exit point of the second of the letters of this pair could then be in Phase 2 (or in Phase 3 after having gone through a non-trivial monotony interval in Phase 2). In such a situation the corresponding picture has to look, apart from assumptions that can be made without loss of generality, like Fig.p_2 does.
Without loss of generality we assume that the corridors the gave the letters of our cancelling pair is horizontal, and that the the corresponding tangency is taken on the lower boundary. With respect to 4.18 we need only discuss the situation where the segments of our paths fall apart before, during, or at least at the end point of their phase of tangency, since otherwise the arguments of 4.18 would give the desired result already. Since the first of the letters of the cancelling pair still belongs to Phase 1, it is clear that the paths have to coincide at least before they go through a segment or point that is tangent to the lower boundary of our corridor. Our proof will be based on showing that, even if the length of the segment that both paths spend on the lower boundary of our corridor need not to coincide any more, the fact that both paths are represented by a reduced word sequence suffices to conclude that the corresponding cancelling pair appears in both cases, or in none of them.
This can be seen if we take a closer look at our geometrical situation: If any of our paths at the end of its phase of tangency moves a tiny bit up to go inside the corridor, according to the assumption of 4.19 it must have fallen apart from the other path already, and hence it is already in Phase 2. Thus our path is bound by the corresponding monotony conditions, and hence the moving away from the horizontal boundary line must already be a part of the Phase-2 monotony. This implies that according to the adjustment of our picture, Q_η must be found somewhere in the upper region of it, perhaps even outside. Recall that we are discussing shortest paths, i.e. there must be some holes in the lower part of our horizontal corridor, since otherwise our paths would not perform the non-monotonous behaviour with respect to the vertical coordinate that is implied by the assumption of them having a tangency. Our paths have to pass some of these holes on their lower side. In C.2.5/D.10.9 we had discussed that there are essentially two situations for such a tangency: There could exist (perhaps just on a much higher indexed lattice) a hole that is tangent to the boundary, and if such holes appear, then the cancelling pair as discussed in this paragraph, would have to appear in the corresponding word sequences also. On the other hand, if there should just exist a sequence of non-tangent holes that are accumulating to the lower boundary, then according to D.10.9(iii) our cancelling pair of letters would not appear in our word sequences. Note that, even if in this situation a geometrically shortest path might just take on a single point on the corridor boundary, for the geometrical situation under discussion a path which corresponds to a reduced word sequence might even take on an

entire segment on the corridor boundary (cf. D.3.10). However, we should also note that as long as our paths are passing one of our holes on its lower side inside our corridor, then the corresponding path is still in Phase 1 and hence has to coincide with the other paths of discussion: For, according to the adjustment of our coordinate system and to the geometrical interpretation of the characterizing condition of Phase 2 (cf. 3.3), $v(t)$ can only be in Phase 2 when for each $s \in [t, b_v]$ one can start going vertically upward at $v(s)$ and reach Q_η that way without coming upon any hole. And this condition gives that our paths have to pass below the same holes during their phase of tangency. This in principle implies what has to be proved here: namely that a cancelling pair of letters either appears in both word sequences, or in none of them: We get this result, since the holes that force our paths inside the corridor of consideration to their non-monotone behaviour are the same for both paths and hence are either both times of Type D.10.9(ii), or both times of Type D.10.9(iii).

(Note that that this results appears natural from another point of view, also: If one of our paths should pass below more holes than the other during its phase of tangency and hence pass the corresponding holes on a different side as the other while moving up to Q_η, then these paths could not be η-close any more.) □

Propositions 4.15–4.19 close the gap that was left in the Proof of Lemma 4.14. □

§5. Concluding the proof of the Lifting Theorem

5.1 Lemma: *Let i_0, ω and ω' be as in 3.2, and compute for both of these word sequences the associated path in its synchronous parametrization. In that context the notation as introduced in 3.2 refers to the what is constructed with respect to ω, while the corresponding objects that refer to ω' all are denoted by attaching a prime ("′"). Then we have that $|\varphi_x(a_{i,j}) - \varphi'_x(a'_{i,j})| < 32 \cdot \eta$ for all $1 \leq j \leq min\{k_i, k'_i\}$ and for all $i \leq i_0$.*

On the Proof: First observe that the letters of ω and of ω' for which the nodes $a_{i,j}$ give the corridor exit points are in most cases the same by 4.14. There is a single exception: if $k_i \neq k'_i$, then by 4.14 $|k_i - k'_i| = 1$. In that case for $j = \min\{k_i, k'_i\}$ the corresponding node $a^{(\prime)}{}_{i,j}$ refers for the longer word ω_i or ω'_i to the last letter, while for the shorter word it just refers to the exceptional node $a^{(\prime)}{}_{i,j} = 1$, i.e. to what the corresponding path does after having completed its last act of crossing an $\mathcal{N}_i$-corridor. This situation has firstly been described in 4.14/4.15 already: Thanks to the different behaviour of our paths their end phase, it may happen that one of them completes an act of crossing but the other does not. However, even in that case we have that $u\|_{[a_{i,j-1}, a_{i,j}]}$ and $u'\|_{[a'_{i,j-1}, a'_{i,j}]}$ must be fairly close to each other. In the subsequent steps of this lengthy proof we will analyze that they must actually be so close to each other that the estimate given in this lemma follows. The first of these steps will consist on an analysis of the $\Delta_{i,j}$-summands which by 2.5(1) are one of the crucial ingredients for our functions φ_x:

5.2 Proposition: *Assumptions and notations as in 5.1 and in 2.5(1)). Then for each level i we have coincidence between the dashed and the undashed $\Delta^{(\prime)}{}_{i,j}$-values for all j except the last — where instead we have that the absolute value of the difference is bounded by $6 \cdot \eta$.*

Proof: According to the principles as discussed in 4.14/4.15 only for the last $\Delta_{i,j}{}^{(\prime)}$-summands (i.e. with the highest j-indices) we may expect a difference, since as long as the corresponding node-intervals $[a^{(\prime)}{}_{i,j-1}, a^{(\prime)}{}_{i,j}]$ belong to Phase 1, the corresponding path-segments coincide and so do the $\Delta^{(\prime)}{}_{i,j}$. There are two situations for the last of the $\Delta^{(\prime)}{}_{i,j}$ at which we should look separately. Let $j_{\max} := \min\{k_i, k'_i\})$. By 4.14, in any case the nodes $a^{(\prime)}{}_{i,j_{\max}}$ belong to the same letter and $a_{i,j_{\max}-1}$ and $a'_{i,j_{\max}-1}$ have to lie on the same boundary line of a corridor. Hence for both of our nodes the constant coordinate of this boundary line is either inside the range of values spanned by Q_η or outside. In the following we discuss these cases separately when analyzing the possible difference of $\Delta'_{i,j} - \Delta_{i,j}$.

(i) $a_{i,j_{\max}-1}$ and $a'_{i,j_{\max}-1}$ might lie on a boundary line of a corridor whose constant coordinate is outside the range of values spanned by Q_η, so that at these points u and u' are still in Phase 1 (or they may be in Phase 2, if the Q_η-coordinate is referred to). This implies that two of the four $\Delta^{(\prime)}{}_{i,j_{\max}}$-summands are the same for $\Delta_{i,j_{\max}}$ and for $\Delta'_{i,j_{\max}}$ (it depends on the precise situation whether this are the max-summands or the min-summands of 2.5(1)), and the other two fall into the range of values spanned by Q_η. Hence each of the corresponding summands can at most differ by $2 \cdot \eta$, so that we get that $|\Delta_{i,j_{\max}} - \Delta'_{i,j_{\max}}| \leq 6 \cdot \eta$ in this case.

(ii) Now we assume that $a^{(\prime)}{}_{i,j_{\max}-1}$ is already inside the range of coordinate values spanned by Q_η. This case may be treated by the same arguments as the first one if on $[a^{(\prime)}{}_{i,j_{\max}-1}, 1]$ our paths should mess around and take on values outside the range spanned by Q_η (since in that case $a^{(\prime)}{}_{i,j_{\max}-1}$ would have still belong to Phase 1), and hence we restrict our considerations to the case where $a^{(\prime)}{}_{i,j_{\max}-1}$ does already belong to Phase 2 for the non-Q_η-coordinate or to Phase 3. According to our assumption on i (cf. 4.14) and to the definition of η in 3.2 it is clear that $u^{(\prime)}|_{[a^{(\prime)}{}_{i,j_{\max}-1},1]}$ cannot perform a completed act of crossing any more. The construction principle of our nodes in 2.2 then forces that $a_{i,j_{\max}} = 1 = a'_{i,j_{\max}}$. In that case we must at first sight take into account that two of our $\Delta^{(\prime)}{}_{?,?}$-summands might differ from each other, namely the $\Delta^{(\prime)}{}_{i,j_{\max}-1}$ and the $\Delta^{(\prime)}{}_{i,j_{\max}}$-values. A closer look then gives that the first actually have to agree: Even though one can construct geometrical situations where $u(a_{i,j_{\max}-1})$ does not coincide with $u'(a_{i,j_{\max}-1})$, what is contributed to the corresponding summand of $\Delta^{(\prime)}{}_{i,j_{\max}-1}$ is just the constant coordinate from then boundary line, and hence this is the same value in the dashed or in the undashed case. Further note that, since $u^{(\prime)}|_{[a^{(\prime)}{}_{i,j_{\max}-2}, a^{(\prime)}{}_{i,j_{\max}-1}]}$ involves a completed act of crossing some corridor of $\mathcal{N}_i$, no matter whether $u^{(\prime)}|_{[a^{(\prime)}{}_{i,j_{\max}-2}, a^{(\prime)}{}_{i,j_{\max}-1}]}$ is a passage or a return (cf. 2.7(i)/(ii)) the constant coordinate from the boundary line that contains $u^{(\prime)}(a^{(\prime)}{}_{i,j_{\max}-1})$ is, depending on the orientation of the passage or return (cf. 2.7), the maximal or minimal coordinate, also, since otherwise the act of crossing would have to be completed earlier. Now, thanks to the completed act of crossing,

the other two summands for $\Delta^{(\prime)}{}_{i,j_{\max}-1}$ must be far outside of the range of values spanned by Q_η and hence contributed by Phase 1 or by the coinciding monotonous behaviour of our paths in Phase 2, so that they must coincide also. This gives that $\Delta'_{i,j_{\max}-1} = \Delta_{i,j_{\max}-1}$ in this case, also (in the preceding Case (i)) it was trivial).
For the $\Delta^{(\prime)}{}_{i,j_{\max}}$-summands, we must in particular analyze whether the fact that $a^{(\prime)}{}_{i,j_{\max}-1}$ falls into a region where u and u' can already differ from each other might not result in a situation, where $a_{i,j_{\max}-2}$ and $a'_{i,j_{\max}-2}$, come out differently, also, so that unlike as in the preceding item (i), all four summands of 2.5(1) are affected and hence we obtain $12 \cdot \eta$ as an estimate for $\Delta_{i,j_{\max}}$ and $\Delta'_{i,j_{\max}}$, instead of $6 \cdot \eta$. However, the following arguments give that we actually get $6 \cdot \eta$ in the case, also: Even if all four summands of 2.5(1) should come out differently in the dashed or undashed version (in fact, we have just proven that it can be at most three), since the range of relevant coordinates is at most that which is spanned by Q_η, for each "max-min"-difference of 2.5(1) $2 \cdot \eta$ is an upper estimate, and so we get $6 \cdot \eta$ in this case also.

Hence we have now observed that for each relevant i only the last $\Delta^{(\prime)}{}_{i,j}$-summand can come out differently for the dashed and for the undashed version, and that this difference is bounded by $6 \cdot \eta$. □

5.3 On the inductive set-up of the proof of Lemma 5.1.

What remains to be done after the successful completion of the first step of the proof of Lemma 5.1 by Proposition 5.2 is to trace what happens to these $6 \cdot \eta$-differences in the course of the inductive process 2.4 that gave us the $\varphi_x{}^{(\prime)}$ -values. It has to be ruled out that in the course of this process the $(6 \cdot \eta)$-differences accumulate to too large values. Hence the second part of this proof is about to show that in most of the cases this inductive process only applies interpolation processes to these differences so that we can come up with some fixed multiple of η as a bound of $|\varphi_x(a_{i,j}) - \varphi'_x(a'_{i,j})|$. At the end we want to prove that

$$(0) \qquad |\varphi_x(a_{i,j}) - \varphi'_x(a'_{i,j})| \leq K \cdot \eta \text{ with } K = 32,$$

but it will be vital to note that a lot of estimates will also work for $K = 16$. So let us now look at the Steps of the inductive process 2.4:

Zero-Step: All $\varphi_x{}^{(\prime)}(s^{(\prime)}{}_{0,j})$ satisfy the corresponding condition, since due to $s^{(\prime)}{}_{0,j} = 1$ these values just arise by computing sums $\sum_{j'=1}^{j} \Delta^{(\prime)}{}_{0,j}$. And by 5.2 at most the last of these summands could come out differently in the dashed and in the undashed version, with the difference bounded by $6 \cdot \eta < K \cdot \eta$.

Inductive step: For the sake of keeping our notation simple, we restrict ourselves to giving the undashed formulae in the following: If the first index is not zero, then the definition scheme of the $\varphi_x(a_{i,j})$-values, as given by 2.4 may be understood as a weighted arithmetic average of type

$$\frac{\gamma_1 \cdot \varphi_x(a_{i,j'}) + \gamma_2 \cdot \varphi_x(a_{i,j''})}{\gamma_1 + \gamma_2} \quad \text{with} \quad \gamma_2 = \sum_{\nu=j'+1}^{j} \Delta_{i,\nu} \text{ and } \gamma_1 = \sum_{\nu=j+1}^{j''} \Delta_{i,\nu}.$$

Hence we have to compare this expression with the dashed version. For that purpose we introduce

$$(1)\qquad \delta_1 := \varphi'_x(a'_{i,j'}) - \varphi_x(a_{i,j'})\ ,\ \delta_2 := \varphi'_x(a'_{i,j''}) - \varphi_x(a_{i,j''})\ \text{ and }\ \delta_3 := \gamma'_1 - \gamma_1.$$

Note that we have that $\gamma'_2 - \gamma_2 = 0$, since 5.2 imply that at most the last of the $\Delta^{(\prime)}{}_{i,j}$-summands can differ in the dashed and in the undashed versions. In addition, we have that $|\delta_\nu| < K \cdot \eta$ for $\nu = 1, 2, 3$. This follows for $\nu = 1, 2$ from the induction hypothesis, and for $\nu = 3$ it can be deduced similarly as the corresponding estimate of the Zero-step. The following computation will then give that the absolute value of the difference between the dashed and undashed versions of the values for $\varphi_x{}^{(\prime)}(a^{(\prime)}{}_{i,j})$ is also bounded by $K \cdot \eta$:

$$\begin{aligned}
&\frac{\gamma'_1 \cdot a' + \gamma_2 \cdot b'}{\gamma'_1 + \gamma_2} - \frac{\gamma_1 \cdot a + \gamma_2 \cdot b}{\gamma_1 + \gamma_2} = \\
&= \frac{(\gamma_1 + \delta_3) \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2 + \delta_3} - \frac{\gamma_1 \cdot a + \gamma_2 \cdot b}{\gamma_1 + \gamma_2} = \\
&= \frac{(\gamma_1 + \delta_3) \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2 + \delta_3} - \frac{\gamma_1 \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2} + \\
&\qquad + \frac{\gamma_1 \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2} - \frac{\gamma_1 \cdot a + \gamma_2 \cdot b}{\gamma_1 + \gamma_2} = \\
&= \Big(\frac{(\gamma_1 \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2} + \frac{\delta_3 \cdot (a + \delta_1)}{\gamma_1 + \gamma_2} \Big) \cdot \frac{\gamma_1 + \gamma_2}{\gamma_1 + \gamma_2 + \delta_3} - \\
&\qquad - \frac{\gamma_1 \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2} + \frac{\gamma_1 \cdot \delta_1 + \gamma_2 \cdot \delta_2}{\gamma_1 + \gamma_2} = \\
&= \Big(\frac{\gamma_1 + \gamma_2}{\gamma_1 + \gamma_2 + \delta_3} - 1 \Big) \cdot \frac{\gamma_1 \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2} + \\
&\qquad + \frac{\delta_3 \cdot (a + \delta_1)}{\gamma_1 + \gamma_2 + \delta_3} + \frac{\gamma_1 \cdot \delta_1 + \gamma_2 \cdot \delta_2}{\gamma_1 + \gamma_2} = \\
&= \frac{\gamma_1 \cdot \delta_1 + \gamma_2 \cdot \delta_2}{\gamma_1 + \gamma_2} + \frac{-\delta_3}{\gamma_1 + \gamma_2 + \delta_3} \cdot \Big(\frac{\gamma_1 \cdot (a + \delta_1) + \gamma_2 \cdot (b + \delta_2)}{\gamma_1 + \gamma_2} - (a + \delta_1) \Big) = \\
&= \frac{\gamma_1 \cdot \delta_1 + \gamma_2 \cdot \delta_2}{\gamma_1 + \gamma_2} - \delta_3 \cdot \frac{((b + \delta_2) - (a + \delta_1)) \cdot \gamma_2}{(\gamma_1 + \gamma_2) \cdot (\gamma_1 + \gamma_2 + \delta_3)} = \\
&\qquad = \frac{1}{\gamma_1 + \gamma_2} \cdot \Big(\gamma_1 \cdot \delta_1 + \gamma_2 \cdot (\delta_2 - \delta_3 \cdot \frac{(b + \delta_2) - (a + \delta_1)}{\gamma_1 + \gamma_2 + \delta_3}) \Big)
\end{aligned} \tag{2}$$

Based on these calculations it turns out that it is fairly easy to complete the proof of Lemma 5.1 provided that our δ_i satisfy another estimate, namely that

$$(3)\qquad |\delta_2 - \delta_3\ | < K \cdot \eta.$$

Hence we will below explain how to bring the proof to an end by using the estimate, before proving it in 5.5–5.7.

5.4 Proposition: *Estimate 5.3(0) holds for $K = 16$ or $K = 32$, provided all δ_3 satisfy 5.3(3) with the same K.*

Proof: Recall that by 2.4(2) and 2.5(1) the ratio between $\varphi_x(a_{i,j''}) - \varphi_x(a_{i,j'})$ and $\sum_{j=j'+1}^{j''} \Delta_{i,j}$ was called $s_{i,j'}$, and since by 2.6 the $s_{i,j'}$-values are always bigger than or equal to one, we now conclude that the sum of these $\Delta_{i,j}$-values cannot be smaller than the corresponding difference of the φ_x-values. A similar estimate holds for the corresponding dashed values that are in Formula 5.3(2) represented as sums of undashed values and the corresponding $\delta_?$-correction term, where, in addition, the ${\varphi^{(\prime)}}_x$-values are replaced by the variable $a^{(\prime)}$ and $b^{(\prime)}$. The above arguments give that the ratio which appears after the last "="-sign in 5.3(2) as the coefficient of δ_3 is a positive real that is smaller than or equal to one. Having made this observation we can now deduce that the absolute value of the parenthesized term after the last "="-sign in Formula 5.3(2) is bounded by $K \cdot \eta$. This follows, since obviously the corresponding value lies in the interval bounded by δ_2 and by $\delta_2 - \delta_3$, both of which values have by 5.3(1) and 5.3(3) a correspondingly bounded value. Having obtained that estimate for the parenthesized term, the entire values of 5.3(2) appears as a weighted arithmetic average of the parenthesized term on the one hand and of δ_1 on the hand. Consequently the entire 5.3(2)-term has an accordingly bounded absolute value, and that was the desired estimate that completes the inductive step and the whole proof of this proposition. □

Recall that 5.3(3) and the above Proposition 5.4 imply Lemma 5.1. Hence having completed the above proof, it now just needs to add a proof of our estimate 5.3(3). The next proposition is the first step of this estimate.

5.5 Proposition: *Assumptions and notations as in 5.1, 5.3(1), and 5.3(3). If up to some level i all $|\delta_2$-$\delta_3|$ are bounded by $K \cdot \eta$ from above, and if $k_i = k'_i$, then the same bound also holds for $|\delta_2 - \delta_3|$ on the level $i + 1$.*

Proof: By having assumed that $k_i = k'_i$, we know that ω_i and ω'_i have the same number of horizontal letters. Hence we know that the rightmost nodes of our discussion satisfy that $a'_{i,k'_i} = 1 = a_{i,k_i}$. This observation is essential for the case of this proposition, since only the final interval (i.e. the case where $j'' = k_i$ or k'_i) actually needs discussion. For, in any other case (cf. the Zero-Step) the according ${\gamma^{(\prime)}}_\nu$-values do not differ in the dashed and in the undashed version, and hence $\delta_3 = 0$ so that 5.3(3) is an immediate consequence of 5.3(1).
Hence δ_2 and δ_3 do both reflect the difference between the behaviour of u and of u' inside or in their approaching of Q_η. At first sight one might expect that we always have $\delta_2 = \delta_3$ in our case, since both of these values reflect the different behaviour of the same two paths in their terminal segment. However, it is not quite as simple, since δ_2 and δ_3 are in fact both obtained from considering the final segment — but according to different segmentations! — In that context recall that $\delta_2 = \varphi'_x(a'_{i,j''}) - \varphi_x(a_{i,j''})$, where the corresponding node $a_{i,j''}$, since it is the last of our discussion, has been fixed on the first step of the inductive process for defining φ_x already (cf. 2.4). Hence δ_2 is the corresponding difference of ${\Delta^{(\prime)}}_{1,?}$-values only; it has not seen further corrections due

to interpolation or $s^{(\prime)}{}_{?,?}$-values. According to 2.4(2)/5.2 δ_3 can also be regarded as a difference of $\Delta^{(\prime)}{}_{?,?}$-values, but here the first index is i when discussing the δ-correction terms of the ith inductive step. If then we replace the second index of our $\Delta^{(\prime)}{}_{?,?}$-values by "last" (for having not to bother about the different adjustments of these second indices with respect to the first ones), we get that the difference we are discussing here is

$$\delta_2 - \delta_3 = \Delta'_{1,\text{last}} - \Delta_{1,\text{last}} - \Delta'_{i,\text{last}} + \Delta_{i,\text{last}} \tag{1}$$

We should now recall that by 2.5(1) each of the $\Delta^{(\prime)}{}_{?,?}$ terms is a sum/differences of four terms again, i.e. we are discussing an alternating sum of 16 terms, in principle. We do not want to introduce variables for denoting these terms, hence we just recall that the four terms for each of the $\Delta^{(\prime)}{}_{?,?}$ are the "entrance coordinate" (i.e. $u_1{}^{(\prime)}(a_{1,\text{last}-1})$ or $u_1{}^{(\prime)}(a_{i,\text{last}-1})$ in our case), the "exit coordinate" (where the difference between entrance coordinate and exit coordinate was in any case adjusted in such a way that it was a positive value), and the maximal and minimal value of $u^{(\prime)}{}_1$ on the relevant interval. Now, why do these 16 terms cancel up to at most $16{\cdot}\eta$? – In this context a similar distinction of cases as 5.2(i) and 5.2(ii) before is required. The particular importance of discussing the according cases separately is given, since the use of max- and min-constructions in 2.5(1) makes it in general impossible to predict whether some exit- or entrance-coordinate appears with positive or with negative sign in 2.5(1) and hence in (1), also. However, if we know the corresponding coordinates to be outside the range of coordinates spanned by Q_η, then we know them to agree for the different paths and hence we can conclude that the adjustments of sign for the corresponding paths is the same, also. In the other case we do not have this information, but we will try to play similar arguments as before in Case 5.2(ii).
However since, other than in the original 5.2(i) and 5.2(ii), this time more paths have to be considered and the number of combining the various 5.2(i)- and 5.2(ii)-assumptions does grow exponentially, we resist discussing all these cases separately. Instead of that we just give below the list of arguments such that for any of our cases an appropriate collection of arguments of that list gives the desired estimate:

5.6 Summary of the estimates that complete the proof of Proposition 5.5:

(1.) First note that in the case with the biggest uncertainty of signs, i.e. when all relevant coordinates fall inside the Q_η-region, the signs inside of Formula 2.5(1) are adjusted in such a way that the corresponding difference of the four values can give at most $4{\cdot}\eta$ for each of the four terms of 5.5(1). Hence we get the desired estimate of $16{\cdot}\eta \leq K \cdot \eta$ in this case. Roughly speaking, the discussion of the other cases will rest on showing that thanks to the agreeing of u and u' in Phase 1 and to their going monotonously over the same interval of Q_η-coordinates in Phase 2, those coordinates of the dashed paths that lie outside Q_η cancel against the corresponding coordinates of the undashed path, and that those coordinates which lie inside Q_η and hence do not need to cancel never contribute more than $4{\cdot}\eta$ per $\Delta^{(\prime)}{}_{?,\text{last}}$-term. However, we will be more precise in the following items:

(2.) If for one of our indices 1 or i the entrance coordinates should lie outside Q_η, they agree for the dashed and undashed path, and then the signs are adjusted in such a way that they cancel completely. The same effect also holds for the max-coordinates and the min-coordinates.

(3.) The following cancelling effect may hold for the exit-coordinates: Since they are the end-point of our paths, they need not to coincide for the dashed and undashed path, but for each of our two paths the exit point of the term that is associated with the first index 1 coincides with that exit point that corresponds to the first index i. A similar coincidence can been seen for the max- and min-coordinates in those cases where the max- and min-coordinates are taken on inside Q_η. Of course, in the case as described in this item it much depends on the adjustment of signs whether the corresponding coinciding values actually cancel.

(4.) Note that the coincide of the entrance coordinates as described in (2.) is independent from the fact whether they lie inside or outside Q_η, since they lie in any case on the same corridor boundary. However, if they are inside Q_η we do in general not know whether the signs are adjusted accordingly (cf. (3.)).

(5.) If our two paths approach Q_η only from one side and if the $u^{(\prime)}(a^{(\prime)}{}_{?,\mathrm{last}-1})$-points are still outside Q_η, then the effects as described in (2.) and (3.) suffice for a 100%-cancellation, i.e. the difference 5.5(1) is zero in this case.

(6.) Hence let us now consider the situation, where the corridor-entrance coordinate for the first index 1 lies below the Q_η-interval, but for the first index i it lies above. This is a case where we have to fear that signs are adjusted in such a way that the (3.)–cancellations will not take place. However, in this case 6 out of 8 max-/min term cancel due to the effect as described in (2.) (at least, we know the index-1 segments to run over the entrance point of the index i, also), and hence we have in this case only to deal with 4 exit coordinates and the two remaining max-/min-terms. The first give two differences, each of which can be at most $2\cdot\eta$, and we get one difference of remaining max-/min-terms. Taking into account that in 2.5(1) the coefficients "2" do only occur before max-/min- terms, we see that our differences can at most accumulate to $8\cdot\eta$.

(7.) We will not continue to discuss the remaining situations in that spirit, i.e. not try to describe all the situations where for the dashed and undashed paths the maxima and minima and entrance points could lie, whether they do or do not coincide for the indices i and 1 and whether our paths, even if the entrance point should already be in the Q_η region, are going to leave that region again or whether they do not. In order to close this discussion, let us view the situation as follows: The 16 terms of 5.5(1) can be grouped into 8 differences, and according to the preceding arguments each of them can give at most $2\cdot\eta$. If one takes into account, that half of the summands of 2.5(1) have the coefficient "2", this would give $24\cdot\eta$. What has be verified in each of the case separately, is that we find a sufficient amount of complete cancellation and lucky adjustment of signs, that one always comes down to $16\cdot\eta$. It does just require a recycling of the arguments given so far, to do this job and to complete the proof of

5.5 that way. □

Having completed the discussions in 5.5 and 5.6, we are now ready to approach a general proof of our Lemma 5.1:

5.7 Proposition: *Assumptions and notations as in 5.1, 5.3(1), and 5.3(3). Then all* $|\delta_2 - \delta_3| < 32 \cdot \eta$.

Proof: By using 5.5, 5.6 and the same inductive mechanism as set up in 5.3, we in principle get 5.3(3) for $K = 16$ if on each level we have that $k_i = k_i'$. Hence only the opposite case is left to be proven below: If k_i' and k_i should not be equal, then by 4.14 they can differ by at most one. Without loss of generality assume that $k_i + 1 = k_i'$. In this case we need only to compare the values $\varphi_x(a_{i,j})$ and $\varphi'_x(a'_{i,j})$ for $j \in \{1, 2, 3, \ldots, k_i\}$; however, the situation now is more complicated, since $\varphi'_x(a'_{i,k_i}) - \varphi_x(a_{i,k_i})$ cannot be simply identified with $\Delta'_{1,\text{last}} - \Delta_{1,\text{last}}$ as in 5.5/5.6. For, $a_{i,k_i} = 1$, and hence $\varphi_x(a_{i,k_i})$ is still a value that is fixed on the zero-step of the induction already, but $\varphi'(a_{i,k_i})$ may been fixed on a later inductive step by some interpolation process. We denote by i'' the lowest first index for which at the parameter value a'_{i,k_i} some node already is defined. Roughly speaking, we will not be able to prove a similar estimate as in 5.5/5.6 for our situation here, and hence we will have to use a more primitive estimate instead here, namely that the absolute value of a difference like $\delta_2 - \delta_3$ can be at most the double of the corresponding estimate for δ_2 and for δ_3, i.e. we want to use an estimate that is worse by a factor "2". However, we still need to take a closer look at our inductive process, since we have to make sure that in the course of the inductive process there can be at most one place at which this factor "2" comes into play (so that our estimate is not spoiled by some power of two whose size could not be controlled any more). In this context we should observe that in all cases as discussed above we appeared to be able to prove a better estimate than the claimed "$32 \cdot \eta$", namely "$16 \cdot \eta$". When tracing our inductive process for occurrences of such factors "2", we get the following pictures:

(i) For $i < i''$ all works as smooth as in 5.5/5.6. I.e. for those first indices our dashed and undashed path have the same numbers of nodes with their final nodes being 1 and hence their nodes $\varphi^{(\prime)}{}_x(a^{(\prime)}{}_{i,\text{last}})$ have been fixed on the zero-step of the induction already. Hence from the arguments above (including 5.5/5.6) we get, if the first indices are smaller than i'' that the δ-correction terms are all bounded by $16 \cdot \eta$, indeed.

(ii) Now consider the i''^{th} step of our induction. Here a phenomenon happens that has already been explained in 4.14/4.15: We have to assume that Q_η is intersected by some boundary line of an $\mathcal{N}_{i''}$-corridor, and that the dashed path behaves in such a way in Phases 2 and 3 it completes the crossing of the corresponding corridor, while the undashed path does not. Hence the difference $\varphi'_x(a'_{i,k_i}) - \varphi_x(a_{i,k_i})$ is the first time to be computed for that level index i'', where $a_{i,k_i} = 1$ and hence $\varphi_x(a_{i,k_i})$ had been defined on the zero-step of the induction already, while $\varphi'_x(a'_{i,k_i})$ is defined now on the i''^{th} level by some interpolation process. Yet the $16 \cdot \eta$-estimate holds for this difference also. This is, because the nodes that are to be interpolated to give a'_{i,k_i} have been fixed on the preceding level, already, and hence the discussion of this process still falls

under 5.5/5.6. In addition, note that even the keeping fix of $\varphi_x(a_{i,k_i})$ between the level indices $i''-1$ and i'' can be regarded as a trivial interpolation process, namely a process with $\gamma_1 = 0$. But since $u'|_{[a'_{i,k_i},1]}$ is contained inside Q_η, the corresponding δ_3-value for comparing the ratios of that trivial interpolation process with the one that defined $\varphi'_x(a_{i,k_i})$ satisfies all conditions of 5.5. Hence the $(16{\cdot}\eta)$-estimate for that difference, as claimed above, and for any other difference of $\varphi_x{}^{(\prime)}$-values that is to be computed on the i''^{th} level, also. There is just one bad feature to note: If our crucial difference $\varphi'_x(a'_{i,k_i}) - \varphi_x(a_{i,k_i})$ appears as a δ_2-coefficient in some estimate for a higher level index, we do not know whether the corresponding difference $\delta_2 - \delta_3$ will still satisfy the $(16{\cdot}\eta)$-estimate. However, the $(32{\cdot}\eta)$-estimate is clearly fulfilled.

(iii) Now consider the situation for $i > i''$. First observe that the situation of (ii) cannot appear again, at least not for the levels to be discussed here for proving Lemma 5.1, i.e. for $i \le i'$. For, this situation requires that a corridor boundary intersects Q_η, and since the width of Q_η is much smaller as the width of the $\mathcal{N}_{i'}$-corridors, there cannot be more crucial boundary lines around. Hence there cannot be an accumulation of that phenomena that have been described at the beginning of 5.7. Hence, what remains to be proved is that with the arguments as given above a $(32{\cdot}\eta)$-estimate clearly can be extended throughout the remaining steps of the induction:
This is not so hard to see: The discussion of the preceding item can be interpreted as giving the zero-step for such an induction from i'' to i', and if we assume a $(32{\cdot}\eta)$-estimate for δ_1 and δ_2 and that we are not in the final interval and hence $\delta_3 = 0$, then similarly as in 5.2(i)–5.2(ii) we can again deduce a $(32{\cdot}\eta)$-estimate for our new correction term. And, if we are in the final interval, then the δ_2-term we have to work with is precisely $\varphi'_x(a'_{i,k_i}) - \varphi_x(a_{i,k_i})$, and according to the discussion of the preceding item (ii) this term satisfies even some $(16{\cdot}\eta)$-estimate. Hence by 5.2 δ_3 satisfies the same inequality so that $|\delta_2 - \delta_3| < 32 \cdot \eta$. Hence the $(32{\cdot}\eta)$-estimate can inductively be extended to all indices $i = i'' + 1, \ldots, i'$. □

Having justified the estimate 5.3(3) in all cases by 5.7 now, we are now able to conclude that the computations 5.3(2) indeed give that a $32 \cdot \eta$ for the differences of φ_x and φ'_x-values persist on all relevant index levels $i \le i_0$. Hence the proof of Lemma 5.1 is complete. □

Having obtained the estimate of 5.1, we are now ready to approach the lemma which shall give the final estimate in order to reach the goal of this paper:

5.8 Lemma: *Assumptions and notations as in 3.2. Then for all* $t \in [0, \max(b_v, b_{v'})]$ *we have that* $\|v'(t) - v(t)\| < \varepsilon$.

On the Proof: The purpose of the first part of this proof is to show that (cf. 4.14/4.15) for all parameter values t our path can have completed at most one act of crossing more or less in each of the basic directions as the path v'. For that purpose we count $\#\{j \mid \varphi_x(a_{i_0,j}) \le t\}$ and compare it for each t with $\#\{j \mid \varphi'_x(a_{i_0,j}) \le t\}$. We want to show that (with some minor exceptions in case t belongs to Phase 3 of our paths already), that these numbers can at most differ by one. Hence, let us assume that their

difference would be more than one. I.e., without loss of generality, for some t we either have the inequality

$$\varphi_x(a_{i_0,j}) \leq t < \varphi'_x(a'_{i_0,j-1}) \tag{1}$$

or we have

$$\varphi'_x(a'_{i_0,j}) \leq t < \varphi_x(a_{i_0,j-1}). \tag{2}$$

We will proceed by showing that (1) and (2) will yield a contradiction in their standard cases (5.9 and 5.10, resp.), while understanding that the remaining cases will for other reasons not contradict to the claim of 5.8. Then we will be able to finally complete the proof of 5.8 in 5.11 by arguing why the first sentence of this paragraph "On the Proof" actually implies this lemma.

5.9 Proposition: *Assume that apart from the general assumptions of 5.8 in addition 5.8(1) holds. If this does not already imply a contradiction, it does for other reasons not contradict to the claim of 5.8.*

Proof: We aim at computing some estimate for $\varphi_x(a_{i_0,j-1})$ for these assumptions. By postponing Phase-3 phenomena to the following paragraph, we may use that $u_{\restriction [a_{i_0,j-1}, a_{i_0,j}]}$ performs at least one completed act of crossing a corridor. This gives (even in case $j = 1$, cf. 1.4(i)) that the crucial $\Delta_{i_0,j}$-summand is at least $1/3^{i_0}$ by 2.8. Consequently, by 5.8(1) and by 2.4(2) the corresponding difference $\varphi'_x(a'_{i_0,j-1}) - \varphi_x(a_{i_0,j-1})$ must be at least $\frac{1}{s_{i_0,?}} \cdot 1/3^{i_0}$. By 3.2(1)–(3) this is still bigger than $32 \cdot \eta$ which was by 5.1 the utmost value for such a difference.
The only subcase of in the proof of this proposition which is not sufficiently treated by the above arguments is that where $[a_{i_0,j-1}, a_{i_0,j}]$ is the last node-interval, since on that interval our path need not perform a completed act of crossing a corridor. However, in that case our Formula 5.8(1) does not imply that v has at the parameter value t passed two vertical $\mathcal{N}_{i_0}$-nodes more than v' has, and hence 5.8(1) does not contradict to what is the claim of the first sentence of this proof in that case. Hence this case is not relevant for us. □

5.10 Proposition: *Assume that apart from the general assumptions of 5.8 in addition 5.8(2) holds. If this does not already imply a contradiction, it does for other reasons not contradict to the claim of 5.8.*

Proof: Note that the required estimate in this case is not entirely symmetric to what had to be shown in 5.9, since in our definition of η in 3.2 only the values $s_{i_0,?}$ of our fixed path v had been taken into account, but not those of our free path v'. Hence, we cannot treat this case by interchanging the roles of the paths v and v'; instead we have to imitate the conclusions of 5.9 by again using some estimate for what could be a lower bound of $\varphi_x(a_{i_0,j}) - \varphi_x(a_{i_0,j-1})$. Unlike as in the preceding case we do not only have to deal with the insecurity, whether the corresponding segment of v does perform a completed act of crossing or whether it does not, but we cannot even be sure whether

the node $a_{i_0,j}$ is defined at all. Hence we have to take a look at all those possibilities separately:

(i) If $u_{\restriction [a_{i_0,j-1}, a_{i_0,j}]}$ performs a complete act of crossing, then in principle the same arguments as in 5.9 hold; this time the corresponding estimates give that $\varphi_x(a_{i_0,j})$ and $\varphi'_x(a'_{i_0,j})$ are further apart than permitted by 5.1.

(ii) If the corresponding segment of v does not complete its act of crossing, then only one situation might actually cause trouble, namely the situation when v' does complete its act of crossings for the node-interval $[a'_{i_0,j-1}, a'_{i_0,j}]$ of consideration. However, in that situation we can argue as follows: We must be in a situation as described in 4.14/4.15 where the dashed of our paths has an exceptional letter, but the undashed has not. Hence Q_η must intersect the corridor boundary that contains $u'(a'_{i_0,j})$, and in that Q_η-square the endpoint of u, i.e. $u(a_{i_0,j})$ must lie. Hence the corresponding segment of u does not complete its act of crossing, but it almost does. Instead of the corridor with $(\frac{1}{3})^{i_0}$, this time we can use $(\frac{1}{3})^{i_0} - 2 \cdot \eta$ as the corresponding estimate. This substitution does not overturn the corresponding estimate as given in 5.9 — in particular, recall that the differences $\varphi'_x(a'_{i_0,j}) - \varphi(a_{i_0,j})$ with such a set-up of nodes have also been discussed in Lemma 5.1, so that the corresponding $(32{\cdot}\eta)$-eta estimate holds in this case, also.

(iii) Let us now discuss the cases that had not been handled in the paragraphs (i) and (ii), i.e. the case where the "If"-clause, but not the main clause of the the first sentence of (ii) are fulfilled. However, in such a case it is not hard to see that Inequality 5.8(2) does not imply that there had to exist a φ_x-value where one of our paths has completed two acts of crossing more than the other. Hence all possibilities of (iii) (including the one that the node $a_{i_0,j}$ is not defined for φ_x at all), are irrelevant, similar as the corresponding subcases of Proposition 5.9. □

5.11 Completion of the Proof of Lemma 5.8:

Firstly we will verify the assertion that was stated in the first sentence of the proof-overview that was given for 5.8:

Assume that there would exist some parameter value t such that $v_{\restriction [0,t]}$ has seen two completed act of corridor crossings more than $v'_{\restriction [0,t]}$ has. Then we would see some inequality as

$$(1) \qquad \varphi_x(a_{i_0,j}) + \varphi_y(a_{i_0,j}) = \varphi(a_{i_0,j}) \leq t < \varphi'(a'_{i_0,j-1}) = \varphi'_x(a'_{i_0,j-1}) + \varphi'_y(a'_{i_0,j-1})$$

In the preceding paragraphs we explained how all those inequalities, that actually contradict to the claim of the first sentence of this proof in 5.8, yield a contradiction, roughly speaking, by computing an upper bound for $\varphi_x(a_{i_0,j-1})$ and showing that the difference to $\varphi'(a'_{i_0,j-1})$ is bigger than allowed by 5.1. The same line of arguments can now be played to Inequality (1) (and to all its symmetric variations); knowing that according to 2.15(ii) the additional summand φ_y belongs to a monotonous function is enough to imitate the above conclusions.

Hence we managed to complete the first step of this proof: The above difference of nodes can be at most one. Of course, the same estimate holds for the $b_{i_0,j}$-nodes. If we combine

this information with the fact that ω_{i_0} and ω'_{i_0} coincide up to at most one appendance or truncation at the end, we can conclude that the x-coordinates and the y-coordinate of $v'(t) - v(t)$ are bounded to what a path could have done if according to a valid reading of $\mathcal{N}_{i_0}$ it just seen zero or one corridor exit points more or less than the other path in any of the two basic directions. This implies that $v'(t)$ and $v(t)$ have to lie in the same (5×5)-block of $\mathcal{N}_{i_0}$-regions. Now in 3.2 i_0 had been adjusted in such a way that a (5×5)-block still fits into an $\frac{\varepsilon}{2}$-neighbourhood, and this gives that $\|v'(t) - v(t)\| < \varepsilon$. □

By having completed the proof of Proposition 5.11, now the proofs of Lemma 5.8 and of the Theorems B.3.2, A.5.6 and of Theorem A are also complete (cf. 3.2 and A.6.5).□

[Alx1] ALEXANDER,J.W.: "On the subdivision of the three-space by a polyhedron", Proc. Nat. Akad. Sci. USA 10 (1924), pp. 6–8

[Alx2] — , "An Example for a simple connected connected surface bounding a region which is not simply connected", Proc. Nat. Akad. Sci. USA 10 (1924), pp. 8–10

[B–S] Bogley, W.A. and SIERADSKI, A.J. : "Omega-Groups", "Universal Path Spaces" and "Weighted Combinatorial Group Theory for wild metric complexes", preprints, Oregon State University and University of Oregon, Corvallis and Eugene (Oregon, USA), May 1997

[C–C] CANNON, J.W. and CONNER, G.R. : "The Combinatorial Structure of the Hawaiian Earring Group", "The Big Fundamenta Group, Big Hawaiian Earrings, and the Big Free Groups" and "On the Fundamental Groups of One-Dimensional Spaces", preprints, Bringham Young University, Provo (Utah, USA), September 1997

[C–P] CIESIELISKI, Kr. and POGODA, Zd.: "The beginning of Polish Topology", The Mathematical Intellegencer, Vol.18, No.3, pp. 32–39

[Eda] EDA, Katsuya: "Free σ-Products and noncommutatively Slender Groups", J. Algebra, 148 (1992), pp. 243–263

[E–S] EILENBERG, Samuel and STEENROD, Norman: "Foundations of Algebraic Topology", Princeton Mathematical Series No. 15, Princeton University Press, Princeton (New Jersey), 1952

[ECHLPT] EPSTEIN, D.B.A., CANNON, J.W., HOLT, D.F., LEVY, S.V.F., PATERSON, M.S. and THURSTON, W.P.: "Word Processing in Groups", Jones & Bartlett, Boston, 1992

[Gr1] GRIFFTHS, H.B.: "The Fundamental Group of two Spaces with a Common Point", Quart. J. Math. Oxford (2), 5 (1954), pp. 175–190; Correction: ibid. 6 (1955), pp. 154–155

[Gr2] GRIFFTHS, H.B.: "Infinite Products of Semi-Groups and Local Connectivity", Proc. London Math. Soc. (3), 6 (1956), pp. 455–480

[Higm] HIGMAN, Graham: "Unrestricted free product, and varieties of topological groups", J. London Math. Soc.(2), 27 (1952), pp. 73–88

[Ho–Y] HOCKING, J.E. and YOUNG, G.S.: "Topology", Addison-Wesley Series in Mathematics, Addison-Wesley, Reading (Massachusetts), 1961

[Kö1] KÖNIG, Dénes: "Über eine Schlußweise aus dem Endlichen ins Unendliche", Acta Litterarum Acta Litt. Sc. R. Univ. Hung. Fr.J., 3 (1927), pp. 121–130

[Kö2] — , "Sur le correspondance multivoques d'ensmbles", Fund. Math. 8 (1926), pp. 114–134

[Kö3] — , "Theorie der endlichen und unendlichen Graphen", Mathematik und ihre Anwendungen in Monographien und Lehrbüchern, Akademische Verlagsgesellschaft, Leipzig, 1936

[Ly–S] LYNDON, R.T. and SCHUPP, P.E.: "Combinatorial Group Theory", Ergebnisse der Mathematik und ihrer Grenzgebiete No. 89, Springer, Berlin, 1977

[MM] MORGAN, John and MORRISION, Ian : "A van-Kampen-Theorem for Weak Joins", Proc. London Math. Soc. (3), 53 (1986), pp. 562–576

[PJS] PEITGEN, H.O., JÜRGENDS, H. and SAUPE, D.: "Chaos and Fractals", Springer, New York, 1992

[Pomk] POMMERENKE, Christian: " Boundary Behavior of Conformal Maps", Grundlehren der mathematischen Wissenschaften No. 299, Springer-Verlag, Berlin – New York, 1992

[Sier] SIERPIŃSKI, Wacław: "Sur une courbe Cantorienne qui contient une image biunivoquet en continue des courbe donnée, Compt. Rendu Akad. Paris 162 (1916), pp. 629–632

[dSm] de SMIT, Bart: "The Fundamental Group of the Hawaiian Earring is not free", Internat. J. Algebra Comput., Vol.2, No.1 (1992), pp. 33–37

[Wald] WALDHAUSEN, Friedhelm: "On irreducible 3-manifolds which are sufficiently large", Ann. Math. II, 87 (1968) pp. 56–88

[Why] WHYBURN, G.T.: "Topological Characterization of the Sierpinski Curve", Fund. Math. 45 (1958), pp. 320–324

[Will] WILLARD, Stephen: "General Topology", Addison-Wesley Series in Mathematics, Addison-Wesley, Reading (Massachusetts), 1970

[Zas1] ZASTROW, Andreas: "The second van-Kampen theorem for topological spaces", Topology and its Applications 59 (1994), 201-232

[Zas2] — , "Construction of an infinitely generated group that is not a free group of surface groups and free abelian groups, but which acts freely on an $\mathbb{R}$-tree", Proc. R. Soc. Edinb. 128A (1998), pp. 433–445

[Zas3] — , "Generalized π_1-determined covering spaces", in preparation

[Zas4] — , "The Non-Abelian Specker-Group is free", to appear J. Alg. 2000

[ZBG] ZIESCHANG, H., BITTER, W. and GRAMBERG, E.: "Dreidimensionale Topologie", Lecture Notes, Ruhr-University Bochum, 1973/74

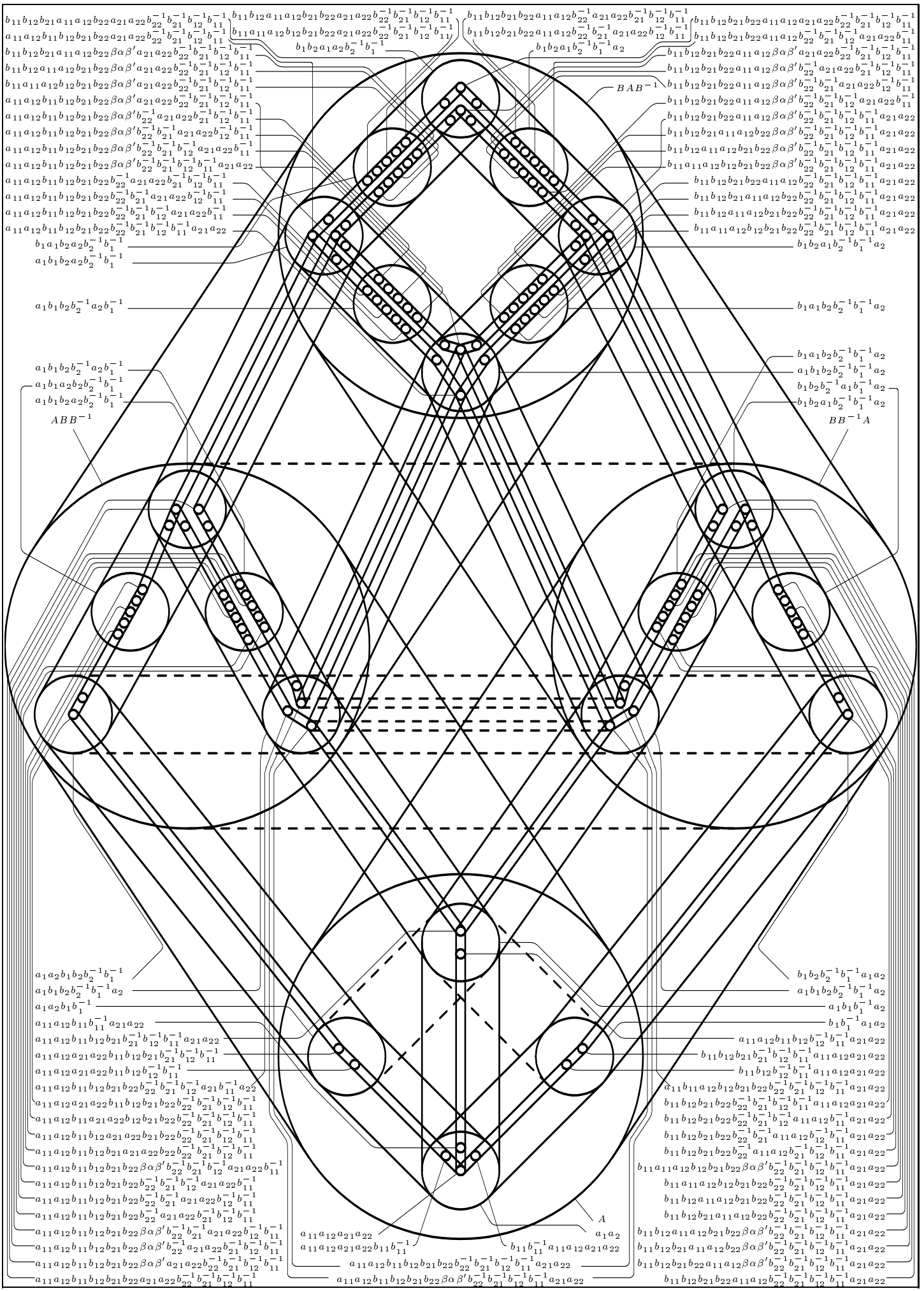
abstract path A.5.1
appendances ..... B.2.2(iii), B.2.2(v), D.1.1(iii)
to acknowledge a tangency ..... C.1.3(iii)
basic direction ..... B.1.2(i)
boundary lines (of corridors) ..... B.2.1(iii)
cancellable D.3.1
cancellations ..... B.2.2(i), B.2.2(iv), D.1.1(i)
cancellation graphs D.7.1
combing ..... A.5.5(i)
commuting letters B.1.8
compatibility of transformation processes ..... B.2.1(ii)/B.2.8/D.1.2
compatibility of word sequences C.1.6/C.1.7
corner-point intersection ..... B.1.2(v)
corridor ..... A.4.6(i)/B.1.1
corridor entrance and exit points D.10.1
corridor lattice B.1.1
corridor network B.1.1/B.1.2
corridor segment ..... B.1.2(iv)
to cross a corridor ..... A.4.6(i)/B.1.4(v)
to crop up A.3.4
crossing region of corridors ..... B.1.2(v)
diamonds D.6.6
dual graphs of corridor lattices ..... C.3.7(i)
dual split-up picture C.5.5
to disregard a tangency ..... C.1.3(i)
end segments (of corridors) ..... B.1.2(iii)
enlarging and reducing equivalence operations B.2.3
exceptional letters (in the final continuity proof) E.4.14
exhaustable ..... D.9.1($\Leftarrow$)
extendable transformation processes D.1.2
extended lattice square ..... B.1.2(iv)
external compatibility C.1.7
far-distance operations ..... B.2.2(iv)–(v), B.2.4
firm equivalence relation ..... B.2.9(iii)
fixed equivalence classes D.3.6
free equivalence classes D.3.6
full-split case ..... C.1.8(i)
gap-lines ..... B.1.2(ii)
ghost-letter ..... C.1.7(ii)
graph of position sequences C.5.5
graphs of reading adjustments D.2.6
graphs of reading possibilities D.2.12
hole ..... A.2.4, B.1.2(iv)

holes that crop up later A.3.4
homotopic distance ..... A.5.2(2)
homotopy combing A.5.5
horizontal corridor ..... B.1.2(iv)
to ignore a tangency ..... C.1.3(i)
index (of a hole of a space) A.3.1
indexing of letters C.1.1
insertions ..... B.2.2(i), B.2.2(iv), D.1.1(i)
internal compatibility C.1.6
to integrate a hole into a corridor lattice ..... A.3.9(fin.rem.)
inverse of a position sequence C.5.24
inverse of a word sequence C.4.9
iterated left or right split-up letter ..... C.5.3(iv)
Königslemma type of conclusions C.5.10
lattice square ..... B.1.2(iv)
left split-up letter ..... C.1.8(i)
letter B.1.7
level index A.3.1
lines (of corridors) ..... B.1.2(ii)
lines (within graphs) C.5.10
maxima and minima of position sequences C.5.16
minimal reading C.2.7
Modified Sierpinski Carpet A.3.3/A.3.5
nested graphs of word sequences D.7.5/D.7.6
Normal Form A.3.3
order of (classes of) position sequences ..... C.5.15/C.5.18(iii)
passage (of a path within a corridor network) ..... E.2.7(i)
phases (of paths in the final continuity proof) E.3.3/E.3.10
phases (of transformation patterns) D.11.10
position sequences D.5.1/D.5.3
position variables D.11.12
pure lattice square ..... B.1.2(iv)
prenormal A.3.7
reading domain ..... B.1.4(iv)-(v)
reading from a higher indexed word C.1.7
reading of a word sequence from a given path ..... B.1.4/B.1.6(1)/Sect.C.2
realizing word sequences by paths ..... B.1.6(2)/Sect.C.3
realization diagrams D.8.1
reduction of word sequences B.2.3/B.2.11
return (of a path within a corridor network) ..... E.2.7(ii)
to regard a tangency ..... C.1.3(i)
right split-up letter ..... C.1.8(i)
root corridor, -line and -letter ..... B.1.2(vi)/B.1.2(ii)/B.1.3

semi-free equivalence classes D.3.6
semi-split case ..... C.1.8(i)
sequence of nested vertices ..... D.7.7(i)
sides (of a corridor) ..... B.1.2(iii)
Sierpinski Carpet ..... Fig.a, A.3.5
single-split case ..... C.1.8(i)
slice-system D.9.2
split-up corridor, -line and -letter ..... B.1.2(vi)/B.1.2(ii)/B.1.3
split-up diagrams C.3.12
split-up graph C.5.5
stabilizing sequence of edge-paths ..... D.7.7(ii)
strong equivalence relation ..... B.2.0(1)/B.2.9(ii)
synchronizing of transformation processes ..... B.2.1(ii)/B.2.8/D.1.2
synchronous parametrization B.3.3, E.1.1, Sect.E.2
tangent point case ..... B.1.4(ii)
transformation diagrams D.1.3
transformation graphs D.4.9
transformation pattern D.1.17/D.1.20
transpositions ..... B.2.2(ii), D.1.1(ii)
triple sequence C.1.5
truncatable D.3.1
truncations ..... B.2.2(iii), B.2.2(v), D.1.1(iii)
upward extension of transformation processes D.1.9, D.3.14, D.6.2
value (of a letter) B.1.7
vertical corridor ..... B.1.2(iv)
weak equivalence relation ..... B.2.0(2)/B.2.9(i)
word sequence A.4.6, B.1.4, C.1.4
zero-split case ..... C.1.8(i)

Andreas Zastrow
Fakultät und Institut für Mathematik[†]
Ruhr-Universität Bochum
D–44780 Bochum
GERMANY

[†] PRESENT ADDRESS: Instytut Matematyczny, Uniwersytet Gdański, ul. Wita Stwosza 57, PL-80-952 Gdańsk, POLAND


$b_{11}b_{12}b_{21}a_{11}a_{12}b_{22}a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}\beta\alpha\beta' a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}a_{11}a_{12}b_{21}b_{22}\beta\alpha\beta' a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}a_{11}a_{12}b_{12}b_{21}b_{22}\beta\alpha\beta' a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}a_{21}a_{22}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}a_{21}a_{22}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}a_{21}a_{22}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}a_{21}a_{22}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}a_{21}a_{22}b_{12}^{-1}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}a_{21}a_{22}b_{11}^{-1}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_1a_1b_2a_2b_2^{-1}b_1^{-1}$
$a_1b_1b_2a_2b_2^{-1}b_1^{-1}$
$b_1b_2a_1b_2^{-1}b_1^{-1}$
$b_1b_2a_1b_2^{-1}b_1^{-1}a_2$
$a_1b_1b_2b_2^{-1}a_2b_1^{-1}$
$b_1a_1b_2b_2^{-1}b_1^{-1}a_2$
$a_1b_1b_2b_2^{-1}a_2b_1^{-1}$
$a_1b_1a_2b_2b_2^{-1}b_1^{-1}$
$a_1b_1b_2a_2b_2^{-1}b_1^{-1}$
ABB^{-1}
BAB^{-1}
$BB^{-1}A$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}b_{22}^{-1}a_{21}a_{22}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}a_{21}a_{22}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}\beta\alpha\beta' a_{21}a_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}\beta\alpha\beta' b_{22}^{-1}a_{21}a_{22}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}a_{21}a_{22}b_{12}^{-1}b_{11}^{-1}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}a_{11}a_{12}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}a_{11}a_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}a_{11}a_{12}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}a_{11}a_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}a_{11}a_{12}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_1b_2a_1b_2^{-1}b_1^{-1}a_2$
$b_1b_2b_2^{-1}b_1^{-1}a_1a_2$
$a_1b_1b_2b_2^{-1}b_1^{-1}a_2$
$a_1b_1b_1^{-1}a_2$
$b_1b_1^{-1}a_1a_2$
$a_{11}a_{12}b_{11}b_{12}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{11}a_{12}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}b_{12}^{-1}b_{11}^{-1}a_{11}a_{12}a_{21}a_{22}$
$a_{11}b_{11}a_{12}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{11}a_{12}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}a_{11}a_{12}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}a_{11}a_{12}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}a_{11}a_{12}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}a_{11}a_{12}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}a_{11}a_{12}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}a_{11}a_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}a_{11}a_{12}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}a_{11}a_{12}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$b_{11}b_{12}b_{21}b_{22}a_{11}a_{12}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
A
a_1a_2
$a_{11}a_{12}a_{21}a_{22}$
$a_{11}a_{12}a_{21}a_{22}b_{11}b_{11}^{-1}$
$b_{11}b_{11}^{-1}a_{11}a_{12}a_{21}a_{22}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$
$a_{11}a_{12}b_{11}b_{12}b_{21}b_{22}\beta\alpha\beta' b_{22}^{-1}b_{21}^{-1}b_{12}^{-1}b_{11}^{-1}a_{21}a_{22}$