

A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/150270>

Copyright and reuse:

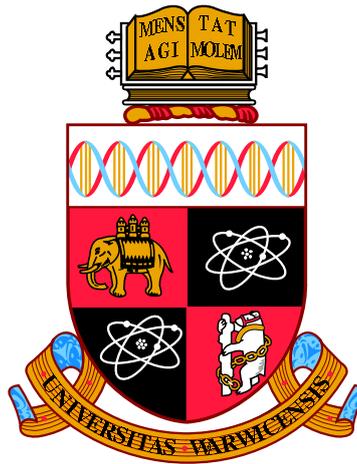
This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk



Polynomial-time efficient position

by

Ronja Johanna Barbara Kuhne

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Mathematics

August 2019



Contents

List of Figures	iv
List of Algorithms	vii
Acknowledgments	ix
Declarations	xi
Abstract	xii
Notation	xiii
Chapter 1 Introduction	1
1.1 Main result	3
1.2 Comparison with previous work	4
1.3 Structure of the thesis	6
1.4 Future research directions	7
Chapter 2 Background	9
2.1 Surfaces, curves, and arcs	9
2.2 Train tracks	9
2.3 Snippets and winding numbers	12
2.3.1 Snippets	13
2.3.2 Winding number	15
2.4 Efficient position and classification of bad snippets	16
2.5 Classification of bad snippets	19
2.5.1 Classification of bad snippets in the tie neighbourhood	19
2.5.2 Bad snippets in complementary regions	24
2.5.3 Classification of bad snippets - an overview	26
2.6 Efficient position for arcs and curves	26

2.7	Algorithms and Python notation for snippet-decomposed arcs and curves	30
2.7.1	Conventions on algorithms	30
2.7.2	Python notation	31
Chapter 3	Local homotopies and trigon arcs	33
3.1	Right/left turning snippets and horizontal/vertical duals	34
3.2	Corner length and reduced corner length	35
3.3	Local homotopies	39
3.4	Trigon homotopies	40
3.4.1	Trigons of type $\mathbb{B}(h, t)$	45
3.4.2	Trigons of type $\mathbb{S}(h, t, 1)$	47
3.4.3	Trigons of type $\mathbb{S}(h, v, 2)$	49
3.4.4	Trigons of type $\mathbb{S}(h, t, 3)$	51
3.4.5	Trigons of type $\mathbb{R}(h, v)$	52
3.4.6	Trigon homotopies - a summary	53
3.5	The algorithm <code>TrigArc</code>	56
3.6	The algorithm <code>TrigCurve</code>	60
Chapter 4	Local bigon homotopies	62
4.1	Local homotopies for bigon arcs	63
4.2	The algorithm <code>BigArc</code>	76
Chapter 5	Obtaining efficient position for all snippets but one	78
5.1	Reducing the number of bad snippets to two	79
5.2	Reducing the number of bad snippets from two to one	82
5.2.1	The idea behind <code>ReduceToOneBadSnippet</code>	82
5.2.2	Subsets of snippets and gluings of arcs	82
5.2.3	The algorithm <code>ReduceToOneBadSnippet</code>	83
Chapter 6	Efficient position for curves with a single bad snippet	85
6.1	<code>WeightOneBigon</code> -algorithm for almost efficient curves	86
6.2	<code>WeightTwoBigon</code> -algorithm for almost efficient curves	88
6.3	<code>AllButHorBigInCompOrBranch</code> -algorithm for almost efficient curves	90
6.4	<code>TwoTrigons</code> -algorithm for almost efficient curves	91
6.5	<code>HorBigonInCompWide</code> -algorithm for almost efficient curves	96
6.6	<code>HorBigonInComp</code> -algorithm for almost efficient curves	99
6.7	<code>HorBigonInBranch</code> -algorithm for almost efficient curves	101

6.8 The algorithm <code>SingleBadSnippet</code>	106
Chapter 7 Polynomial-time efficient position	109

List of Figures

1.1	An arc whose snippet length is growing under a semi-local homotopy.	5
2.1	Examples of a large, small, and mixed branch.	10
2.2	A tie neighbourhood rectangle.	11
2.3	Identifications of the vertical boundaries of branch and switch rectangles.	11
2.4	These subsurfaces have indices 0, $-2/4$, and $-10/4$ respectively.	12
2.5	A branch rectangle	13
2.6	A switch rectangle	13
2.7	A horizontal boundary component of $R \in \mathcal{R}_{\text{comp}}$ containing more than two points of $\partial^2\mathcal{R}$.	14
2.8	Snippets in a switch rectangle and complementary region.	14
2.9	Winding number 5.	16
2.10	Winding number -20 .	16
2.11	Winding number 0.	16
2.12	Winding number 10.	16
2.13	Examples of carried snippets in a switch rectangle.	18
2.14	A dual snippet inside a branch rectangle.	18
2.15	Examples of snippets in efficient position that lie inside a simply connected complementary region.	18
2.16	Examples of snippets in efficient position that lie inside a peripheral complementary region.	18
2.17	All possible types of bad snippets inside a branch rectangle.	20
2.18	Snippets inside a switch rectangle of weight zero, one, two, and three respectively.	21
2.19	The six different strong snippet homotopy classes of bigon snippets inside a switch rectangle, up to symmetry.	23
2.20	The four different strong snippet homotopy classes of trigon snippets inside a switch rectangle, up to symmetry.	23

2.21	The five types of non-peripheral bad snippets inside a complementary region.	24
2.22	Weakly but not strongly snippet homotopic snippets.	25
2.23	Adjacent snippets of different types in an arc or curve in efficient position.	27
3.1	A right-turning trigon inside a switch rectangle.	34
3.2	A left-turning trigon inside a complementary region.	34
3.3	A right horizontal dual.	35
3.4	A left vertical dual.	35
3.5	These snippets have corner length zero, one, four, and $2s_N$ respectively.	36
3.6	A left blocker.	38
3.7	A trigon inside a switch rectangle whose boundary contains more than one point of $\partial^2\mathcal{R}$	41
3.8	A trigon inside a complementary region whose boundary contains more than one point of $\partial^2\mathcal{R}$	41
3.9	A trigon that contains a unique point of $\partial^2\mathcal{R}$	42
3.10	A trigon that contains more than one point of $\partial^2\mathcal{R}$	42
3.11	A homotopy of type $\mathbb{B}(h, t)$ applied to a trigon snippet that intersects the large end of a switch rectangle.	46
3.12	A homotopy of type $\mathbb{B}(h, t)$ applied to a trigon snippet that intersects a small end of a switch rectangle such that the trigon does not contain any corners of the adjacent complementary region in its boundary.	46
3.13	A homotopy of type $\mathbb{B}(h, t)$ applied to a trigon snippet that intersects a small end of a switch rectangle such that the trigon contains a corner of the adjacent complementary region in its boundary.	46
3.14	A homotopy of type $\mathbb{S}(h, t, 1)$ applied to a trigon snippet that intersects a small end of a switch rectangle.	48
3.15	A homotopy of type $\mathbb{S}(h, t, 1)$ applied to a trigon snippet that intersects the large end of a switch rectangle.	48
3.16	A homotopy of type $\mathbb{S}(h, v, 2)$	50
3.17	A homotopy of type $\mathbb{S}(h, v, 2)$ creating a trigon of type $\mathbb{R}(h, v)$	50
3.18	A homotopy of type $\mathbb{S}(h, t, 3)$	51
3.19	A homotopy of type $\mathbb{R}(h, v)$	53
3.20	The trigon homotopy graph for right-turning trigons.	55
4.1	Homotopies of type $\mathbb{S}(t, t, 0)$	64
4.2	Homotopies of type $\mathbb{B}(t, t)$	64

4.3	A homotopy of type $\mathbb{R}(v, v)$	64
4.4	A homotopy of type $\mathbb{S}(v, v, 0)$	64
4.5	A homotopy of type $\mathbb{S}(h, h, 0)$	64
4.6	A homotopy of type $\mathbb{S}(t, v, 1)$	68
4.7	A homotopy of type $\mathbb{S}(t, t, 2)$	68
4.8	A homotopy of type $\mathbb{S}(t, t, 2)$ applied to a curve of snippet length two.	69
4.9	Homotopies of type $\mathbb{R}(h, h)$ for a curve of snippet length two.	72
4.10	Two different kinds of trigon snippets of type $\mathbb{R}(h, h)$	72
4.11	A homotopy of type $\mathbb{B}(h, h)$	74
5.1	Gluing of an arc whose first and last snippet have identical middle thirds. For clarity, small push-offs have been performed in the figure.	83
6.1	A homotopy of type $\mathbb{S}(t, v, 1)$ followed by a homotopy of type $\mathbb{R}(h, v)$	88
6.2	Applying the algorithm <code>TwoTrigons</code> to a curve of snippet length three.	94
6.3	Applying two trigon homotopies on an input of <code>TwoTrigons</code> which has snippet length at least four.	94
6.4	A curve with two adjacent trigons of which one lies in a complementary region and one inside a switch rectangle.	94
6.5	A trigon snippet of type $\mathbb{B}(h, h)$ that is adjacent to two vertical duals that turn the same way.	103
6.6	Examples of trigon snippets of type $\mathbb{B}(h, h)$ that are adjacent to two vertical duals which turn into different directions and both have corner length greater than one.	103
6.7	A trigon snippet of type $\mathbb{B}(h, h)$ that is adjacent to two vertical duals which turn into different directions and which both have corner length one.	103
6.8	A trigon snippet of type $\mathbb{B}(h, h)$ that is adjacent to two vertical duals which turn into different directions and of which exactly one has corner length one.	103

List of Algorithms

3.5.1 TrigArc - Homotoping the inside of arcs with at most one bad snippet into efficient position, where the bad snippet is of trigon type.	56
3.6.1 TrigCurve - Homotoping a curve with at most one bad snippet into efficient position or shortening it to consist of a single snippet. The bad snippet of the input curve is required to be a trigon snippet.	60
4.2.1 BigArc - Achieving efficient position for the inside of an arc that contains at most one bad snippet, which must be its penultimate snippet.	76
5.1.1 ReduceToTwoBadSnippets - Achieving efficient position for all but the first and last snippet of a given arc or curve.	79
5.2.1 ReduceToOneBadSnippet - Reducing the number of bad snippets of a given curve to at most one.	83
6.1.1 WeightOneBigon - Homotoping an almost efficient curve with a bad snippet of type $\mathbb{S}(t, v, 1)$ into an input for TrigCurve or into an almost efficient curve of shorter reduced corner length.	87
6.2.1 WeightTwoBigon - Homotoping an almost efficient curve with a bigon of type $\mathbb{S}(t, t, 2)$ into an input for TrigCurve	89
6.3.1 AllButHorBigInCompOrBranch - Homotoping an almost efficient curve of bigon type that does not contain a bigon snippet of type $\mathbb{R}(h, h)$ or $\mathbb{B}(h, h)$ into an input for TrigCurve or into an almost efficient curve of shorter reduced corner length.	91
6.4.1 TwoTrigons - Homotoping a curve with exactly two bad snippets which must be adjacent trigon snippets in the tie neighbourhood that turn into the same direction into an input for TrigCurve or into a curve of shorter reduced corner length containing a single bigon.	92
6.5.1 HorBigonInCompWide - Homotoping an almost efficient curve with a wide bigon of type $\mathbb{R}(h, h)$ into an input for TrigCurve or into an almost efficient curve of bigon type and shorter reduced corner length.	97

6.6.1	HorBigonInComp - Homotoping an almost efficient curve with a snippet of type $\mathbb{R}(h, h)$ into an input for TrigCurve or into an almost efficient curve of bigon type and shorter reduced corner length.	100
6.7.1	HorBigonInBranch - Homotoping an almost efficient curve with a bigon snippet of type $\mathbb{B}(h, h)$ into an input for TrigCurve or into a curve of shorter reduced corner length containing a single bigon snippet. . . .	101
6.8.1	SingleBadSnippet - Homotoping a curve with at most one bad snippet into efficient position or shortening it to consist of a single snippet. . .	107
7.0.1	EfficientPosition - Achieving efficient position for essential and non-peripheral arcs and curves or shortening inessential or peripheral arcs and curves so that they consist of a single snippet.	110

Acknowledgments

Firstly, I would like to thank my supervisors Saul Schleimer and Brian Bowditch. Saul, for his kind and patient guidance throughout my PhD, for being such an open, curious and engaging person, for always knowing when it was time to make a silly joke, and for using a GriGri when his eagerness to help was once again too great. Brian, for making sure that everything was going smoothly and for offering his help on innumerable occasions.

Secondly, I would like to thank the geometry and topology community in and beyond Warwick. The last four years have been full of wonderful people, stimulating conversations, and exciting adventures. My special thanks goes to Beatrice Pozzetti for her continuous guidance and support, for all the nice evenings spent together and for never allowing me to take myself too seriously. I also like to thank Yoav Moriah, Thomas Vogel, and Stefan Friedl for their invitations and great hospitality.

Thirdly, I am highly indebted to the University of Warwick and the mathematics department for providing such a welcoming and stimulating environment. The PhD community as well as the administrative staff have always offered a smile and comforting word and have helped whenever possible.

I am grateful to the Engineering and Physical Sciences Research Council for funding my studies and for giving me the chance to live in the UK. It has been a pleasure and invaluable experience.

I am very lucky to have spent my time at Warwick in the company of such amazing people. My special thanks goes to my housemate, Simon, and all my climbing partners for the many hours that were filled with laughter and great conversations. No words can describe how grateful I am for spending the last years with Esmee and

Alex and knowing that they would always have my back.

Last, but certainly not least, I would like to express my gratitude for the unlimited support I have received from Bernhard and my parents. I can not thank you enough for all your love, trust, and encouragement.

Declarations

I declare that the material in this thesis is, to the best of my knowledge, my own work except where otherwise indicated in the text, or where the material is widely known. No part has been submitted by me for any other degree.

This thesis was typeset in LaTeX2e using the style package *warwickthesis*. All figures were created by the author using TikZ.

Abstract

Suppose that S is a surface of positive complexity and $N \subset S$ a tie neighbourhood of a large train track τ in S . We present a polynomial-time algorithm that, given a properly immersed, essential, and non-peripheral arc or curve $\alpha \subset S$, homotopes α into efficient position with respect to the tie neighbourhood N .

Proofs for the existence of efficient position were previously given in [Tak00] and [MMS12]. In [Tak00], a constructive proof for the existence of efficient position is given for immersed curves on closed surfaces of genus greater than or equal to two. There is no discussion of the complexity of the implied algorithm. In [MMS12], the existence of efficient position is proved for embedded curves with respect to birecurrent train tracks on surfaces of positive complexity. The implied algorithm operates via an exhaustive search. No time bounds can be deduced.

We note that the algorithm presented in this thesis and the algorithm suggested by a careful reading of [Tak00] coincide in the case of closed surfaces. However, this thesis constitutes more than a time-complexity analysis of Takarajima's constructive proof. Firstly, we are more general as we allow surfaces with boundary, whereas Takarajima only considers closed surfaces. Secondly, our combinatorial set-up uses arcs and curves with transverse self-intersection, whereas the barycentric subdivision of complementary regions carried out in [Tak00] forces non-transverse self-intersection even for curves which are initially embedded. Thirdly, the algorithm in this thesis is formulated purely in terms of local homotopies, whereas [Tak00] requires semi-local arguments. Thus, we can, and do, give pseudocode for our algorithm as well as prove its correctness.

Notation

Standard notation	Description
S	A compact, connected, oriented surface
∂S	The boundary of S
\tilde{S}	The universal cover of S
$\chi(S)$	The Euler characteristic of S
$\xi(S)$	The complexity of S
α	An immersed arc or curve
S^1	The circle, identified with $[0, 1]/\sim$
\square	An indicator that the proof is complete, is omitted, or is given elsewhere
Further notation	Description (with page references)
τ	A train track (9)
$N(\tau)$	A tie neighbourhood of a train track τ (11)
T	A subsurface with corners (12)
$\partial^2 T$	The set of corners of T (12)
$\text{index}(T)$	The index of T (12)
R	A branch or switch rectangle or a complementary region (11)
$\partial_h R$	The horizontal boundary of R (11)
$\partial_v R$	The vertical boundary of R (11)
\mathcal{R}_{tie}	The set of branch and switch rectangles of a tie neighbourhood (11)

$\mathcal{R}_{\text{comp}}$	The set of complementary regions of a tie neighbourhood (11)
\mathcal{R}	The set of all branch and switch rectangles and complementary regions of a tie neighbourhood (11)
$\partial\mathcal{R}$	The union of all boundaries of elements in \mathcal{R} (13)
$\partial^2\mathcal{R}$	The union of all corners of elements in \mathcal{R} (13)
$\text{wind}(a)$	The winding number of a snippet a inside a complementary region (15)
$\mathbb{B}(\bullet, \bullet)$	The type of a bad snippet inside a branch rectangle (20)
$\mathbb{S}(\bullet, \bullet, \bullet)$	The type of a bad snippet inside a switch rectangle (23)
$\mathbb{R}(\bullet, \bullet)$	The type of a bad snippet inside a complementary region (24)
$\text{len}(\alpha)$	The snippet length of α (27)
$\alpha[i + \epsilon : i + \delta]$	The subarc of the i -th snippet of α which is the image of the interval $[\epsilon, \delta]$ under the map $\alpha[i]$ (31)
α_{trim}	The inside of α (31)
s_N	The maximal side length of N (35)
$\text{len}_{\text{corn}}(\alpha)$	The corner length of α (36)
$\text{len}_{\text{block}}(\alpha)$	The number of blockers of α (38)
$\text{len}_{\text{red}}(\alpha)$	The reduced corner length of α (38)
$\text{carr}(\alpha)$	The number of carried snippets of α (45)
$\text{dual}_R(\alpha)$	The number of right duals of α (45)
$\text{dual}_L(\alpha)$	The number of left duals of α (45)

Chapter 1

Introduction

The compactification of Teichmüller space by the space of projective measured laminations, \mathcal{PML} , lies at the heart of Thurston’s pioneering approach to geometric topology. Studying curves and their limits under iterations of a mapping class is essential to much of Thurston’s work, including the Nielsen-Thurston classification of surface homeomorphisms, the topological characterization of rational maps, and the geometric classification of mapping tori [Hub06, Preface].

By introducing train tracks as a combinatorial tool to coordinatise \mathcal{PML} , Thurston opened up new ways of answering questions in topology, one-dimensional complex dynamics, and geometry. His “tangential train track coordinate system” provides a canonical piecewise integral projective structure for \mathcal{PML} and is perfectly adapted to studying curves that are carried by the stable train track of a pseudo-Anosov map ([Thu97, Chapter 8.9],[FM12, Chapter 15]). However, this coordinate system of \mathcal{PML} is not global. Therefore, comparing curves that lie in different charts, that is, curves that are not carried by the same maximal train track, is difficult.

A first step towards defining a global coordinate system for \mathcal{PML} involving train tracks was made by Penner, a student of Thurston. He introduced the terminology of two train tracks “hitting efficiently” if they intersect transversely and do not bound any bigon between them [PH92, Chapter 1.3]. This leads to a dual concept of tangential train track coordinates called “transverse train track coordinates” [Thu98, Section 9]. Instead of counting carried arcs of the curve, we now count transverse intersections of the curve with the train track. However, this coordinate system for curves is not only coarser than the “tangential train track coordinate system”, but suffers, by duality, from the same problem as tangential train track coordinates: it is not global.

Yet, there are global coordinate systems for curves on surfaces, with the two

most prominent of the last century being normal coordinates and Dehn-Thurston coordinates.

Normal coordinates count intersections of curves with the edges of an ideal triangulation of a punctured surface. For a formal definition and an extensive list of their applications, we refer the reader to [EN13]. To determine normal coordinates, curves are assumed to be in “normal form” with respect to the ideal triangulation. That is, they intersect the ideal triangles in standardized arcs, none of which cuts off a bigon. The existence of this normal form for every homotopy class of curves follows from a step-by-step elimination of bigons, each reducing by two the number of intersections between the curve and the edges of the triangulation. Normal coordinates are very well-suited to compute images of iterations of a mapping class on curves: mapping classes can be represented as paths in the flip graph of the surface [BW16, Section 1.2], and computing the change of coordinates under a single flip is straightforward [Bel15, Theorem 2.2.1]. However, this global coordinate system has one significant drawback: the introduction of an artificial marked point on closed surfaces leads to a non-finite ambiguity. This requires special care when working with closed surfaces and aiming to apply the same algorithms as in the non-closed case [BW16, Section 1.1].

Introduced by Dehn in 1922 [Deh87, Paper 7, Section 2] and later revived by Thurston [Thu88], *Dehn-Thurston coordinates* use a pants decomposition of the surface and a set of twisting parameters to provide a global coordinate system to the set of curves and laminations [PH92, Theorem 3.1.1]. Here, closed and non-closed surfaces can be treated alike. As in the case of normal coordinates, the “normal position” of a curve with respect to the pants decomposition can be achieved via a straightforward argument. Furthermore, explicit piecewise integral linear formulas for the change of coordinates under a single pants move are given by Penner in [Pen06, Section 3]. Yet, these are quite complex and applications of Dehn-Thurston coordinates for computational purposes have been rather sparse [Pen06, Section 3].

In 1998, Takarajima combined the two versions of train track coordinates to prepare the ground for a new global coordinate system for curves: “efficient coordinates”. He introduced a third normal position for curves with respect to a train track, called “quasi-transversality” [Tak00, Definition 3.2]. This constitutes a simultaneous generalization of “carried” curves and “dual” curves. An almost identical concept was defined independently by Masur, Mosher, and Schleimer in 2010 under the name of “efficient position” [MMS12, Definition 2.3]. The existence of efficient position is much more subtle than that of normal position required for normal coordinates: a subdivision of the tie neighbourhood of the train track induces

a tiling of the surface by polygons. If a curve is not in efficient position, a subarc of the curve cuts off a region of positive index inside a tile. Pushing the subarc across this region can increase the number of intersections of the curves with the boundary of the tiles. Furthermore, it might not reduce the overall amount of positive index cut off by subarcs in the tiles (see page 42, Figure 3.10).

In 2000, Takarajima proved the existence of efficient position for immersed curves on closed surfaces using explicit semi-local homotopies that transform a given curve into efficient position [Tak00, Proposition 4.1]. There is no discussion of the complexity of the implied algorithm in [Tak00]. In 2010, Masur, Mosher, and Schleimer gave an existence proof for embedded curves on surfaces of positive complexity with respect to birecurrent train tracks [MMS12, Theorem 4.1]. We note that their proof is constructive but relies on an exhaustive search. Thus, no time bounds can be deduced from the work in [MMS12]. The results in this thesis imply that their algorithm halts in exponential time. We further remark that the existence of efficient position with respect to Reebless bigon tracks on the torus is proved in [Gué09, Lemma 14]. The proof is constructive but no time bounds are discussed.

1.1 Main result

We give an algorithmic proof of the existence of efficient position for immersed curves on surfaces of positive complexity. The presented algorithm halts in quadratic-time in the length of the input curve. More precisely, suppose that S is a surface of positive complexity. Fix a large train track $\tau \subset S$. A subdivision of a tie neighbourhood of τ into “branch rectangles” and “switch rectangles” provides a tiling of S by polygons and peripheral annuli. We assume that curves are given as cutting sequences with respect to the one-skeleton of this tiling. Such curves are said to be *snippet-decomposed*, and we refer to subarcs of the curve that lie inside a single tile as *snippets*.

Theorem 7.0.2. *There is an algorithm that takes as input*

- *a surface $S = S_{g,b}$ satisfying $\xi(S) = 3g - 3 + b \geq 1$,*
- *a tie neighbourhood $N = N(\tau)$ of a large train track τ in S , and*
- *a properly immersed arc or curve α given via its snippet decomposition with respect to N ,*

and outputs an arc or curve α' homotopic to α , relative to endpoints in the case of arcs, such that

- α' is in efficient position with respect to N or
- α' has snippet length one.

This algorithm terminates in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time. Moreover, the output α' has snippet length one if and only if α is inessential or peripheral. If α is peripheral, the boundary component that α is homotopic to, as well as the corresponding power, can be read off from the snippet-decomposition of α' .

As we work purely combinatorially, the notion of efficient position is defined with respect to the tie neighbourhood of a train track (see page 17, Definition 2.4.2). We note that an arc or curve is in efficient position with respect to a tie neighbourhood of a large train track τ as defined here if and only if it is in efficient position with respect to τ following the definition given in [MMS12, Definition 2.3]. We remark that there already exist polynomial-time algorithms that decide if arcs or curves are essential and non-peripheral: the word problem for fundamental groups can be decided in linear time [Deh87], and normal coordinates allow the detection of peripheral curves.

The mathematical content of Theorem 7.0.2 can be rephrased as follows.

Corollary 7.0.3. *Suppose that $S = S_{g,b}$ is a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. Suppose that $\tau \subset S$ is a large train track and $N = N(\tau)$ is a tie neighbourhood of τ in S . Let $\alpha \subset S$ be an essential, non-peripheral, properly immersed multiarc or multicurve given via its snippet decomposition. Then efficient position for α with respect to N exists and can be obtained in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time.*

1.2 Comparison with previous work

Even though the language and combinatorial set-up differs greatly from the one in [Tak00], the algorithm presented in this thesis and the algorithm suggested by a careful reading of [Tak00] coincide in the case of closed surfaces. However, this thesis constitutes more than a time-complexity analysis of Takarajima's constructive proof. Firstly, we are more general as we allow surfaces with boundary, whereas Takarajima only considers closed surfaces. Secondly, our combinatorial set-up uses arcs and curves with transverse self-intersection, whereas the barycentric subdivision of complementary regions carried out in [Tak00] forces non-transverse self-intersection even for curves which are initially embedded. Thus, one cannot easily recognize embedded arcs and curves or compute intersection numbers in the set-up of [Tak00]. This is contrary to the aim of using efficient position to provide a coordinate system

for embedded curves on surfaces. Thirdly, the algorithm in this thesis is formulated purely in terms of local homotopies, whereas [Tak00] requires semi-local arguments. Thus, we can, and do, give pseudocode for our algorithm as well as prove its correctness.

Studying Takarajima’s proof for the existence of efficient position, one is tempted to look for a lexicographic complexity involving the length or curvature of a curve which decreases when applying local or semi-local homotopies to the arc or curve. All attempts by the author to find a simple linear or lexicographic measure decreasing for all types of local homotopies were unsuccessful. When considering arcs containing a single bad snippet of trigon type, the search for a lexicographic order is partially successful: the sum of *right* or *left duals* and carried snippets decreases under all but one type of local trigon homotopies. The number of times this “exceptional” type of homotopy is applied when eliminating one trigon snippet from a subarc can be bounded. However, the number of snippets of the underlying arc might be multiplied by a constant in these cases. For an illustration, we refer the reader to Figure 1.1. Thus, if we apply such homotopies repeatedly to achieve efficient position for longer and longer subarcs of the given arc or curve, we could see an exponential growth of the curve. We also note that this discussion would not be sufficient for curves containing a single bad snippet.



Figure 1.1: An arc whose snippet length is growing under a semi-local homotopy.

The crucial observation is that this multiplicative growth is caused by snippets cutting off index zero regions in complementary regions of the tie neighbourhood. However, subarcs created during such expanding homotopies cut off regions of index zero on one side, thus are “blocked” by a significant amount of index on the other side. Hence, their “preferred” way of moving under subsequent homotopies is back into their original “shorter” state. An index-argument shows that if we want to turn even a single snippet of these subarcs into a “potentially growing” snippet again, we first have to eliminate one bad snippet of the underlying curve. As the number of bad snippets is finite, the growth of the arc or curve under these trigon homotopies can thus be bounded. This interplay of index and length is captured in two combinatorial notions of length, the *corner length* and the *reduced corner length*. A careful analysis shows that the reduced corner length indeed only increases under the trigon homotopies if the trigon is “eliminated”. As bigon homotopies naturally

reduce to trigon homotopies, this will be sufficient to yield a proof of Theorem 7.0.2.

1.3 Structure of the thesis

The layout of this thesis is as follows. In Chapter 2 we introduce the relevant notions needed for the proof of Theorem 7.0.2. In particular, this includes the decomposition of the curve into *snippets*, our first combinatorial notion of length, the *snippet length* of an arc or curve, as well as the classification of snippets which are not in efficient position. Chapter 3 provides two further combinatorial notions of length, the *corner length* and the *reduced corner length*. We define a family of local homotopies and study their effects on the type of bad snippets and combinatorial lengths of almost efficient arcs and curves of *trigon type*. We end this chapter by presenting the algorithms `TrigArc` and `TrigCurve`, which can be used to achieve efficient position for the “inside” of almost efficient arcs and curves of trigon type. Both algorithms run in polynomial time in the reduced corner length and increase the latter only by an additive constant.

In Chapter 4 we discuss how to reduce the case of almost efficient arcs of *bigon type* to the case of almost efficient arcs of trigon type. We then present the algorithm `BigArc`, which achieves efficient position for the inside of almost efficient arcs of bigon and trigon type. Again, this algorithm runs in polynomial time in the reduced corner length of the arc and increases the latter only by a constant. We note that we restrict ourselves to the case of almost efficient arcs which have their bad snippets at the penultimate position. This simplifies subsequent running time analyses.

In Chapter 5 we give an algorithm that yields efficient position or reduces the number of bad snippets to at most one if α is an essential arc or an essential and non-peripheral curve respectively. Since the reduced corner length of an almost efficient arc or curve α differs from the snippet length $\text{len}(\alpha)$ by at most a constant, the resulting algorithm halts in polynomial time in the length of α .

In Chapter 6, we discuss how to achieve efficient position for an essential and non-peripheral curve that contains a unique bad snippet. We present a series of algorithms, each designed to deal with a certain type of bigon snippet, whose application lowers the reduced corner length of the underlying curve by at least one. Each of these algorithms terminates in polynomial time in the reduced corner length of the curve.

In Chapter 7 we combine the algorithms presented in Chapter 5 and 6 to give a proof of Theorem 7.0.2 and Corollary 7.0.3.

1.4 Future research directions

Given the existence of efficient position for embedded curves, one can study “efficient coordinates” for curves on surfaces. We note that this yields, in particular, a global coordinate system for curves on closed surfaces. However, efficient position is only unique up to rectangle and annulus swaps ([Tak00, Proposition 5.4], [MMS12, Theorem 4.1]). This implies a similar, though finite, ambiguity as in the case of normal coordinates on closed surfaces.

First of all, one might ask how the graph of efficient positions of a homotopy class of curves looks like. We note that Takarajima suggests a preferred efficient position [Tak00, Theorem 6.7], which might be used to obtain a linear subspace or quotient space of efficient coordinates. This approach could provide a set of global, injective coordinates for curves on closed surfaces.

In the context of studying mapping classes, efficient coordinates might prove useful in the following way: mapping classes can be encoded by paths in the train track graph [Ham09, Section 3]. Two vertices in the train track graph are connected if one train track can be obtained from the other by a single split. Since efficient position can be achieved in polynomial time with the algorithm presented in this thesis, one can compute the change of efficient coordinates under a split in polynomial-time as well. This raises hopes that there is a formula for the coordinate change which yields a piecewise-linear map on \mathcal{PML} . However, this seems to require a delicate analysis of carried curves of weight at most two that are combed on one side. Going further one can study how coordinates change under splits and folds if the curves are encoded in terms of weighted train tracks which are in efficient position with respect to a fixed train track. The latter could give a new way of finding minimal positions of curves that are given in normal coordinates, which might give an alternative approach to calculating distances in the curve graph [BW16]. We remark that any algorithm computing intersection numbers and minimal positions of curves that builds directly on the work presented in this thesis will be dependent on the genus of the underlying surface and must require at least a quadratic number of operations in the lengths of the input curves. Using coordinates of curves that stem from surfaces decomposed into quadrilaterals, such an algorithm with quadratic running time was recently presented in [DL17].

With a detailed understanding of the changes of efficient coordinates under splits and folds, the proof of the polynomial-time recognition of mapping class types in [BW16] could be reformulated in terms of (compressed) efficient coordinates. The efficient coordinate approach might eliminate exponential dependencies on the Euler

characteristic in the running time and the need for introducing an artificial puncture in the closed case. Furthermore, an implementation of these algorithms in efficient coordinates could give a similar computer programme to “Flipper”, which computes the action of mapping classes on laminations on a punctured surface using normal coordinates [Bel18].

Chapter 2

Background

In this section, we provide the definitions needed for the proofs of Theorem 7.0.2 and Corollary 7.0.3.

2.1 Surfaces, curves, and arcs

Let $S = S_{g,b}$ be a compact, connected, oriented surface of genus g with b boundary components. The boundary components of S receive the induced orientation. The *complexity* of S is defined as $\xi(S) = 3g - 3 + b$. A *curve* in S is a proper immersion of the circle S^1 into S . We require curves to be in *general position*; that is, we only consider smooth curves with transverse self-intersections. We say that an immersed curve $\alpha \subset S$ is *essential* if it represents a non-trivial conjugacy class in the fundamental group of S . We further say that an immersed curve is *peripheral* if it is homotopic to a power of a boundary component of S .

An *arc* in S is an immersion of the interval $[0, 1]$ into S . As for curves, we require arcs to be in general position. In particular, they intersect the boundary of S transversely. We say that an arc in S is *proper* if it is a proper immersion of the interval $[0, 1]$ into S . We say that a proper arc is *essential* if it is not homotopic, relative its endpoints, into the boundary of S . An arc or curve $\alpha \subset S$ that does not have any self-intersections is said to be *embedded*.

2.2 Train tracks

This brief introduction to train tracks is based on [PH92, Chapter 1], [MMS12, Section 2.3],[Mos03, Chapter 3], and [MS18, Section 3.4].

Definition 2.2.1. A *pretrack* $\tau \subset S$ is a non-trivial, properly embedded, locally finite graph in S with the following properties: for each vertex v of τ there is a unique tangent line $L \subset T_v(S)$ such that for some neighbourhood U of v the intersection $\tau \cap U$ is a union of smooth open arcs in S , all of which are tangent to L at v (see Figure 2.1). The edges of τ are called *branches* and are smoothly embedded in S . The vertices of τ are called *switches*. We require every switch to have valence three. A smooth immersion $\rho : \mathbb{R} \rightarrow S$ is a *train-route* if

- $\rho(\mathbb{R}) \subset \tau$, and
- $\rho(n)$ is a switch if and only if $n \in \mathbb{Z}$.

Every branch b of a pretrack is required to have a train-route travelling along b .

If a train-route is periodic, it is called a *closed train-route*. If b is a branch of a pretrack τ and $p \in b$, a component of $b - p$ is called a *half-branch* of τ . If the intersection of two half-branches of a branch b is again a half-branch of b , the two half-branches are regarded as being equivalent. An equivalence class of half-branches of b is called an *end* of b . An end e of a branch b is said to be *incident* to a switch v of τ if v lies in the closure of a half-branch representing e . For each switch v of τ , fix a direction in the tangent line $L \subset T_v(S)$ at v . We orient branch ends towards the switches they are incident to. A branch end e incident to a switch v is called *incoming* if its orientation coincides with the chosen direction of the tangent line L at v , *outgoing* otherwise. Hence, the collection of branch ends incident to a switch can be split into two non-empty sets. If one of these sets consists of one branch end only, the corresponding branch end is called *large*, otherwise *small*. A branch with large ends on both sides is called *large*, with a large end on one side only *mixed*, and *small* if both of its branch ends are small (see Figure 2.1).

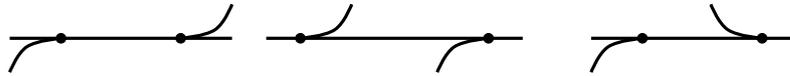


Figure 2.1: Examples of a large, small, and mixed branch.

From a pretrack $\tau \subset S$ we build a *tie neighbourhood* $N = N(\tau)$ as follows: Let $\mathcal{B} = \mathcal{B}(\tau)$ be the set of branches of τ and $\mathcal{S} = \mathcal{S}(\tau)$ its set of switches. We take one rectangle R_b for each branch $b \in \mathcal{B}$ and one rectangle R_s for each switch $s \in \mathcal{S}$. All rectangles are foliated by vertical arcs (the *ties*). The boundary of each such rectangle R consists of four edges: two edges that are parallel to the ties, and two that are perpendicular to the ties. We refer to the union of the first two edges as the *vertical boundary* of R and denote it by $\partial_v R$. The union of the perpendicular edges

is called the *horizontal boundary* of R and is denoted by $\partial_h R$. For an illustration, we point the reader to Figure 2.2. Suppose that e is a branch end of $b \in \mathcal{B}$ that is incident to a switch $s \in \mathcal{S}$. We then glue the edge of $\partial_v R_b$ corresponding to e to a subset of $\partial_v R_s$ as determined by the combinatorics of the train track (see Figure 2.3). The resulting surface $N = N(\tau)$ can be embedded into S in such a way that it is disjoint from ∂S and that the train track τ is properly embedded in it. We call this surface $N = N(\tau)$ the *tie neighbourhood* of the train track τ .

Remark 2.2.2. In the literature, there are three common models for neighbourhoods of train tracks and branched surfaces: the *cusp model* ([PH92, page 90]), the *corner model* (introduced above, see also [MS18, page 11]), and the *smooth model* ([Mos96, page 28]). The combinatorial setting of this thesis asks for generic intersections of arcs and curves with themselves and the boundary of the tie neighbourhood. Thus, the corner model is the natural choice.

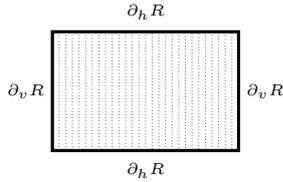


Figure 2.2: A tie neighbourhood rectangle.

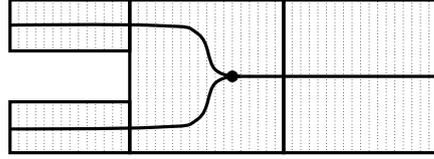


Figure 2.3: Identifications of the vertical boundaries of branch and switch rectangles.

The images of the rectangles R_b for $b \in \mathcal{B}$ are called *branch rectangles*, and, in an abuse of notation, are again denoted by R_b . In a similar abuse of notation we have *switch rectangles* R_s for $s \in \mathcal{S}$ and ties and vertical and horizontal boundaries of the embedded rectangles in S . The collection of branch and switch rectangles of a tie neighbourhood $N = N(\tau)$ is denoted by $\mathcal{R}_{\text{tie}} = \mathcal{R}_{\text{tie}}(N)$. The *horizontal boundary* $\partial_h N$ is the union of $\partial_h R$ for $R \in \mathcal{R}_{\text{tie}}$, while the *vertical boundary* is $\partial_v N = \overline{\partial N} - \partial_h N$.

Let $N = N(\tau)$ be a tie neighbourhood of a pretrack τ . Let R be a *complementary region* of N , that is, a component of $\overline{S - N}$. The collection of complementary regions of N is denoted by $\mathcal{R}_{\text{comp}} = \mathcal{R}_{\text{comp}}(N)$. We set $\mathcal{R} = \mathcal{R}(N) = \mathcal{R}_{\text{tie}}(N) \cup \mathcal{R}_{\text{comp}}(N)$. For each complementary component $R \in \mathcal{R}_{\text{comp}}$, the *horizontal boundary* and *vertical boundary* of R are defined as $\partial_h R = \partial R \cap \partial_h N$ and $\partial_v R = \partial R \cap \partial_v N$. So, for any $R \in \mathcal{R}_{\text{comp}}$ there is a, possibly empty, subset $B \subset \partial S$ such that $\partial R = \partial_h R \cup \partial_v R \cup B$. Connected subsets of the horizontal and vertical boundary of each complementary region $R \in \mathcal{R}_{\text{comp}}$ meet perpendicularly at their boundaries.

Suppose that $T \subset S$ is a subsurface with ∂T being a finite union of one-

dimensional smooth submanifolds meeting perpendicularly at their boundaries. We say that c is a *corner* of T if c lies in the perpendicular intersection of two one-dimensional smooth submanifolds of ∂T . The set of all corners of T is denoted by $\partial^2 T$. Following [MMS12, Section 2.3], we set

$$\text{index}(T) = \chi(T) - \frac{\partial^2 T^-}{4} + \frac{\partial^2 T^+}{4},$$

where $\partial^2 T^-$ and $\partial^2 T^+$ denote the number of outward- and inward-pointing corners of T respectively. We note that the index is additive under unions if the interiors of the two subsurfaces in question are disjoint.

Suppose that $N = N(\tau)$ is a tie neighbourhood of a pretrack τ and $R \in \mathcal{R}_{\text{comp}}$ is a finite-sided complementary region of N . Then all corners of R are pointing outwards and the index simplifies to

$$\text{index}(R) = \chi(R) - \frac{|\partial^2 R|}{4}.$$

Since the tie neighbourhood of a train track is a union of rectangles whose interiors are disjoint, we know that $\text{index}(N) = 0$.

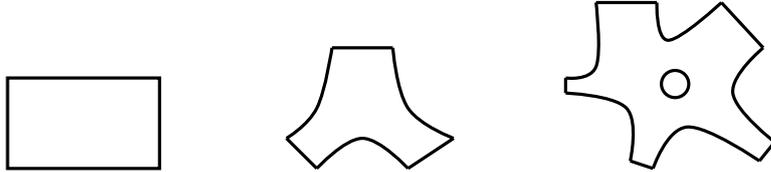


Figure 2.4: These subsurfaces have indices 0, $-2/4$, and $-10/4$ respectively.

Definition 2.2.3. Suppose that $\tau \subset S$ is a pretrack and $N = N(\tau)$ is a tie neighbourhood of τ in S . We say that τ is a *train track* if τ is compact and every complementary region R of N has negative index.

Following [MM99, Section 3.1], we say that a train track $\tau \subset S$ is *large* if all complementary regions of $N(\tau)$ are discs or peripheral annuli. For the remainder of this thesis, we restrict ourselves to large train tracks.

2.3 Snippets and winding numbers

Let $S = S_{g,b}$ be a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. Let $\tau \subset S$ be a large train track and $N = N(\tau)$ be a tie neighbourhood of τ in S .

2.3.1 Snippets

As seen in the previous section, N is the union of branch and switch rectangles $R \in \mathcal{R}_{\text{tie}}$. Each rectangle is modelled on a rectangular box. Boxes for switch rectangles are equipped with two short horizontal dashes on their vertical boundary indicating the gluings with the adjacent branch rectangles. As τ is a large train track, each complementary region can be modelled on a, perhaps peripheral, polygon. Hence, \mathcal{R} provides a tiling of the surface S by rectangles, polygons, and peripheral annuli. For any region $R \in \mathcal{R}$, we label subsets of ∂R by h or v if they are part of the horizontal or vertical boundary of R respectively. All subsets of ∂R that are (identified with) a tie of a branch rectangle are labelled by t . For an illustration see Figures 2.5 and 2.6.

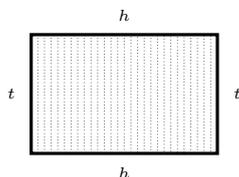


Figure 2.5: A branch rectangle

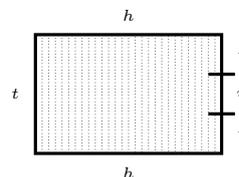


Figure 2.6: A switch rectangle

We define $\partial\mathcal{R}$ to be the union of all boundaries of the regions $R \in \mathcal{R}$. That is, $\partial\mathcal{R} = \bigcup_{R \in \mathcal{R}} \partial R$ is a trivalent graph whose edges meet at an angle of π or $\pi/2$. We further set $\partial^2\mathcal{R}$ to be the union of all points in $\partial^2 R$ for $R \in \mathcal{R}$. Suppose that $R \in \mathcal{R}$ is a rectangle or polygon. In the following, we refer to the closure of a component of $\partial R - \partial^2 R$ as a *side* of R . Hence, for $R \in \mathcal{R}_{\text{comp}}$, a side of R can contain more than two points of $\partial^2\mathcal{R}$ if it is part of the horizontal boundary of R (see Figure 2.7). If $R \in \mathcal{R}_{\text{tie}}$ is a switch rectangle, then one vertical boundary side of R contains two points of $\partial^2\mathcal{R}$ in its interior. In our figures of switch rectangles, these two extra corners are indicated by short horizontal dashes.

Definition 2.3.1. Suppose that $R \in \mathcal{R}$. Suppose that $D = (I, \partial I)$ or $D = (S^1, \emptyset)$. We say that $a : D \rightarrow (R, \partial R)$ is a *snippet* if a is a transverse and self-transverse immersion of pairs.

We note that the transversality of the map $D \rightarrow (R, \partial R)$ implies that ∂D misses the corners of R . In an abuse of notation, we denote by a the snippet as well as its image in the surface S . For some examples of snippets we refer the reader to Figure 2.8.

For the remainder of this thesis, we assume that S^1 is parametrized as $[0, 1]/\sim$. Hence, every snippet carries a canonical orientation induced by its parametrization.

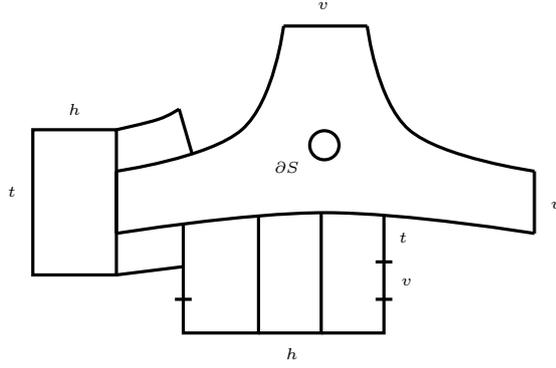


Figure 2.7: A horizontal boundary component of $R \in \mathcal{R}_{\text{comp}}$ containing more than two points of $\partial^2\mathcal{R}$.

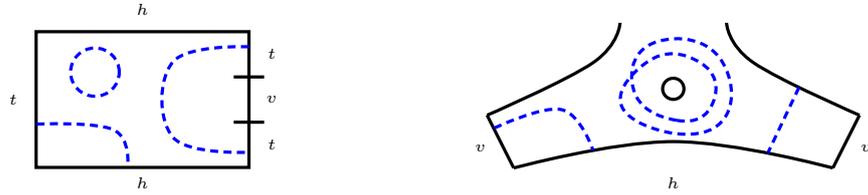


Figure 2.8: Snippets in a switch rectangle and complementary region.

Definition 2.3.2. Suppose that $a_1, a_2 : D \rightarrow R$ are two snippets contained in the same region $R \in \mathcal{R}$. We say that a_1 and a_2 are *strongly snippet homotopic* if a_1 and a_2 are homotopic via a transverse homotopy of pairs $D \rightarrow (R, \partial R)$ which keeps the k -skeleton of $[0, 1]$ or S^1 inside the $(k + 1)$ -skeleton of \mathcal{R} at all times. The strong snippet homotopy class of a snippet $a \subset R \in \mathcal{R}$ will be denoted by $[a]$.

We note that strong snippet homotopies preserve the orientation of the snippets.

Remark 2.3.3. In the further course of this thesis, a coarser equivalence relation on snippets will turn out to be important, too. This will be called *weak snippet homotopy*. Homotopies of this kind are only required to avoid the corners of the region that contains the snippet, and not all corners of $\partial^2\mathcal{R}$. For a formal statement, we point the reader to Definition 2.5.14 on page 26.

Lemma 2.3.4. *Suppose that $a \subset R$ is a snippet that lies inside a simply connected region $R \in \mathcal{R}$. Then $[a]$ is, up to orientation, uniquely determined by the components of $\partial\mathcal{R} - \partial^2\mathcal{R}$ containing ∂a . \square*

For peripheral regions $R \in \mathcal{R}$, recording the intersection with the components of $\partial\mathcal{R} - \partial^2\mathcal{R}$ is not sufficient to determine a strong snippet homotopy class. In

addition to the intersection data we have to provide information about the winding behaviour of the snippet around the boundary component of S .

2.3.2 Winding number

Traditionally, the winding number of an arc or curve around a point is an integer representing the total number of times that the arc or curve travels around that point. However, the combinatorial set-up of this thesis makes it convenient to take a different approach: suppose that $R \in \mathcal{R}_{\text{comp}}$ is a peripheral annulus and $a \subset R$ is a snippet. We set the winding number of a to equal the number of outward-pointing corners of R that a passes while travelling around the component of ∂S . Hence, winding numbers in this thesis are integers but do not count an integer number of winding.

Suppose that $R \in \mathcal{R}$. Recall that we refer to the closure of a component of $\partial R - \partial^2 R$ as a side of R . By construction of the tie neighbourhood, the sides of R are smoothly embedded arcs meeting perpendicularly at their endpoints. For any snippet $a \subset R$ we may assume that a meets $\partial R \cup \partial S$ perpendicularly.

Definition 2.3.5. Suppose that $R \in \mathcal{R}_{\text{comp}}$ is a $(2n + 1)$ -sided peripheral annulus. Suppose that $a \subset R$ is a snippet of minimal self-intersection. We assign a *winding number* to a as follows:

- If $\partial a \cap \partial S \neq \emptyset$, we set $\text{wind}(a) = 0$.
- If $\partial a \neq \emptyset$ and $\partial a \subset \partial_h R \cup \partial_v R$, we consider a lift \tilde{a} of a to the universal cover \tilde{R} of R . This lift bounds a region $Q \subset \tilde{R}$ of finite index. In this case, the winding number of a is defined as

$$\text{wind}(a) = -4 \cdot \text{index}(Q) + 2.$$

- If $\partial a = \emptyset$, then a is a curve inside a peripheral annulus. As $\pi_1(R) = \mathbb{Z}$, a is homotopic to the k -th power of a component of ∂S , where $k \in \mathbb{Z}$, and we say that $\text{wind}(a) = k \cdot 2n$.

We further decorate $\text{wind}(a)$ with a sign obtained as follows: The winding number is equipped with a positive sign if $\tilde{a} \subset \tilde{R}$ with its induced orientation from a bounds the finite index region Q or a strip with the lift of ∂R on its right-hand side. Else, we equip $\text{wind}(a)$ with a negative sign.

For examples of snippets and their winding numbers we refer the reader to Figures 2.9-2.12.

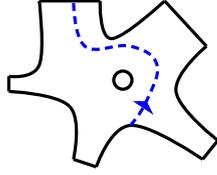


Figure 2.9: Winding number 5.

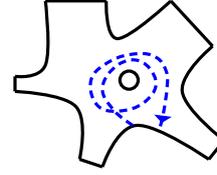


Figure 2.10: Winding number -20 .

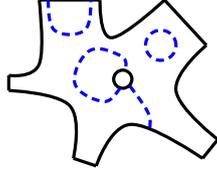


Figure 2.11: Winding number 0.

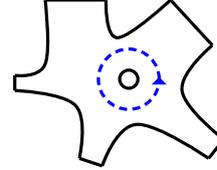


Figure 2.12: Winding number 10.

Any two snippets $a_1, a_2 \in \mathcal{R}_{\text{comp}}$ that have minimal self-intersection and lie in the same snippet homotopy class satisfy $\text{wind}(a_1) = \text{wind}(a_2)$. Hence, we can set $\text{wind}([a]) = \text{wind}(a)$ where $a \in [a]$ is any snippet of minimal self-intersection inside the peripheral annulus R . According to our set-up, $\text{wind}(a)$ counts the minimal number of corners of R that a passes as it travels within the region R , possibly winding around ∂S . Here, winding around the component of ∂S in accordance with its orientation induced by the orientation of S , results in a positive winding number.

Lemma 2.3.6. *Suppose that $a \subset R$ is a snippet that lies inside a peripheral annulus $R \in \mathcal{R}$. Then $[a]$ is uniquely determined by*

- *the sides of R (in order) containing $a(0)$ and $a(1)$ and*
- *its winding number.*

Proof. This follows from the fact that strong snippet homotopies cannot move the points of ∂a out of their containing sides. We remark that the winding number is redundant in case of snippets that have non-empty intersection with ∂S . \square

2.4 Efficient position and classification of bad snippets

Let $S = S_{g,b}$ be a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. Let $\tau \subset S$ be a large train track and $N = N(\tau)$ be a tie neighbourhood of τ in S . From this point on we no longer distinguish between snippets and their strong homotopy classes unless otherwise stated. For any snippet $a \subset R \in \mathcal{R}$, we always assume that a has minimal self-intersection, is perpendicular to ∂R , and misses $\partial^2 \mathcal{R}$. Thus, snippets contained in simply connected regions are assumed to be embedded.

Definition 2.4.1. Suppose that $a \subset R \in \mathcal{R}_{\text{tie}}$ is a snippet. We say that a is *carried* by N if a is transverse to the ties of R .

Definition 2.4.2. Suppose that $\tau \subset S$ is a large train track and $N = N(\tau)$ is a tie neighbourhood of τ in S . Suppose that $a \subset R$ is a snippet, where $R \in \mathcal{R}(N) = \mathcal{R}$. We say that a is in *efficient position with respect to N* if exactly one of the following conditions holds:

- [*Track*]: $R \in \mathcal{R}_{\text{tie}}$ and a is a tie of N or a is carried by N .
- [*Disc*]: $R \in \mathcal{R}_{\text{comp}}$ is a disc and a divides R into two regions each of which has non-positive index.
- [*Passing through peripheral annulus*]: $R \in \mathcal{R}_{\text{comp}}$ is a peripheral annulus, $\emptyset \neq \partial a \subset \partial \mathcal{R} - \partial S$ and $|\text{wind}(a)| \geq 2$.
- [*Start or end in peripheral annulus*]: $R \in \mathcal{R}_{\text{comp}}$ is a peripheral annulus, $\partial a \neq \emptyset$ and exactly one of the points in ∂a lies on the boundary of S .

Otherwise, we say that a is a *bad snippet with respect to N* .

Suppose that $a \subset R$ is a snippet, where $R \in \mathcal{R}$. We say that a is *dual* to N if a is in efficient position with respect to N but not carried by N . In other words, snippets in efficient position are either carried or dual.

For examples of snippets in efficient position we refer the reader to Figures 2.13-2.16.

Rather than considering examples of bad snippets, we provide a complete classification of them up to symmetries and orientation in the next section. We close the current section by pointing out two observations following the definition of efficient position.

Lemma 2.4.3. *Suppose that $a \subset R \in \mathcal{R}_{\text{tie}}$ is a snippet. Then a is a bad snippet with respect to N if and only if a cuts off a region in R that has positive index. \square*

Lemma 2.4.4. *Suppose that $a \subset R \in \mathcal{R}_{\text{comp}}$ is a snippet. Then a is a bad snippet with respect to N if and only if a is a peripheral curve or a is embedded and cuts off a region of R that has positive index.*

Proof. If R is homeomorphic to a disc, the statement follows directly from the definition. Hence, let us assume that R is a peripheral annulus. As a is a snippet, it is an immersion of S^1 or $[0, 1]$ into R .

If a is an immersion of the circle into R it follows from the definition that a must be bad as its boundary is empty. We note that any immersion of S^1 into a

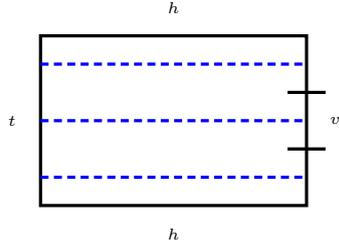


Figure 2.13: Examples of carried snippets in a switch rectangle.

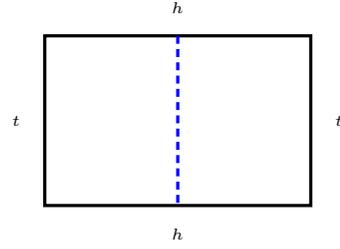


Figure 2.14: A dual snippet inside a branch rectangle.

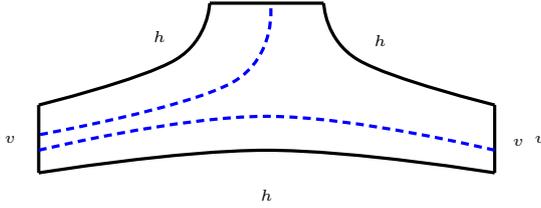


Figure 2.15: Examples of snippets in efficient position that lie inside a simply connected complementary region.

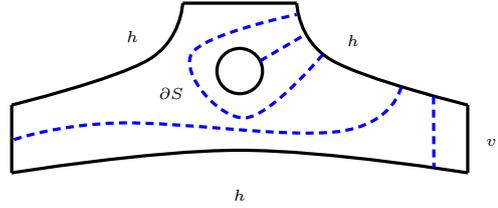


Figure 2.16: Examples of snippets in efficient position that lie inside a peripheral complementary region.

peripheral annulus is an inessential or peripheral curve. Since we assume that our snippets have minimal self-intersection, any inessential curve cuts off a region of index one. Hence, snippets that are an immersion of S^1 into R are bad if and only if they are a peripheral curve or cut off a region of R of positive index.

On the other hand, assume that a is a proper immersion of $[0, 1]$ into R . Hence, $|\partial a| = 2$, and one of the following three statements holds:

- $\partial a \subset \partial S$.
- $\partial a \subset \partial R - \partial S$.
- Exactly one of the points of ∂a lies in ∂S .

If a is a snippet satisfying $\partial a \subset \partial S$, then the definition of efficient position implies that a is a bad snippet. Since we assume that snippets have minimal self-intersection, are smooth, and intersect $\partial S \subset \partial R$ transversely, a must be embedded and divides R into two regions, one of which has exactly two outward-pointing corners occurring at the points $a(0)$ and $a(1)$. Hence, this region has index $1/2$. Thus, if $\partial a \subset \partial S$, then a is a bad snippet if and only if it is embedded and cuts off a region of R of positive index.

If a is a snippet satisfying $\partial a \subset \partial R - \partial S$, then a is bad if and only if $|\text{wind}(a)| < 2$. This is the case if and only if a cuts off a region of R that contains at

most one point of ∂R . This is true if and only if a cuts off a region of index $1/4$ or $1/2$. The last equivalence follows from the fact that the region cut off by a must be a disc with at least two outward-pointing corners occurring at the points $a(0)$ and $a(1)$. Thus, if $\partial a \subset \partial R - \partial S$, then we also see that a is a bad snippet if and only if it is embedded and cuts off a region of positive index in R .

Lastly, if exactly one of the points of ∂a lies in ∂S , then the snippet a is in efficient position according to the definition. We remark that a does not cut off a region of positive index in this case as complementary regions of train tracks have negative index. Thus, a snippet $a \subset R \in \mathcal{R}_{\text{comp}}$ is bad if and only if it is a peripheral curve or cuts off a region of R of positive index. \square

2.5 Classification of bad snippets

Definition 2.5.1. Suppose that $R \in \mathcal{R}$ is a region. We say that a self-homeomorphism $\phi : R \rightarrow R$ is a *symmetry* of R if ϕ preserves $\partial_h R$ and $\partial_v N \cap R$ setwise.

Remark 2.5.2. It may seem more natural to define a symmetry of a region as a self-homeomorphism that preserves $\partial_h R$ and $\partial_v R$ setwise. However, we need to remember if the boundary of a snippet lies inside $\partial_v N$ or on a tie of an adjacent branch rectangle, as adjacent snippets in these two cases are of very different types. Thus, we require symmetries to preserve $\partial_h R$ and $\partial_v N \cap R$ setwise.

In the following, we are classifying bad snippets up to symmetry and orientation.

2.5.1 Classification of bad snippets in the tie neighbourhood

According to Lemma 2.4.3, a snippet inside a branch or switch rectangle is bad if and only if it cuts off a region T of positive index. We claim that the index of T must be $1/4$, $1/2$, or 1 . If the index of T is $1/4$ (respectively $1/2$ or 1), we say that T is a *trigon* (respectively a *bigon* or a *disk*).

Lemma 2.5.3. *Suppose that $R \in \mathcal{R}_{\text{tie}}$ is a branch or switch rectangle. Suppose further that $a \subset R$ is a snippet that cuts off a region T of R of strictly positive index. Then T is a trigon, bigon, or disk.*

Proof. Since $a \subset R$ is a snippet, we know that a is a properly embedded arc in R or an embedded curve in R . As R is a rectangle, any embedded curve in R is trivial and cuts off a disk in R . If a is a properly embedded arc, then a intersects ∂R transversely. Thus, the region T of positive index cut off by a of R must have at

least two outward-pointing corners. As T must be homeomorphic to a disk, it has Euler characteristic one, so $0 \leq \text{index}(T) \leq 1/2$ and the claim follows. \square

Lemma 2.5.4. *Suppose that $R \in \mathcal{R}_{tie}$ is a branch rectangle. Up to symmetries of R and orientation of the snippet, there are four strong snippet homotopy classes of bad snippets in R .*

Proof. Suppose that $a \subset R$ is a bad snippet inside a branch rectangle R . Following Lemma 2.4.3 and Lemma 2.5.3, a cuts off a region $T \subset R$ which is a trigon, bigon, or disk. If T is a disk, then a is a trivial curve in R . If T is a bigon, then $a(0)$ and $a(1)$ must lie on the same side of R . Up to symmetries of R , there are two different kinds of snippets of this type: snippets where $a(0)$ and $a(1)$ lie on a horizontal boundary side of R and snippets where $a(0)$ and $a(1)$ lie on a vertical boundary side of R . If T is a trigon, then ∂T contains exactly one corner of R . Hence, $a(0)$ and $a(1)$ lie on adjacent sides of R , that is, one lies on a horizontal boundary side of R and one on a vertical boundary side of R . All such snippets are equivalent up to symmetries of R and a choice of orientation of the snippets.

As snippets of these four kinds have a distinct intersection pattern with ∂R , they are non-equivalent under symmetries of R which concludes the proof of the lemma. \square

For examples of bad snippets inside branch rectangles, we refer the reader to Figure 2.17.

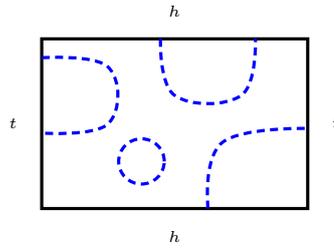


Figure 2.17: All possible types of bad snippets inside a branch rectangle.

Definition 2.5.5. Suppose that $a \subset R$ is a bad snippet inside a branch rectangle $R \in \mathcal{R}$. We say that a is of type $\mathbb{B}(x, y)$ where

- $(x, y) = (\overset{\circ}{R}, \overset{\circ}{R})$ if $\partial a = \emptyset$.
- $x, y \in \{h, t\}$ if $\partial a \neq \emptyset$, with $x = y = h$ or $x = y = t$ if and only if $\partial a \subset \partial_h R$ or $\partial_v R$ respectively. Else, $(x, y) = (h, t)$.

Corollary 2.5.6. *Suppose that $R \in \mathcal{R}_{tie}$ is a branch rectangle. Suppose further that $a \subset R$ is a bad snippet. Then a is of type $\mathbb{B}(\dot{R}, \dot{R})$, $\mathbb{B}(h, h)$, $\mathbb{B}(t, t)$, or $\mathbb{B}(h, t)$. Two bad snippets in R are of the same type if and only if they are equivalent up to symmetries of R and orientation. Hence, if two bad snippets $a_1, a_2 \subset R$ are strongly snippet homotopic, then they are of the same type. Moreover, if two bad snippets $a_1, a_2 \subset R$ are of the same type and intersect the same components of $\partial R - \partial^2 R$, they must be strongly snippet homotopic up to orientation. \square*

Thus, up to orientation and symmetries of R , there are four different types of bad snippets inside a branch rectangle.

We now proceed by classifying bad snippets inside switch rectangles. As one vertical side of each switch rectangle contains a component of $\partial_v N$ in its interior, not all snippets intersecting the horizontal and vertical boundary of the switch rectangle turn out to be equivalent under symmetries of R . To aid the classification of bad snippets in switch rectangles, we introduce the notion of *weight* for such a bad snippet.

Definition 2.5.7. Suppose that $R \in \mathcal{R}_{tie}$ is a switch rectangle. Suppose further that $a \subset R$ is a snippet that cuts off a region T of R of positive index. We set $\text{weight}(a) = |\partial^2 \mathcal{R} \cap \partial T|$ and say that $\text{weight}(a)$ is the *weight* of the snippet $a \subset R$.

For examples of snippets and their weight, we refer the reader to Figure 2.18.

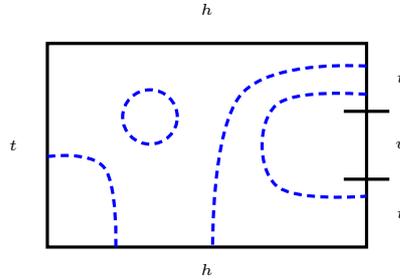


Figure 2.18: Snippets inside a switch rectangle of weight zero, one, two, and three respectively.

Remark 2.5.8. As switch rectangles are simply connected, any two bad snippets belonging to the same strong snippet homotopy class have equal weight. We further remark that the weight of any bad snippet inside a switch rectangle is bounded by three.

Lemma 2.5.9. *Suppose that $R \in \mathcal{R}_{tie}$ is a switch rectangle. Up to symmetries of R and orientation of the snippet, there are eleven strong snippet homotopy classes of bad snippets in R . Each class $[a]$ is uniquely determined by*

- *its weight and*
- *the information whether $a(0)$ and $a(1)$ each are contained in a horizontal boundary side of R that has empty or non-empty intersection with $\partial_v N$, a vertical boundary side of R , or a component of $\partial_v N$.*

Proof. Suppose that $a \subset R$ is a bad snippet inside a switch rectangle R . Following Lemma 2.4.3 and Lemma 2.5.3, a cuts off a region T of R that is a trigon, bigon, or disk. If T is a disk, then a is a trivial curve in R .

If T is a bigon, then $a(0)$ and $a(1)$ lie on the same side of R . Recalling that symmetries of R preserve $\partial_v N$ and $\partial_h N$ setwise, we see that they also preserve the single vertical boundary sides of R setwise. Hence, up to symmetries of R , we obtain the following possibilities for a snippet a to cut off a bigon:

- $a(0)$ and $a(1)$ lie on a horizontal boundary side of R .
- $a(0)$ and $a(1)$ lie on a vertical boundary side of R which has empty intersection with $\partial_v N$.
- $a(0)$ and $a(1)$ lie on a vertical boundary side of R which has non-empty intersection with $\partial_v N$. In this case, the region T can contain up to two points of $\partial^2 \mathcal{R}$ in its boundary. As we want to classify snippets up to strong snippet homotopy, we further distinguish snippets according to their weight:
 - $\text{weight}(a) = 0$. Then $a(0)$ and $a(1)$ both either lie in $\partial_v N$ or in the same component of $\partial_v R - \partial_v N$.
 - $\text{weight}(a) = 1$. Then one point of ∂a lies in $\partial_v N$ and one point of ∂a lies in $\partial_v R - \partial_v N$.
 - $\text{weight}(a) = 2$. Then $a(0)$ and $a(1)$ lie in different components of $\partial_v R - \partial_v N$.

Thus, up to symmetries of R and the orientation of the snippet, there are six different strong snippet homotopy classes of bad snippets that cut off a bigon of R . For an illustration we refer the reader to Figure 2.19.

If T is a trigon, then $a(0)$ and $a(1)$ lie on different sides of R . Up to orientation of the snippet, we may assume that $a(0)$ lies on a horizontal side of R . Thus, $a(1)$ either lies on the vertical side of R that has empty intersection with $\partial_v N$ or on the vertical side of R that has non-empty intersection with $\partial_v N$. In the latter case, the snippet is uniquely defined by its weight $w(a) \in \{1, 2, 3\}$. Thus, up to symmetries of R and the orientation of the snippet, there are four different strong snippet homotopy

classes of bad snippets that cut off a trigon of R . For an illustration we refer the reader to Figure 2.20.

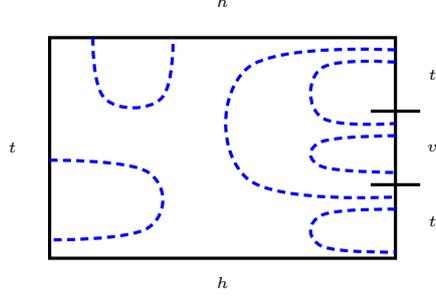


Figure 2.19: The six different strong snippet homotopy classes of bigon snippets inside a switch rectangle, up to symmetry.

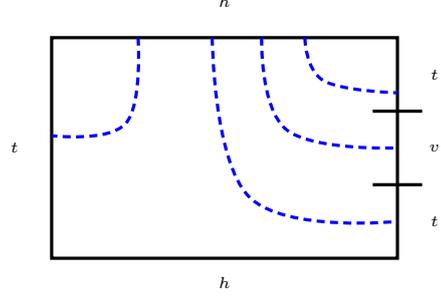


Figure 2.20: The four different strong snippet homotopy classes of trigon snippets inside a switch rectangle, up to symmetry.

□

Definition 2.5.10. Suppose that $a \subset R$ is a bad snippet inside a switch rectangle $R \in \mathcal{R}$. We say that a is of type $\mathbb{S}(x, y, k)$ where

- $(x, y) = (\overset{\circ}{R}, \overset{\circ}{R})$ if $\partial a = \emptyset$.
- $x = h$ (respectively $x = v$ or $x = t$) if $a(0) \subset \partial_h R$ (respectively $a(0) \subset \partial_v N$ or $a(0) \subset (\partial_v R - \partial_v N)$).
- $y = h$ (respectively $x = v$ or $x = t$) if $a(1) \subset \partial_h R$ (respectively $a(1) \subset \partial_v N$ or $a(1) \subset (\partial_v R - \partial_v N)$).
- $k = \text{weight}(a) \in \{0, 1, 2, 3\}$.

In the following, we will not distinguish between the types $\mathbb{S}(x, y, k)$ and $\mathbb{S}(y, x, k)$.

Corollary 2.5.11. Suppose that $R \in \mathcal{R}_{tie}$ is a switch rectangle. Suppose further that $a \subset R$ is a bad snippet. Then a is of one of the following nine types: $\mathbb{S}(\overset{\circ}{R}, \overset{\circ}{R}, 0)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, $\mathbb{S}(t, v, 1)$, $\mathbb{S}(t, t, 2)$, $\mathbb{S}(h, t, 1)$, $\mathbb{S}(h, v, 2)$, or $\mathbb{S}(h, t, 3)$. If two snippets in R are not of type $\mathbb{S}(t, t, 0)$ or $\mathbb{S}(h, t, 1)$, then they are equivalent up to symmetries of R and orientation of the snippets if and only if they are of the same type. If two bad snippets $a_1, a_2 \subset R$ are strongly snippet homotopic, then they are of the same type. Moreover, if two bad snippets $a_1, a_2 \subset R$ are of the same type and intersect the same components of $\partial R - \partial^2 R$, then they must be strongly snippet homotopic up to orientation. □

Summarizing, we see that up to symmetries of the switch rectangle and orientation of the snippet, there are eleven different strong snippet homotopy classes of bad snippets inside switch rectangles. However, our combinatorial set-up requires us to only distinguish between nine different types of these: two bad snippets of type $\mathbb{S}(t, t, 0)$ or $\mathbb{S}(h, t, 1)$ might not be related by a symmetry of R , but turn out to behave very similarly in all relevant situations.

2.5.2 Bad snippets in complementary regions

Definition 2.5.12. Suppose that $a \subset R$ is a non-peripheral bad snippet inside a complementary region $R \in \mathcal{R}_{\text{comp}}$. We say that a is of type $\mathbb{R}(x, y)$ where

- $(x, y) = (\mathring{R}, \mathring{R})$ if $\partial a = \emptyset$.
- $(x, y) = (\partial S, \partial S)$ if $\partial a \subset \partial S$.
- $x, y \in \{h, v\}$ with $x = y = h$ or $x = y = v$ if and only if $\partial a \subset \partial_h R$ or $\partial_v R$ respectively. Else, $(x, y) = (h, v)$.

For an illustration of the different types of non-peripheral snippets inside complementary regions we point the reader to Figure 2.21.

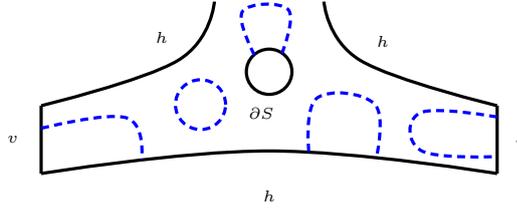


Figure 2.21: The five types of non-peripheral bad snippets inside a complementary region.

As in the previous section, we see that every non-peripheral bad snippet inside a complementary region is of one of the just defined five types:

Lemma 2.5.13. *Suppose that $a \subset R$ is a non-peripheral bad snippet inside a complementary region $R \in \mathcal{R}_{\text{tie}}$. Then a is of type $\mathbb{S}(\mathring{R}, \mathring{R})$, $\mathbb{S}(\partial S, \partial S)$, $\mathbb{S}(h, h)$, $\mathbb{S}(v, v)$, or $\mathbb{S}(h, v)$. Disregarding orientation, two non-peripheral bad snippets in R are of the same type if and only if they are equivalent up to symmetries of R .*

Proof. Suppose that $a \subset R$ is a non-peripheral bad snippet inside a complementary region R . By Lemma 2.4.4 we know that a is embedded and cuts off a region T of R of positive index. If $a : S^1 \rightarrow R$, it, therefore, bounds a disc inside R and is of type $\mathbb{S}(\mathring{R}, \mathring{R})$. Any two discs in R are homeomorphic to each other. Thus,

two non-peripheral bad snippets $a : S^1 \rightarrow R$ of the same type are equivalent up to symmetries of R .

If $a : [0, 1] \rightarrow R$ is a proper immersion of the interval into R , then either $\partial a \subset \partial S$ or $\partial a \subset \partial R - \partial S$. This follows from the fact that snippets with exactly one of their boundary points in ∂S are in efficient position.

A snippet satisfying $\partial a \subset \partial S$ is, by definition, of type $\mathbb{S}(\partial S, \partial S)$. As any two such snippets equivalent under symmetries of R , the claim of the lemma follows in this case.

Suppose that $a : [0, 1] \rightarrow R$ is a bad snippet satisfying $\partial a \subset \partial R - \partial S$. As a meets ∂R perpendicularly, the region T cut off by a in R must have at least two outward-pointing corners. Since a is smooth, T has positive index and R has outward-pointing corners only, T must be a disk whose boundary contains either none or exactly one corner of R . If T contains no corner of R , then $a(0)$ and $a(1)$ lie on the same component of $\partial R - \partial^2 R$. Hence, $\text{index}(T) = 1/2$ and a is of type $\mathbb{S}(h, h)$ or $\mathbb{S}(v, v)$. Up to orientation, any two snippets of type $\mathbb{S}(h, h)$ (respectively $\mathbb{S}(v, v)$) are equivalent under symmetries of R . If T contains one corner of R , then $a(0)$ and $a(1)$ lie on different components of $\partial R - \partial^2 R$. Hence, $\text{index}(T) = 1/4$ and a is of type $\mathbb{S}(h, v)$. Up to orientation, any two snippets of type $\mathbb{S}(h, v)$ are equivalent under symmetries of R . \square

We remind ourselves that horizontal sides of complementary regions can contain several points of $\partial^2 \mathcal{R}$ in their interiors. Thus, in contrary to the case of branch and switch rectangles, two non-peripheral bad snippets inside a complementary region R that are of the same type and intersect the same sides of R need not be strongly snippet homotopic. For an example of two snippets that intersect the same sides of a complementary region but are not strongly snippet homotopic we refer the reader to Figure 2.22.

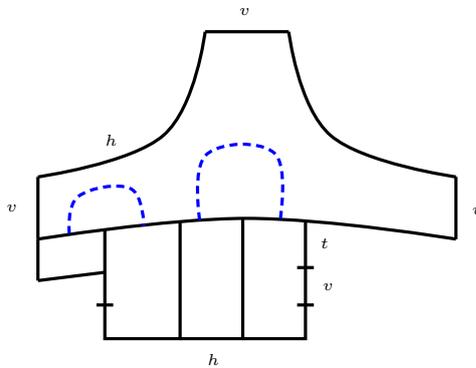


Figure 2.22: Weakly but not strongly snippet homotopic snippets.

However, snippets in complementary regions that are homotopic up to “moving their boundary points” along the sides of the region play an important role in this thesis. We, therefore, close this section by introducing the following coarser equivalence relation on snippets:

Definition 2.5.14. Fix a region $R \in \mathcal{R}$. Suppose that $a_1, a_2 : D \rightarrow R$ are two snippets in R . We say that a_1 and a_2 are *weakly snippet homotopic* if a_1 and a_2 are homotopic via a transverse homotopy of pairs $D \rightarrow (R, \partial R)$ which keeps the k -skeleton of $[0, 1]$ or S^1 inside the $(k + 1)$ -skeleton of R at all time.

Lemma 2.5.15. *Suppose that $a_1, a_2 : D \rightarrow R$ are two snippets inside a region $R \in \mathcal{R}$. If a_1 and a_2 are weakly snippet homotopic and a_1 is in efficient position, then a_2 is in efficient position, too.* \square

2.5.3 Classification of bad snippets - an overview

The following corollary is an immediate consequence of the discussion in the previous section:

Corollary 2.5.16. *Suppose that $a \subset R \in \mathcal{R}$ is a bad snippet. Then a is either a peripheral curve or it is of type*

- $\mathbb{B}(\mathring{R}, \mathring{R}), \mathbb{S}(\mathring{R}, \mathring{R}, 0)$, or $\mathbb{R}(\mathring{R}, \mathring{R})$, that is, a is an inessential curve. In this case we say that a is a trivial snippet.
- $\mathbb{R}(\partial S, \partial S)$. In this case we say that a is an inessential bigon snippet.
- $\mathbb{B}(h, h), \mathbb{B}(t, t), \mathbb{S}(h, h, 0), \mathbb{S}(t, t, 0), \mathbb{S}(v, v, 0), \mathbb{S}(t, v, 1), \mathbb{S}(t, t, 2), \mathbb{R}(h, h)$, or $\mathbb{R}(v, v)$. That is, a cuts off a bigon of R . In this case we say that a is a bigon snippet.
- $\mathbb{B}(h, t), \mathbb{S}(h, t, 1), \mathbb{S}(h, v, 2), \mathbb{S}(h, t, 3)$, or $\mathbb{R}(h, v)$. That is, a cuts off a trigon of R . In this case we say that a is a trigon snippet.

Hence, there are four and nine types of bad snippets inside branch and switch rectangles respectively. There are five types of non-peripheral bad snippets inside complementary regions. \square

2.6 Efficient position for arcs and curves

Let $S = S_{g,b}$ be a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. Let $\tau \subset S$ be a large train track and $N = N(\tau)$ be a tie neighbourhood of τ in S .

Suppose that $\alpha \subset S$ is an immersed arc or curve in S . For the remainder of this thesis, we always assume that α is self-transverse and transverse to $\partial\mathcal{R}$. Moreover, if α is an arc, we assume that $\alpha : (I, \partial I) \rightarrow (S, \partial\mathcal{R})$ is an immersion of pairs. Hence, $\alpha(0)$ and $\alpha(1)$ lie in $\partial\mathcal{R}$ and α admits a canonical decomposition into snippets.

Definition 2.6.1. Suppose that $\alpha \subset S$ is a snippet-decomposed immersed arc or curve in S . The number of snippets occurring in the snippet decomposition of α is denoted by $\text{len}(\alpha)$ and is called the *snippet length* of α .

A snippet-decomposed arc or curve $\alpha \subset S$ is said to be *carried* by N or *dual* to N if all its snippets are carried or dual respectively.

Definition 2.6.2. Suppose that $\tau \subset S$ is a large train track and $N = N(\tau)$ is a tie neighbourhood of τ in S . Suppose that $\alpha \subset S$ is an immersed arc or curve in S . We say that α is in *efficient position with respect to N* if all snippets in its snippet-decomposition are in efficient position with respect to N .

We remark that the intersection of carried and dual subarcs of an arc or curve in efficient position must lie in $\partial_v N$ (see Figure 2.23).

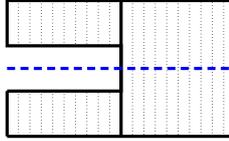


Figure 2.23: Adjacent snippets of different types in an arc or curve in efficient position.

The following observations on short properly immersed arcs and curves follow directly from the classification of snippets in the previous sections:

Lemma 2.6.3. *Suppose that $\alpha \subset S$ is a properly immersed, snippet-decomposed arc or curve in S . If $\text{len}(\alpha) = 1$, then α is inessential or peripheral.*

Proof. As $\text{len}(\alpha) = 1$, the arc or curve α consists of a single snippet $a \subset R$ for some region $R \in \mathcal{R}$. If $\partial R \cap \partial S = \emptyset$, then $a = \alpha$ must be an inessential curve. If R is a peripheral region, either $\partial a = \emptyset$ and hence $a = \alpha$ is an inessential or peripheral curve, or $\partial a \subset \partial S$ as α is proper. The latter implies that a is of type $\mathbb{R}(\partial S, \partial S)$. Hence, α is an inessential arc. \square

Lemma 2.6.4. *Suppose that $\alpha \subset S$ is a properly immersed, snippet-decomposed arc in S . Then $\text{len}(\alpha) \neq 2$. If $\text{len}(\alpha) \geq 3$, then the snippets containing $\alpha(0)$ and $\alpha(1)$ are in efficient position.*

Proof. For any properly immersed arc $\alpha \subset S$, $\alpha(0)$ and $\alpha(1)$ lie in ∂S . As any component of ∂S is contained in the boundary of a peripheral complementary region $R \in \mathcal{R}_{\text{comp}}$, this implies that the first and last snippet of α must lie in peripheral complementary regions. As α is transverse to $\partial \mathcal{R}$, no two adjacent snippets of α can lie inside the same region $R \in \mathcal{R}$. As any two complementary regions of N are separated by the tie neighbourhood, this implies that $\text{len}(\alpha) = 1$ or $\text{len}(\alpha) \geq 3$. If $\text{len}(\alpha) \geq 3$, then the snippets containing $\alpha(0)$ and $\alpha(1)$ must both intersect ∂N and ∂S , and hence are in efficient position. \square

Lemma 2.6.5. *Suppose that $\alpha \subset S$ is a properly immersed, snippet-decomposed arc or curve in S . If $\text{len}(\alpha) > 1$, then any bad snippet of α is of one of the following two kinds:*

- *a bigon snippet, that is of type $\mathbb{B}(h, h)$, $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, $\mathbb{S}(t, v, 1)$, $\mathbb{S}(t, t, 2)$, $\mathbb{R}(h, h)$, or $\mathbb{R}(v, v)$.*
- *a trigon snippet, that is of type $\mathbb{B}(h, t)$, $\mathbb{S}(h, t, 1)$, $\mathbb{S}(h, v, 2)$, $\mathbb{S}(h, t, 3)$, or $\mathbb{R}(h, v)$.*

Proof. This follows from the classification of bad snippets in Corollary 2.5.16. \square

Lemma 2.6.6. *Suppose that $N = N(\tau)$ is a tie neighbourhood of a large train track $\tau \subset S$. If α is an immersed curve which is in efficient position with respect to N , then α is essential and nonperipheral in S .*

Proof. Suppose that α is a snippet-decomposed curve that is in efficient position with respect to N . Let $\tilde{\alpha} \subset \tilde{S}$ be a preimage of α in the universal cover of S .

First, assume that α is inessential. As α represents a trivial conjugacy class in $\pi_1(S)$, we know that $\tilde{\alpha}$ is a closed curve in \tilde{S} . Let us first assume that $\tilde{\alpha}$ is embedded. Thus, $\tilde{\alpha}$ bounds an embedded disk $D \subset \tilde{S}$. We recall that \mathcal{R} provides a tiling of S by rectangles, polygons, and peripheral annuli. Hence, $\tilde{\mathcal{R}}$ provides a tiling of \tilde{S} . The induced tiling of D consists of lifts of tiles in \mathcal{R} and lifts of subsets of tiles in \mathcal{R} cut off by snippets of α . We know that $\text{index}(D) = 1$. Therefore, at least one of the tiles of D , in the following called T , must have positive index. Since tiles $R \in \mathcal{R}$ have non-positive index, they lift to non-compact tiles or tiles of non-positive index. Thus, T must be a subtile of some $\tilde{R} \in \tilde{\mathcal{R}}$ that is cut off by the lift of a snippet $a \subset \alpha$. Since T is embedded and has positive index, we know that a must be embedded and cuts off a region of positive index inside the tile $R \in \mathcal{R}$. This contradicts our assumption on α being in efficient position.

Let us now assume that $\tilde{\alpha}$ is not embedded. Since \tilde{S} is simply connected, we know that one of the regions bounded by $\tilde{\alpha}$ must be an embedded monogon following

[HS85, Lemma 1.1.]. Since the index of this monogon is greater than or equal to $3/4$, a contradiction is derived as in the previous paragraph.

Secondly, let us assume that α is peripheral. If $\tilde{\alpha}$ is not embedded, [HS85, Lemma 1.1.] implies that $\tilde{\alpha}$ bounds an embedded monogon. We derive a contradiction as in the previous cases. Therefore, let us assume that $\tilde{\alpha}$ is embedded. Thus, together with a lift $\tilde{\beta}$ of a component β of ∂S , the arc $\tilde{\alpha}$ bounds a bi-infinite strip. Let $\delta \subset \tilde{S}$ be an embedded arc connecting $\tilde{\alpha}$ with $\tilde{\beta}$. Without loss of generality, we may assume that δ is contained inside the strip bounded by $\tilde{\beta}$ and $\tilde{\alpha}$. After applying a small isotopy, we can further assume that δ meets $\tilde{\alpha}$, $\tilde{\beta}$, as well as $\partial\tilde{\mathcal{R}}$ perpendicularly. Then δ and its image under the covering transformation corresponding to α cut an embedded rectangle $A \subset \tilde{S}$ out of the bi-infinite strip bounded by $\tilde{\alpha}$ and $\tilde{\beta}$. As $\tilde{\alpha}$ and $\tilde{\beta}$ do not have any corners, A has exactly four outward-pointing corners. Thus, we know that $\text{index}(A) = 0$. Let $R \in \mathcal{R}_{\text{com}}$ be the peripheral complementary region of N whose boundary contains β . If $\tilde{\alpha} \subset \tilde{R}$, then α is a snippet-decomposed curve of length one, thus is not in efficient position. If $\tilde{\alpha} \cap \partial\tilde{R} \neq \emptyset$, then $\tilde{\alpha}$ and $\partial\tilde{R}$ co-bound an embedded bigon lying outside of \tilde{R} of index greater than or equal to $1/2$. Hence, this bigon must have at least one positive-index tile, which contradicts the assumption that all tiles cut off by snippets of $\tilde{\alpha}$ have non-positive index. Thus, we can assume that $\tilde{\alpha} \cap \tilde{R} = \emptyset$. Therefore, A must contain a tile of index smaller or equal to the index of R . Snippets of $\tilde{\alpha}$ cut off non-positive tiles. By additivity of the index, the index of tiles of A cut off by the arc δ and the index of tiles of A cut off by the image of δ under the covering transformation add up to the index of the entire tile (potentially cut off by $\tilde{\alpha}$ and $\tilde{\beta}$), which must be non-positive. Since R has strictly negative index, this implies that A must have strictly negative index as well. This yields a contradiction. \square

Lemma 2.6.7. *Suppose that $N = N(\tau)$ is a tie neighbourhood of a large train track $\tau \subset S$. If α is a properly immersed arc which is in efficient position with respect to N , then α is essential in S .*

Proof. Suppose that α is a properly immersed, snippet-decomposed arc in efficient position. Let $\tilde{\alpha} \subset \tilde{S}$ be one preimage of α in the universal cover of S . Suppose that α is inessential. If $\tilde{\alpha}$ is not embedded, then $\tilde{\alpha}$ bounds an embedded monogon following [HS85, Lemma 1.1.]. As in the proof of the previous lemma, this contradicts our assumption that α is in efficient position. Hence, let us assume that $\tilde{\alpha}$ is embedded. Suppose that $R \in \mathcal{R}_{\text{com}}$ is the peripheral complementary region that contains the boundary component which α is homotopic into. Since $\text{len}(\alpha) > 2$, we know that $\tilde{\alpha}$ bounds an embedded region of index at least $1/2$ with a subset $\partial\tilde{R}$. This implies

that there must be a positive-index subtile of $\widetilde{\mathcal{R}}$ cut off by a lift of a snippet $a \subset \alpha$. Again, this contradicts our assumption that all snippets of α are in efficient position and thus cut off only non-positive regions. \square

We recall that the goal of this thesis is to show the converse to the previous two lemmas, that is, that essential and non-peripheral arcs and curves in S can be homotoped into efficient position. We do this by constructing an explicit algorithm that homotopes essential and non-peripheral arcs and curves into efficient position and runs in polynomial-time in the length of its input.

2.7 Algorithms and Python notation for snippet-decomposed arcs and curves

We end this chapter by introducing algorithms and giving some further notation for snippet-decomposed arcs and curves. This notation is based on standard conventions within the Python programming language. For a reference, we refer the reader to [Wal13].

2.7.1 Conventions on algorithms

In this thesis, an algorithm consists of

- the name of the algorithm,
- a brief comment on the purpose of the algorithm,
- a specification of the input and output of the algorithm, and
- a block of pseudo-code.

For an example we refer the reader to Algorithm 3.5.1 on page 56. Following [CLRS09, 1], we say that an algorithm is *correct* if, for every input instance, it halts with the correct output.

Remark 2.7.1. By convention, an algorithm should verify that the input meets the specified requirements for input instances of the algorithm. For all algorithms presented in this thesis, the input consists of a surface S , a tie neighbourhood $N = N(\tau) \subset S$ of a large train track τ in S , and a snippet-decomposed arc or curve α . From a computer science point of view, it is most convenient to assume that the tie neighbourhood is given via the collection of branch and switch rectangles with corresponding identifications. Likewise, the surface is assumed to be given via

its polygonal decomposition, that is, as a collection of rectangles, polygons, and peripheral annuli with the corresponding gluings. Then, verifying the requirements for input instances are routine checks and can be done in $O(|\chi(S)| \cdot \text{len}(\alpha))$ time. Thus, these checks will be omitted from the pseudocode of the algorithm.

In the course of this thesis, algorithms often perform “look-ups” in data structures of size $O(|\chi(S)|)$ (see for example page 44, Remark 3.4.5). Examples of such data structures include the list of all sides of regions in \mathcal{R} or lists of tuples of sides that yield bad snippets. In general, the length of the arc or curve is a greater contributor to the running time of the algorithms than $O(|\chi(S)|)$. Therefore, we do not aim to give best-possible bounds on the number of operations required to perform such “look-up” operations. Instead, we simply assume that they can be done in $O(|\chi(S)|)$ time without worrying about the exact computational models.

2.7.2 Python notation

Let α be a snippet-decomposed arc or curve of length $k = \text{len}(\alpha)$. Recall that we parametrize S^1 as $[0, 1]/\sim$. We always consider snippet-decomposed arcs and curves to be oriented. We say that the *first snippet* of α is the snippet that begins with $\alpha(0)$ and that the *last snippet* of α is the snippet that ends with $\alpha(1)$. Suppose that $0 \leq i < j \leq k$. We set $\alpha[i : j]$ to be the subarc of α that begins with the $(i + 1)$ -th snippet of α and ends with the j -th snippet of α . For $0 \leq i \leq k$ we set $\alpha[i : i]$ to be the empty subarc. We abbreviate $\alpha[i : i + 1]$ by $\alpha[i]$, and note that this gives us the $(i + 1)$ -th snippet of α . By convention, negative indices count from the end of α , that is $\alpha[-j : -i] = \alpha[k - j : k - i]$.

Suppose that $\beta, \gamma \subset S$ are two adjacent subarcs of α such that $\beta(1) = \gamma(0)$. Then, we denote their *concatenation* by $\beta \cdot \gamma$. Hence, $\alpha = \alpha[0 : i] \cdot \alpha[i : k]$ for any $0 \leq i \leq k$. To address all snippets of α from its beginning up to the i -th snippet, we use the notation $\alpha[: i] = \alpha[0 : i]$. Similarly, $\alpha[i :] = \alpha[i : k]$ denotes the subarc of α starting with the $(i + 1)$ -th snippet and ending with the last snippet of α . If α is a curve, we consider indexing to be *circular*. That is, for $0 \leq i, j < \text{len}(\alpha)$, we set $\alpha[i : \text{len}(\alpha) + j] = \alpha[i :] \cdot \alpha[: j]$.

Recall that snippets are parametrized by the unit interval. Let $0 \leq \epsilon \leq \delta \leq 1$. As $\alpha[i] = \alpha[i : i + 1]$, we set $\alpha[i + \epsilon : i + \delta] = \alpha[i][[\epsilon, \delta]]$ in an abuse of notation. We remark that with this convention, $\alpha[i] = \alpha[i : i + \epsilon] \cdot \alpha[i + \epsilon : i + \delta] \cdot \alpha[i + \delta : i + 1]$.

Definition 2.7.2. Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve in S . If α is an arc, we say that the *inside* of α consists of the snippet-decomposed arc $\alpha_{\text{trim}} = [1 : \text{len}(\alpha) - 1]$. Thus, if α is an arc and $\text{len}(\alpha) < 3$, the inside of α is empty.

If α is a curve, we set $\alpha_{\text{trim}} = \alpha$.

Chapter 3

Local homotopies and trigon arcs

In this chapter, we present the most important algorithm of this thesis, the routine `TrigArc`. Almost every further algorithm is built upon it, in particular, the algorithm we use for the proof of Theorem 7.0.2. As input, the algorithm `TrigArc` takes an arc α which contains exactly one bad snippet in its inside. This bad snippet is required to be a trigon snippet. The output is an arc α' homotopic, relative its endpoints, to α such that α' does not contain any bad snippets in its inside. In other words, $\alpha'[1 : -1]$ is in efficient position. This is achieved by applying a series of local homotopies depending on the type of the initial trigon.

The outline of this chapter is as follows: First, we introduce the notion of a “turning direction” for certain snippets and define *horizontal* and *vertical duals*. Both concepts are important throughout the remainder of this thesis. Secondly, we define two further notions of length for arcs and curves, the *corner length* and the *reduced corner length*, and study how they are related to the snippet length of an arc or curve. Thirdly, we introduce a family of local homotopies and discuss the effects they have on arcs or curves with unique trigons in their insides. Here, we put a special emphasis on the various notions of length and their changes under the trigon homotopies. We then present the algorithm `TrigArc` and use our prior observations to give bounds on its running time and the length of its output arc. We close this chapter by defining and analysing the algorithm `TrigCurve`, which homotopes snippet-decomposed curves with a single bad snippet of trigon type into efficient position.

Throughout this chapter, we fix a surface $S = S_{g,b}$ satisfying $\xi(S) = 3g - 3 + b \geq 1$. We further fix a large train track $\tau \subset S$ and a tie neighbourhood $N = N(\tau)$

of τ in S . We recall that we do not distinguish between snippets and their strong snippet homotopy classes unless otherwise stated. For any snippet $a \subset R \in \mathcal{R}$, we always assume that a has minimal self-intersection, intersects ∂R perpendicularly and misses $\partial^2 \mathcal{R}$. We further recall that arcs and curves in S are assumed to be self-transverse and transverse to $\partial \mathcal{R}$. Moreover, if α is an arc, we assume that $\alpha(0)$ and $\alpha(1)$ lie in $\partial \mathcal{R}$. Thus, arcs and curves in S admit canonical decompositions into snippets.

3.1 Right/left turning snippets and horizontal/vertical duals

Recall that snippets are parametrized and are therefore canonically oriented. Hence, we can talk about regions cut off by the snippet on its right- or left-hand side.

Definition 3.1.1. Fix a region $R \in \mathcal{R} = \mathcal{R}_{\text{tie}} \cup \mathcal{R}_{\text{comp}}$. Suppose that $a \subset R$ is an embedded snippet that cuts off a region T of R of positive or non-negative index, where $R \in \mathcal{R}_{\text{tie}}$ or $R \in \mathcal{R}_{\text{comp}}$ respectively. We further suppose that the boundary of this region contains at least one point of $\partial^2 \mathcal{R}$. We say that the snippet a is *turning right* if it cuts off the region T on its right-hand side. Similarly, we say that a is *turning left* if it cuts off the region T on its left-hand side (see Figures 3.1-3.2).

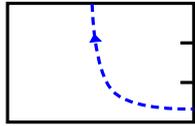


Figure 3.1: A right-turning trigon inside a switch rectangle.

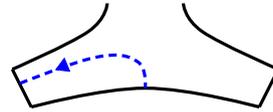


Figure 3.2: A left-turning trigon inside a complementary region.

It follows from the additivity of the index and the definition of a large train track that a snippet cannot be turning left and right at the same time.

Definition 3.1.2. Suppose that $a \subset R$ is a dual snippet inside a complementary region $R \in \mathcal{R}_{\text{comp}}$. Suppose further that a is embedded and cuts off a region of R of index zero. We say that a is a *horizontal dual* if $\partial a \subset \partial_v N$. Similarly, we say that a is a *vertical dual* if $\partial a \subset \partial_h N$ (see Figures 3.3-3.4).

Combining these two concepts, we obtain *right- or left-turning horizontal or vertical duals*. In the following, we simply call them *right or left horizontal or vertical duals*.

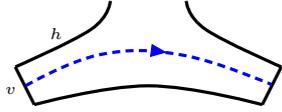


Figure 3.3: A right horizontal dual.

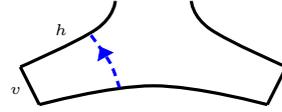


Figure 3.4: A left vertical dual.

3.2 Corner length and reduced corner length

Definition 3.2.1. Suppose that C is a component of $\partial_h N$. We set

$$s(C) = (|C \cap \partial^2 \mathcal{R}| - 2) \cdot 2 + 1.$$

In other words, $s(c)$ is a weighted count of the components of $C - \partial^2 \mathcal{R}$, where a component is attributed weight one or three if it is contained in the boundary of a branch or switch rectangle respectively. We further set

$$s = s_N = \max \{s(c) \mid C \text{ a component of } \partial_h N\},$$

and call s the *maximal side length of N* .

Remark 3.2.2. As S is compact, τ has a finite number of branches and switches. Therefore, the tie neighbourhood $N = N(\tau)$ is compact and $\partial_h N$ has a finite number of components. Hence, the maximal side length of N is well-defined. We remark that no horizontal side of $R \in \mathcal{R}_{\text{comp}}$ is homeomorphic to a circle. This follows from the fact that complementary regions of tie neighbourhoods of large train tracks must have at least two outwards pointing corners to satisfy the requirements on index. As branch and switch rectangles alternate in the tie neighbourhood and the first and last component of $C - \partial^2 \mathcal{R}$ belongs to the horizontal boundaries of branch rectangles, $s(C)$ is indeed a weighted count of the boundaries of branch and switch rectangles contained in C . We further remark that $s = O(|\chi(S)|)$ by [PH92, Corollary 1.1.3]. This will be of importance when we wish to obtain complexity bounds for our algorithms.

Lemma 3.2.3. *Suppose that $N = N(\tau)$ is a tie neighbourhood of a large train track $\tau \subset S$. Then $s_N \geq 5$.*

Proof. As train tracks are non-trivial by assumption, the tie neighbourhood N contains at least one switch rectangle. Each component of the horizontal boundary of this switch rectangle must be contained in a component C of $\partial_h R_C$ for some complementary region $R_C \in \mathcal{R}_{\text{comp}}$. As no horizontal boundary side is homeomorphic to a circle, branch and switch rectangles alternate in the tie neighbourhood and the

first and last component of $C - \partial^2 \mathcal{R}$ must be contained in the horizontal boundaries of branch rectangles, this implies that $|C - \partial^2 \mathcal{R}| \geq 4$. Hence, $s(C) \geq 5$ and the claim follows. \square

Definition 3.2.4. Suppose that $a \subset R$ is a snippet inside a complementary region $R \in \mathcal{R}_{\text{comp}}$. If a is non-peripheral, embedded, and cuts off a region T of R of non-negative index, then $\partial T = a \cup b$ for some set $b \subset (\partial \mathcal{R} - \partial S)$. By $|b|_v$ (respectively $|b|_B$ or $|b|_S$) we denote the number of components of $b - \partial^2 \mathcal{R}$ whose closure is a component of $\partial_v N$ (respectively of the horizontal boundary of a branch or switch rectangle of N). We set

$$\text{len}_{\text{corn}}(a) = |b|_v + |b|_B + 3 \cdot |b|_S.$$

If $a \subset R$ is peripheral, not embedded, or does not cut off a region of R of non-negative index, we set

$$\text{len}_{\text{corn}}(a) = 2s_N.$$

For any snippet $a \subset R \in \mathcal{R}_{\text{comp}}$, we call $\text{len}_{\text{corn}}(a)$ the *corner length of the snippet* a .

For some examples of snippets and their respective corner length we refer the reader to Figure 3.5.

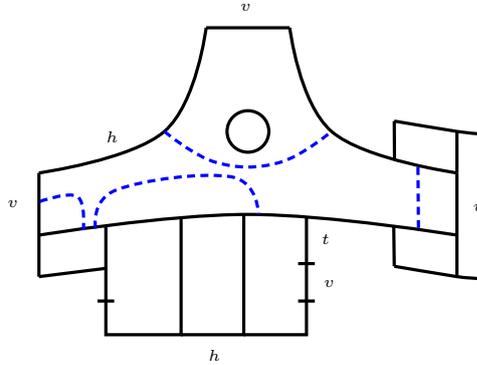


Figure 3.5: These snippets have corner length zero, one, four, and $2s_N$ respectively.

Remark 3.2.5. Following an index argument, any embedded snippet $a \subset R \in \mathcal{R}_{\text{comp}}$ cutting off a region T of R of non-negative index cuts off exactly one such region of R . If $\text{len}_{\text{corn}}(a) = 0$, then a must be a bigon snippet, trigon snippet, or an inessential curve. This stems from the fact that every simply connected region of non-positive index has at least four outward facing corners, of which at least two must belong to

$\partial^2\mathcal{R}$. Hence, its boundary contains at least one component of $\partial\mathcal{R} - \partial^2\mathcal{R}$. For any trigon or bigon snippet $a \subset R$, $\text{len}_{\text{corn}}(a)$ is bounded by $s_N - 1$. Suppose that $a \subset R$ is a horizontal dual. Then a is parallel to a horizontal boundary side C of $\partial_h R$ and $\text{len}_{\text{corn}}(a) = s(C) \leq s_N$. Similarly, the corner length of any vertical dual is bounded by $2s_N - 1$. Thus, $\text{len}_{\text{corn}}(a) \leq 2s_N$ for any snippet $a \subset R \in \mathcal{R}_{\text{comp}}$.

Definition 3.2.6. Suppose that $a \subset R$ is a snippet inside a branch or switch rectangle $R \in \mathcal{R}_{\text{tie}}$. We set $\text{len}_{\text{corn}}(a) = 1$ or $\text{len}_{\text{corn}}(a) = 3$ respectively and call $\text{len}_{\text{corn}}(a)$ the *corner length of the snippet* a . The *corner length* $\text{len}_{\text{corn}}(\alpha)$ of any snippet-decomposed arc or curve $\alpha \subset S$ is the sum over the corner lengths of all its snippets. In other words,

$$\text{len}_{\text{corn}}(\alpha) = \sum_{\substack{a \text{ snippet} \\ \text{of } \alpha}} \text{len}_{\text{corn}}(a).$$

Lemma 3.2.7. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve. Then*

$$\text{len}(\alpha) \leq \text{len}_{\text{corn}}(\alpha) + m_\alpha \leq 2s_N \cdot \text{len}(\alpha) + m_\alpha,$$

where m_α is the number of bad snippets of α .

Proof. Since $s_N \geq 4$, both bounds follow from the definition of corner length and Remark 3.2.5. \square

Throughout the various algorithms presented in this thesis, one type of subarc is occurring again and again. Its presence turns out to be a valuable indicator for the growth of the arc or curve under local homotopies.

Definition 3.2.8. Suppose that $\alpha \subset S$ is an arc in efficient position which contains exactly three snippets. Suppose further that $\alpha[0]$ and $\alpha[2]$ are both right or left vertical duals and $\alpha[1]$ is a dual snippet inside a branch rectangle. We then call α a *right or left blocker* respectively.

For an example of a blocker we refer the reader to Figure 3.6.

Lemma 3.2.9. *Suppose that $\alpha \subset S$ is a right or left blocker. Then $\text{len}_{\text{corn}}(\alpha) \geq 7$.*

Proof. As α is in efficient position, we know that $\text{len}_{\text{corn}}(\alpha[0])$ and $\text{len}_{\text{corn}}(\alpha[2])$ are greater than zero. We further know that $\text{len}_{\text{corn}}(\alpha[1]) = 1$ as $\alpha[1]$ is a snippet inside a branch rectangle. Let us assume that $\text{len}_{\text{corn}}(\alpha[0]) = 1$. Then $\alpha[2]$ cuts off a region of index zero whose boundary contains at least one component of the horizontal boundary of the switch rectangle that is adjacent to the branch rectangle containing

Proof. By Lemma 3.2.10, if every snippet of α is contained in at least one blocker, α consists of blockers only. These overlap along their vertical duals. As every blocker consists of three snippets, $\text{len}(\alpha)$ must be odd.

We prove the desired statement by induction on the snippet length of α . For $\text{len}(\alpha) = 3$, the claim follows from Lemma 3.2.9. Now suppose that $\text{len}(\alpha) > 3$. Then $\alpha[0 : 5]$ consists of two blockers, hence $\alpha[4 :]$ contains two fewer blockers than α . We remind ourselves that for any dual snippet $a \subset \alpha$ we know that $\text{len}_{\text{red}}(a) = \text{len}_{\text{corn}}(a) \geq \text{len}(a)$. As α must be in efficient position, induction and Lemma 3.2.9 give us

$$\begin{aligned} \text{len}_{\text{red}}(\alpha) &= \text{len}_{\text{corn}}(\alpha[0 : 4]) + \text{len}_{\text{red}}(\alpha[4 :]) - 4 \\ &\geq 8 + (\text{len}(\alpha[4 :])) - 4 \\ &\geq \text{len}(\alpha[: 4]) + 4 + \text{len}(\alpha[4 :]) - 4 \\ &\geq \text{len}(\alpha) \end{aligned}$$

and the claim follows. \square

Lemma 3.2.13. *Suppose that α is an arc. Then $\text{len}_{\text{red}}(\alpha) \geq \text{len}(\alpha) - m_\alpha$, where m_α denotes the number of bad snippets of α .*

Proof. We begin by splitting the arc α into maximal subarcs such that each subarc either consists of snippets that are all contained in at least one blocker or that no snippet is contained in any blocker. Then, the reduced corner length of α equals the sum of the reduced corner lengths of these subarcs. Furthermore, the reduced corner length of subarcs not containing any blockers is equal to their corner length. Lemma 3.2.7 implies that $\text{len}_{\text{corn}}(\alpha) \geq \text{len}(\alpha) - m_\alpha$. Hence, it remains to prove that the reduced corner length provides an upper bound for subarcs of the first type, which follows from Lemma 3.2.12. \square

3.3 Local homotopies

Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve. As seen in the previous chapter, arcs or curves consisting of more than one snippet may contain bad snippets of up to fourteen different types. They all have one characteristic in common: they cut off a region of positive index. In the following, we define a family of local homotopies that can be applied to (neighbourhoods of) such bad snippets. In the course of this thesis, we will see that those are sufficient to achieve efficient position for arcs and curves.

Definition 3.3.1. Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve consisting of at least two snippets, of which one, $a = \alpha[k]$, is a bad snippet. If α is an arc, we furthermore suppose that $0 < k < \text{len}(\alpha) - 1$. As a is a bad snippet meeting $\partial\mathcal{R} - \partial S$, it lies inside a region $R \in \mathcal{R}$ and cuts off a trigon or bigon. Let $N(T)$ be a small regular neighbourhood of T in S . We assume that α meets $\partial N(T)$ perpendicularly. Then a regular neighbourhood of $a \subset \alpha$, in the following called $N(a)$, divides $N(T)$ into two regions, exactly one of which, A , contains T . Replacing the subarc $N(a) \cap \partial A$ of α by the arc $\partial A - N(a)$ and smoothing out the resulting two corners yields a smooth arc or curve $\alpha' =: \text{Hom}(\alpha, k)$ homotopic to α . We refer to this as *applying a local homotopy to α at $a = \alpha[k]$* . If $T \subset R$ is a trigon or a bigon, we refer to this as *applying a local trigon or bigon homotopy to α at $\alpha[k]$* respectively.

For examples of local homotopies we refer the reader to Figures 3.11-3.19 and Figures 4.1-4.11.

Remark 3.3.2. Suppose that $\alpha[k]$ is a bad snippet of a snippet-decomposed arc or curve α . Applying a local homotopy to α at $\alpha[k]$ only alters a small neighbourhood of the snippet $\alpha[k]$ in α . Without loss of generality, we may always assume that this neighbourhood is contained in the subarc $\alpha[k - 1/3 : k + 4/3]$. For a reminder on the latter notation we refer the reader to page 31. If $\text{len}(\alpha) > 2$, this implies that the boundary of the subarc $\alpha[k - 1 : k + 2]$ is fixed, and any local homotopy affects at most these three snippets.

In Section 3.4, we study the effects of trigon homotopies on the number and types of snippets of the underlying arc or curve. We begin with a series of general observations and then draw our conclusions for each of the five trigon types separately.

3.4 Trigon homotopies

We recall that $\partial\mathcal{R}$ is a trivalent graph whose edges meet at angles $\pi/2$ or π .

Lemma 3.4.1. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) > 2$. Suppose further that α_{trim} contains a trigon snippet $\alpha[k] \subset R \in \mathcal{R}$. That is, $\alpha[k]$ cuts off a trigon $T \subset R$ and ∂T contains a unique corner x of R . Set $b = \partial T - \alpha[k]$. Let E be the set of (half-)edges of the trivalent graph $\partial^2\mathcal{R}$ that meet b but are not contained in b . Then the following statements hold.*

1. $b - x$ consists of two components b_1 and b_2 , where $b_1 \cap \partial^2\mathcal{R} = \emptyset$.
2. Either $|E| = 1$ or all (half-)edges of E meet b_2 at angle $\pi/2$.

$$3. \text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha) - 2 + |\partial T \cap \partial^2 \mathcal{R}|.$$

Proof. We prove the statements of the lemma in order. First, recall that horizontal sides of branch and switch rectangles and vertical sides of complementary regions do not contain any points of $\partial^2 \mathcal{R}$ in their interiors. As vertical and horizontal boundary sides of any region alternate, one of the components b_1 and b_2 of $b - x$ must have empty intersection with $\partial^2 \mathcal{R}$. In the following, we assume that $b_1 \cap \partial^2 \mathcal{R} = \emptyset$.

Secondly, if $|E| > 1$, that is, if b_2 has non-empty intersection with $\partial^2 \mathcal{R}$, b_2 must be a subarc of a vertical side of a switch rectangle or a subarc of a horizontal side of a complementary region. Set e to be the edge of E at the corner x of the trigon. If b_2 is a subarc of a vertical side of a switch rectangle, then e and b_1 are subarcs of $\partial_h N$ (see Figure 3.7). Thus, they meet at an angle of π at x . As b_1 and b_2 meet at an angle of $\pi/2$ at x , this implies that e and b_2 meet at an angle of $\pi/2$ at x as well. If b_2 is a subarc of a horizontal side of a complementary region, then e and b_2 are subarcs of a vertical boundary side of a switch rectangle (see Figure 3.8). Hence, they meet at an angle of π at x and e and b_2 must therefore meet at an angle of $\pi/2$ at x . As b_2 is a smooth subarc, all edges of $E - e$ must be perpendicular to b_2 , which proves the claim.

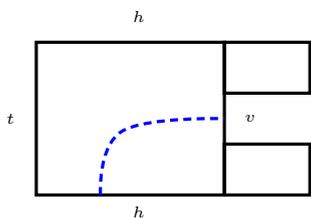


Figure 3.7: A trigon inside a switch rectangle whose boundary contains more than one point of $\partial^2 \mathcal{R}$.

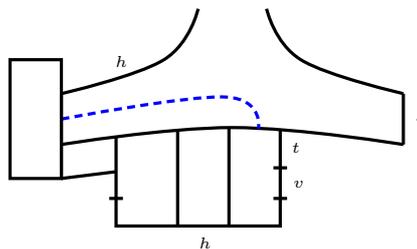


Figure 3.8: A trigon inside a complementary region whose boundary contains more than one point of $\partial^2 \mathcal{R}$.

Thirdly, let β be the image of the subarc $\alpha[k - 1/3, k + 4/3]$ under the local homotopy applied to $\alpha[k]$ (see Figures 3.9-3.10). For each point in $\partial T \cap \partial^2 \mathcal{R}$, β intersects an edge of $\partial \mathcal{R}$ once. Thus, the arc $\alpha[k - 1 : k + 2]$ is replaced by $\text{len}(\alpha[k - 1 : k + 2]) - 1 + |\partial T \cap \partial^2 \mathcal{R}| - 1$ many snippets. This implies that $\text{len}(\alpha') = \text{len}(\alpha) - 1 + |\partial T \cap \partial^2 \mathcal{R}| - 1 = \text{len}(\alpha) - 2 + |\partial T \cap \partial^2 \mathcal{R}|$. \square

Definition 3.4.2. Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose further that α_{trim} contains exactly one bad snippet $\alpha[k]$. If $\alpha[k]$ is a trigon or bigon snippet, we say that α is an *almost efficient arc or curve of trigon or bigon type* respectively.

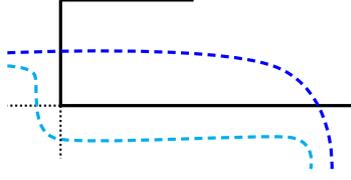


Figure 3.9: A trigon that contains a unique point of $\partial^2\mathcal{R}$.

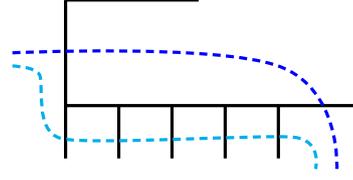


Figure 3.10: A trigon that contains more than one point of $\partial^2\mathcal{R}$.

Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. The following lemma shows that applying a local homotopy at the unique trigon snippet yields an arc or curve α' which is in efficient position in its inside or is again an almost efficient arc or curve of trigon type. In the latter case, the trigon snippet of α'_{trim} turns the same way as the trigon snippet of α_{trim} . In addition, the lemma determines the position of the trigon snippet of α' as well as gives restrictions on the types of snippets that the homotopy gives rise to.

Lemma 3.4.3. *Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. Set $0 \leq k \leq \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . We further set $\alpha' = \text{Hom}(\alpha, k)$ as well as $\beta = \alpha[k-1 : k+2]$ and $\beta' = \text{Hom}(\beta, 1)$. Then the following statements hold:*

1. *The subarc $\beta'[1 : -1]$ is in efficient position. If $\alpha[k]$ is turning right, then all snippets of $\beta[1 : -1]$ cut off a region of index zero on their left-hand side.*
2. *If $\text{len}(\alpha) > 2$, then either $\beta[0]$ is weakly snippet homotopic to $\beta'[0]$ or $\beta[-1]$ is weakly snippet homotopic to $\beta'[-1]$.*
3. *Suppose that $\alpha[k]$ is turning right and that $\beta[0]$ is embedded and cuts off a simply connected region on its right-hand side that is not a bigon. If $\beta[0]$ and $\beta'[0]$ are not weakly snippet homotopic, then the region cut off by $\beta'[0]$ on its right-hand side has one outward-pointing corner less than the region cut off by $\beta[0]$ on its right-hand side.*
4. *Suppose that $\beta[0]$ lies inside a peripheral complementary region $R \in \mathcal{R}_{\text{comp}}$. If $\beta[0]$ is not weakly snippet homotopic to $\beta'[0]$, then*

$$\text{wind}(\beta[0]) = \text{wind}(\beta'[0]) \pm 1.$$

Else, $\text{wind}(\beta[0]) = \text{wind}(\beta'[0])$.

5. *α'_{trim} contains at most one bad snippet, which must be a trigon turning into the same direction as $\alpha[k]$.*

6. Suppose that α'_{trim} is not in efficient position and that $\alpha[k]$ is turning right. If α'_{trim} contains a trigon snippet of type $\mathbb{R}(h, v)$, then α'_{trim} contains one right dual less than α_{trim} .

Proof. We prove the statements of the lemma in order. Without loss of generality, we assume that $\alpha[k] \subset R \in \mathcal{R}$ is a right-turning trigon snippet. By T we denote the trigon cut off by $\alpha[k]$. As discussed before, $b = \partial T - \alpha[k]$ contains a unique corner x of R and $b - x$ consists of two components b_1, b_2 , where we assume that $b_1 \cap \partial^2 R = \emptyset$. Let E be the set of (half-)edges of the trivalent graph $\partial^2 \mathcal{R}$ that meet b but are not contained in b .

First, we want to show that $\beta'[1 : -1]$ is in efficient position. If $|E| = 1$, then $\beta'[1 : -1]$ is empty. Else, its snippets are parallel to $b_2 \subset \partial R$ (see page 42, Figure 3.10). As $\alpha[k]$ is cutting off a trigon of R on its right-hand side, the snippets of $\beta[1 : -1]$ cut off rectangles on their left-hand sides.

Secondly, assume that $\text{len}(\alpha) > 2$. Thus, we have that $\alpha[k-1] \neq \alpha[k+1]$. Without loss of generality, we may assume that $\beta[0]$ is the snippet that intersects $\alpha[k]$ in $\overline{b_1}$. Let $e \in E$ be the edge of $\partial \mathcal{R}$ that is adjacent to the corner x . If $|E| > 1$, then e meets b_1 at an angle of π (see Figure 3.10). Thus, x is not a corner of the region containing the snippet $\beta[0]$ and $\beta[0]$ is weakly snippet homotopic to $\beta'[0]$. However, as e meets b_1 at an angle of $\pi/2$, $\beta[-1]$ and $\beta'[-1]$ are not weakly snippet homotopic. If $|E| = 1$, then either e meets b_1 at an angle of π or e meets b_2 at an angle of π . Hence, x is a corner of either the region containing $\beta[0]$ or the region containing $\beta[-1]$, which determines whether $\beta[0]$ and $\beta'[0]$, or $\beta[-1]$ and $\beta'[-1]$, are weakly snippet homotopic.

Thirdly, assume that $\beta[0]$ is embedded and cuts off a simply connected region on its right-hand side that is not a bigon. Suppose further that $\beta[0]$ and $\beta'[0]$ are not weakly snippet homotopic. Then $\beta'[0]$ cuts off a region on its right-hand side that has one outward-pointing corner less than the region cut off by $\beta[0]$, namely the corner x . Thus, if $\beta[0]$ is in efficient position, then $\beta'[0]$ is in efficient position or cuts off a trigon on its-right hand side.

Fourthly, assume that $\beta[0]$ lies inside a peripheral complementary region $R \in \mathcal{R}_{\text{com}}$. If $\beta[0]$ is not weakly snippet homotopic, then $\beta[0](1)$ is moved ‘‘across’’ the corner x under the homotopy. This implies that $\text{wind}(\beta[0]) = \text{wind}(\beta'[0]) \pm 1$. Else, that is if $\beta[0]$ and $\beta'[0]$ are weakly snippet homotopic, then their winding numbers coincide by definition of the winding number.

To prove the fifth claim we first assume that $\text{len}(\alpha) > 2$ and that $\beta[-1]$ and $\beta'[-1]$ are weakly snippet homotopic. Thus, if $\beta[0] \subset \alpha_{trim}$, we know that $\beta[0]$ is in efficient position, that $\beta'[0] \subset \alpha'_{trim}$ and that $\beta'[0]$ is in efficient position or cuts off

a trigon on its-right hand side. If $\beta[-1] \subset \alpha_{\text{trim}}$, then Lemma 2.5.15 implies that $\beta'[-1]$ is in efficient position as $\beta[-1]$ and $\beta'[-1]$ are weakly snippet homotopic and $\beta[-1]$ is in efficient position. Since all snippets of $\beta'[1 : -1]$ are in efficient position, this implies that α'_{trim} is in efficient position or contains exactly one bad snippet which must be a right-turning trigon snippet. If $\text{len}(\alpha) = 2$, then $\alpha[k-1]$ is in efficient position and $\alpha'[k-1]$ is in efficient position or a right-turning trigon snippet. Since all other snippets of α'_{trim} are parallel to $\partial\mathcal{R}$, they must be in efficient position. Thus, α' is in efficient position or contains a unique right-turning trigon snippet $\alpha'[k-1]$.

The sixth claim follows from statements 3 and 4 and the fact that each corner of a region contributes $-1/4$ towards the index of that region. So, if α'_{trim} contains a right-turning trigon snippet inside a complementary region, then $\alpha[k-1]$ or $\alpha[k+1]$ must lie inside a complementary region and cut off a right dual. \square

Remark 3.4.4. We note that we obtain the equivalent statements of Lemma 3.4.3 for left-turning trigons by replacing every occurrence of the word “right” by the word “left”.

Remark 3.4.5. We recall that any properly immersed arc or curve $\alpha \subset S$ is uniquely determined by its cutting sequence and the winding number of the respective snippets. For the remainder of this thesis we assume that arithmetic on winding numbers can be done in constant time. We claim that this implies that trigon homotopies can be computed in $O(|\chi(S)|)$ time: Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve that contains a snippet $\alpha[k]$ of trigon type in its inside. Let T be the trigon cut off by the snippet $\alpha[k]$. Lemma 3.4.1 implies that the snippets of the subarc $\alpha[k-1 : k+2]$ are replaced by $(|\partial T \cap \partial^2\mathcal{R}| + 1) \leq s$ many snippets. Remark 3.3.2 implies that all other snippets of α remain unchanged. Following the conventions agreed upon in Remark 2.7.1, this implies that the cutting sequence can be adjusted in $O(|\chi(S)|)$ time. Lemma 3.4.3 implies that the winding number of the first and last snippet of $\alpha[k-1 : k+2]$ change by at most one. Furthermore, if a snippet of $\text{Hom}(\alpha[k-1 : k+2], 1)[1 : -1]$ lies inside a peripheral annulus region, it is a horizontal or vertical dual so the modulus of its winding number equals two. As $\chi(S) \neq 0$ and $s = O(|\chi(S)|)$, this implies that the winding numbers of $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time. Thus, $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time.

For each of the five trigon types we are now going to draw our conclusion from the previous lemma, putting special emphasis on the occurring types of snippets and their length. We remark that this is a mere application of the results in Lemma

3.4.3. However, stating the results for the single trigon types separately help us to understand the implications that repeated applications of local homotopies have on an almost efficient arc or curve of trigon type. For the remainder of this thesis, we employ the following notation

- $\text{carr}(\alpha)$ denotes the number of carried snippets of α .
- $\text{dual}_R(\alpha)$ denotes the number of right duals of α .
- $\text{dual}_L(\alpha)$ denotes the number of left duals of α .

We note that all these quantities are bounded from above by $\text{len}(\alpha)$ and $\text{len}_{\text{corn}}(\alpha)$.

3.4.1 Trigons of type $\mathbb{B}(h, t)$

Lemma 3.4.6. *Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. Set $0 \leq k \leq \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . Set $\alpha' = \text{Hom}(\alpha, k)$. If $\alpha[k]$ is a trigon snippet of type $\mathbb{B}(h, t)$, then the following statements hold.*

1. $\text{len}(\alpha') = \text{len}(\alpha) - 1$.
2. α'_{trim} contains at most one bad snippet. This bad snippet is a trigon snippet inside a switch rectangle or complementary region and turns the same way as $\alpha[k]$.
3. If α'_{trim} contains a trigon snippet inside a switch rectangle, then $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}}) - 1$ and α'_{trim} contains the same number of right and left duals as α_{trim} .
4. If α'_{trim} contains a right-turning trigon of type $\mathbb{R}(h, v)$, then $\text{dual}_R(\alpha'_{\text{trim}}) = \text{dual}_R(\alpha_{\text{trim}}) - 1$ and $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}})$.
5. If α'_{trim} is not in efficient position, then $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}})$.
6. If α'_{trim} is in efficient position, then $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s$.

Proof. We prove the statements of the lemma in order. First, suppose that T is the trigon cut off by $\alpha[k]$. As $\alpha[k]$ is of type $\mathbb{B}(h, t)$, we know that $|\partial T \cap \partial^2 \mathcal{R}| = 1$. Hence, Lemma 3.4.1 implies that $\text{len}(\alpha') = \text{len}(\alpha) - 2 + 1 = \text{len}(\alpha) - 1$, giving 1.

For the remainder of this proof we assume that $\alpha[k-1]$ and $\alpha[k+1]$ lie inside a switch rectangle and complementary region respectively. Thus, $\alpha'[k-1]$ and $\alpha'[k]$ lie inside a switch rectangle and complementary region respectively. Hence, if one of

these is a trigon snippet, it must be of type $\mathbb{S}(h, t, 1)$, $\mathbb{S}(h, v, 2)$, $\mathbb{S}(h, t, 3)$ or $\mathbb{R}(h, v)$. Combining this analysis with Lemma 3.4.3 gives 2.

Thirdly, if α'_{trim} contains a bad snippet of type $\mathbb{S}(h, t, 1)$, $\mathbb{S}(h, v, 2)$, or $\mathbb{S}(h, t, 3)$, then this must be the snippet $\alpha'[k-1]$. Hence, $\alpha[k-1] \subset \alpha_{\text{trim}}$ and $\alpha[k-1]$ must be carried. As no other carried snippets of α are affected by the homotopy, we see that $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}}) - 1$. Since $\alpha[k+1]$ and $\alpha'[k]$ must be weakly snippet homotopic in this case, α'_{trim} contains the same number of right and left duals as α_{trim} , giving 3.

Fourthly, if α'_{trim} contains a right-turning trigon of type $\mathbb{R}(h, v)$, this must be the snippet $\alpha'[k]$. Hence, $\alpha[k+1]$ lies in α_{trim} and must be a right dual. As no duals of α but $\alpha[k+1]$ are affected by the homotopy, this implies that

$$\text{dual}_R(\alpha'_{\text{trim}}) = \text{dual}_R(\alpha_{\text{trim}}) - 1.$$

Since $\alpha[k-1]$ and $\alpha'[k-1]$ must be weakly snippet homotopic in this case, we know that $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}})$, giving 4.

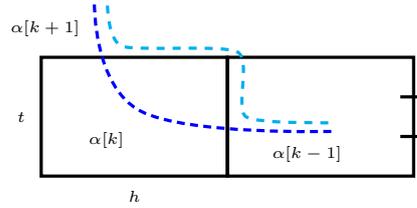


Figure 3.11: A homotopy of type $\mathbb{B}(h, t)$ applied to a trigon snippet that intersects the large end of a switch rectangle.

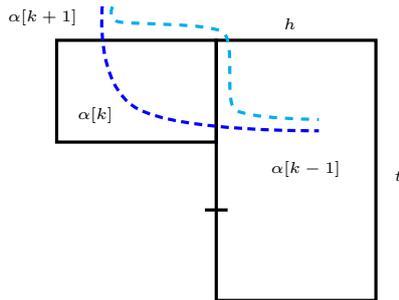


Figure 3.12: A homotopy of type $\mathbb{B}(h, t)$ applied to a trigon snippet that intersects a small end of a switch rectangle such that the trigon does not contain any corners of the adjacent complementary region in its boundary.

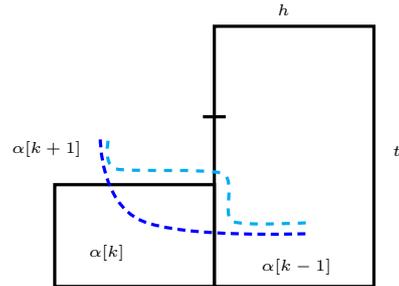


Figure 3.13: A homotopy of type $\mathbb{B}(h, t)$ applied to a trigon snippet that intersects a small end of a switch rectangle such that the trigon contains a corner of the adjacent complementary region in its boundary.

We prove the remaining two statements at the same time. Since $\alpha[k-1]$

and hence $\text{Hom}(\alpha[k-1 : k+2], 1)[0]$ lie inside a switch rectangle, we know that $\text{len}_{\text{corn}}(\alpha'[k-1]) = \text{len}_{\text{corn}}(\alpha[k-1])$. We further know that $\text{len}_{\text{corn}}(\alpha[k]) = 1$.

If $\alpha[k+1] \not\subset \alpha_{\text{trim}}$, this implies that $\text{len}_{\text{corn}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{corn}}(\alpha_{\text{trim}}) - 1$. As no blockers of α_{trim} are affected by the homotopy in this case, we see that $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) - 1$.

For the remainder of this proof we assume that $\alpha[k+1] \subset \alpha_{\text{trim}}$. We have to distinguish two cases: either $\alpha[k+1]$ is weakly snippet homotopic to $\alpha'[k]$ or it is not. If $\alpha[k+1]$ is weakly snippet homotopic to $\alpha'[k]$, then either

$$\text{len}_{\text{corn}}(\alpha'[k]) = \text{len}_{\text{corn}}(\alpha[k+1]) = 2s$$

or

$$\text{len}_{\text{corn}}(\alpha'[k]) \leq \text{len}_{\text{corn}}(\alpha[k+1]) + 1.$$

The last inequality follows from the fact that the boundary of the region cut off by $\alpha'[k]$ contains at most one horizontal side of a branch rectangle more than the boundary of the region cut off by $\alpha[k+1]$. Since $\alpha[k+1]$ is weakly snippet homotopic to $\alpha'[k]$, the number of blockers of α_{trim} and α'_{trim} coincides, so $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}})$.

If $\alpha[k+1]$ is not weakly snippet homotopic to $\alpha'[k]$, then we have to distinguish the following two cases: either $\text{len}_{\text{corn}}(\alpha'[k]) = 2s$ or $\text{len}_{\text{corn}}(\alpha'[k]) < 2s$. In the first case we know that α'_{trim} must be in efficient position. As $\alpha[k+1]$ is in efficient position, we further know that $\text{len}_{\text{corn}}(\alpha[k+1]) \geq 1$. This implies that

$$\text{len}_{\text{corn}}(\alpha'[k]) \leq \text{len}_{\text{corn}}(\alpha[k+1]) + 2s - 1.$$

Thus, we see that $\text{len}_{\text{corn}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{corn}}(\alpha_{\text{trim}}) + 2s - 2$. As the homotopy decreases the number of blockers by at most one, this implies that

$$\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s.$$

In the second case, that is if $\text{len}_{\text{corn}}(\alpha'[k]) < 2s$, an index-argument shows that $\text{len}_{\text{corn}}(\alpha[k+1]) = 2s$. Thus, we see that $\text{len}_{\text{corn}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{corn}}(\alpha_{\text{trim}}) - 2$. As the homotopy decreases the number of blockers again by at most one, this implies that $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}})$, which concludes the proof of the lemma. \square

3.4.2 Trigons of type $\mathbb{S}(h, t, 1)$

Lemma 3.4.7. *Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. Set $0 \leq k \leq \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . Set*

$\alpha' = \text{Hom}(\alpha, k)$. If $\alpha[k]$ is a trigon snippet of type $\mathbb{S}(h, t, 1)$, then the following statements hold.

1. $\text{len}(\alpha') = \text{len}(\alpha) - 1$.
2. α'_{trim} contains at most one bad snippet. This snippet is of type $\mathbb{B}(h, t)$ and turns the same way as $\alpha[k]$.
3. If α'_{trim} is not in efficient position, then $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}}) - 1$ and α'_{trim} contains the same number of right and left duals as α_{trim} .
4. $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}})$.

Proof. We prove the statements of the lemma in order. First, suppose that T is the trigon cut off by $\alpha[k]$. As $\alpha[k]$ is of type $\mathbb{S}(h, t, 1)$, we know that $|\partial T \cap \partial^2 \mathcal{R}| = 1$. Hence, Lemma 3.4.1 implies that $\text{len}(\alpha') = \text{len}(\alpha) - 2 + 1 = \text{len}(\alpha) - 1$, giving 1.

For the remainder of this proof we assume that $\alpha[k-1]$ lies inside a branch rectangle. Denote by $R \in \mathcal{R}_{\text{com}}$ the complementary region that contains the snippet $\alpha[k+1]$. As $\alpha[k]$ is of type $\mathbb{S}(h, t, 1)$, the corner of $\partial^2 \mathcal{R}$ that lies in the boundary of the trigon T is not a corner of the complementary region R (see Figures 3.14-3.15). Thus, $\alpha[k+1]$ and $\alpha'[k]$ are weakly snippet homotopic. Lemma 3.4.3 then implies that if α'_{trim} contains a bad snippet, this must be the snippet $\alpha'[k-1]$, which lies inside a branch rectangle. This gives 2.

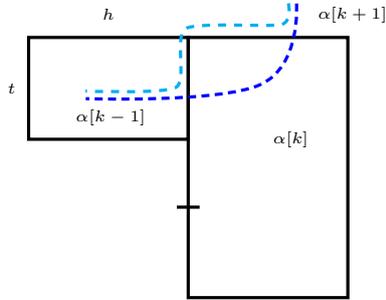


Figure 3.14: A homotopy of type $\mathbb{S}(h, t, 1)$ applied to a trigon snippet that intersects a small end of a switch rectangle.

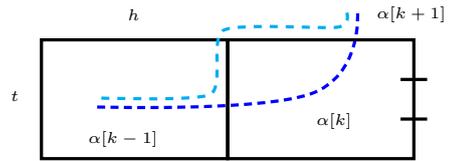


Figure 3.15: A homotopy of type $\mathbb{S}(h, t, 1)$ applied to a trigon snippet that intersects the large end of a switch rectangle.

If $\alpha'[k-1] \subset \alpha'_{\text{trim}}$, then we know that $\alpha[k-1]$ lies in α_{trim} and must therefore be carried. As no other carried snippets of α_{trim} are affected by the homotopy, we see that $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}}) - 1$. Furthermore, since $\alpha[k+1]$ and $\alpha'[k]$ are weakly snippet homotopic and no other duals of α_{trim} are affected by the homotopy, α'_{trim} contains the same number of right and left duals as α_{trim} , giving 3.

Lastly, we note that $\text{len}_{\text{corn}}(\alpha[k-1]) = \text{len}_{\text{corn}}(\alpha'[k-1])$ since both snippets lie inside a branch rectangle. As $\alpha[k]$ lies inside a switch rectangle, we know that $\text{len}_{\text{corn}}(\alpha[k]) = 3$. Unless $\alpha[k+1]$ and $\alpha'[k]$ both have corner length $2s$, the boundary of the region cut off by $\alpha'[k]$ of R either contains one side of a branch rectangle less or one side of a switch rectangle more than the boundary of the region cut off by $\alpha[k+1]$ of R . Thus, we see that $\text{len}_{\text{corn}}(\alpha'[k]) \leq \text{len}_{\text{corn}}(\alpha[k+1]) + 3$, which implies that $\text{len}_{\text{corn}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{corn}})$. Since $\text{len}_{\text{block}}(\alpha'_{\text{trim}}) = \text{len}_{\text{block}}(\alpha_{\text{trim}})$, statement 4 follows. \square

3.4.3 Trigons of type $\mathbb{S}(h, v, 2)$

Lemma 3.4.8. *Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. Set $0 \leq k \leq \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . Set $\alpha' = \text{Hom}(\alpha, k)$. If $\alpha[k]$ is a trigon snippet of type $\mathbb{S}(h, v, 2)$, then the following statements hold.*

1. $\text{len}(\alpha') = \text{len}(\alpha)$.
2. α'_{trim} contains at most one bad snippet. This snippet is of type $\mathbb{R}(h, v)$ and turns the same way as $\alpha[k]$.
3. α'_{trim} contains a right-turning trigon of type $\mathbb{R}(h, v)$ if and only if

$$\text{dual}_R(\alpha'_{\text{trim}}) = \text{dual}_R(\alpha_{\text{trim}}) - 1.$$

4. If α'_{trim} is not in efficient position, then $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}})$ and $\text{carr}(\alpha'_{\text{trim}}) = \text{carr}(\alpha_{\text{trim}})$.
5. If α'_{trim} is in efficient position, then $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s$.

Proof. We prove the statements of the lemma in order. First, suppose that T is the trigon cut off by $\alpha[k]$. As $\alpha[k]$ is of type $\mathbb{S}(h, v, 2)$, we know that $|\partial T \cap \partial^2 \mathcal{R}| = 2$. Hence, Lemma 3.4.1 implies that $\text{len}(\alpha') = \text{len}(\alpha) - 2 + 2 = \text{len}(\alpha)$, giving 1.

For the remainder of this proof we assume that $\alpha[k-1]$ lies inside a complementary region $R \in \mathcal{R}_{\text{com}}$ and intersects $\alpha[k]$ in $\partial_v N$ (see Figure 3.16). Let $R' \in \mathcal{R}_{\text{com}}$ be the complementary region that contains the snippet $\alpha[k+1]$. We note that it is possible that $R = R'$. This holds especially true if $\text{len}(\alpha) = 2$. Since $\alpha[k]$ is of type $\mathbb{S}(h, v, 2)$, the boundary of T contains two corners of $\partial^2 \mathcal{R}$. One of them is neither a corner of R nor R' , whereas the other is a corner of R . Thus, if α'_{trim} contains a bad snippet, this is the snippet $\alpha'[k-1]$. Since $\alpha'[k-1]$ lies inside a complementary

Suppose that $\alpha[k]$ turns right. We recall that $\alpha[k+1]$ and $\alpha'[k+2]$ are weakly snippet homotopic and that $\alpha'[k]$ must be a left vertical dual by Lemma 3.4.3. As no further duals of α_{trim} are created or affected by the homotopy, α'_{trim} contains the same number of right duals as α_{trim} , giving 4.

Lastly, we note that $\text{len}_{\text{corn}}(\alpha[k-1]) = \text{len}_{\text{corn}}(\alpha'[k-1])$ since both snippets lie inside a branch rectangle. As $\alpha[k]$ lies inside a switch rectangle, we know that $\text{len}_{\text{corn}}(\alpha[k]) = 3$. We further know that $\text{len}_{\text{corn}}(\alpha'[k:k+2]) = 2$. Unless $\alpha[k+1]$ and $\alpha'[k]$ both have corner length $2s$, the boundary of the region cut off by $\alpha'[k]$ of R either contains one side of a branch rectangle less or one side of a switch rectangle more than the boundary of the region cut off by $\alpha[k+1]$ of R . Thus, we see that $\text{len}_{\text{corn}}(\alpha'[k+2]) \leq \text{len}_{\text{corn}}(\alpha[k+1]) + 3$. This implies that

$$\text{len}_{\text{corn}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{corn}}) + 2.$$

However, we note that $\text{len}_{\text{corn}}(\alpha'[k+2]) = \text{len}_{\text{corn}}(\alpha[k+1]) + 3$ if and only if $\alpha'[k+2]$ is a left vertical dual. In this case, α'_{trim} contains one blocker more than α_{trim} , which implies that $\text{len}_{\text{block}}(\alpha'_{\text{trim}}) = \text{len}_{\text{block}}(\alpha_{\text{trim}})$. Else, we know that $\text{len}_{\text{corn}}(\alpha'[k]) \leq \text{len}_{\text{corn}}(\alpha[k+1])$. As the homotopy does not decrease the number of blockers, this implies that $\text{len}_{\text{block}}(\alpha'_{\text{trim}}) = \text{len}_{\text{block}}(\alpha_{\text{trim}})$, giving 5. \square

3.4.5 Trignons of type $\mathbb{R}(h, v)$

Lemma 3.4.10. *Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. Set $0 \leq k \leq \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . Set $\alpha' = \text{Hom}(\alpha, k)$. If $\alpha[k]$ is a trigon snippet of type $\mathbb{R}(h, v)$, then the following statements hold.*

1. $\text{len}(\alpha') \leq \text{len}(\alpha) + s_N - 2$.
2. α'_{trim} contains at most one bad snippet. This snippet is of type $\mathbb{B}(h, t)$, $\mathbb{S}(h, t, 1)$, or $\mathbb{S}(h, t, 3)$ and turns the same way as $\alpha[k]$.
3. If α'_{trim} is not in efficient position, then $\text{carr}(\alpha'_{\text{trim}}) \leq \text{carr}(\alpha_{\text{trim}}) + s_N - 1$.
4. The numbers of right and left duals of α_{trim} and α'_{trim} coincide.
5. $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}})$.

Proof. We prove the statements of the lemma in order. First, suppose that T is the trigon cut off by $\alpha[k]$. As $\alpha[k]$ is of type $\mathbb{R}(h, v)$, we know that $|\partial T \cap \partial^2 \mathcal{R}| \leq s$. Hence, Lemma 3.4.1 implies that $\text{len}(\alpha') \leq \text{len}(\alpha) - 2 + s$, giving 1.

For the remainder of this proof we assume that $\alpha[k-1]$ lies inside a switch rectangle $R \in \mathcal{R}_{\text{tie}}$ and intersects $\alpha[k]$ in $\partial_v N$ (see Figure 3.19). We further assume that $\alpha[k]$ turns right. We remark that $\text{len}(\alpha) > 2$ as $\alpha[k-1]$ is in efficient position. Set $\beta = \alpha[k-1 : k+2]$ and $\beta' = \text{Hom}(\beta, 1)$. As the corner of T that is also a point of $\partial^2 \mathcal{R}$ is no outward-pointing corner of R we see that $\beta[0]$ and $\beta'[0]$ are weakly snippet homotopic. Lemma 3.4.1 implies the boundary of the region cut off by $\beta[-1]$ on its right-hand side contains exactly one more point of $\partial^2 \mathcal{R}$ than the boundary of the region cut off by $\beta'[-1]$ on its right-hand side. Since $\beta'[-1]$ lies inside a branch or switch rectangle, Lemma 3.4.3 therefore implies that $\beta'[-1]$ can only be of type $\mathbb{B}(h, t)$, $\mathbb{S}(h, t, 1)$, or $\mathbb{S}(h, t, 3)$ and if so, turns the same way as $\alpha[k]$, giving 2.

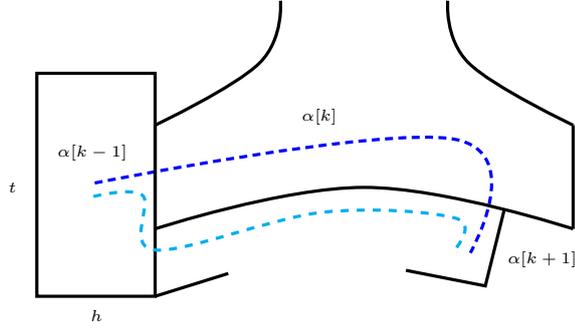


Figure 3.19: A homotopy of type $\mathbb{R}(h, v)$.

If $\beta[0] \subset \alpha_{\text{trim}}$, then $\beta[0]$ as well as $\beta'[0]$ are carried. Lemma 3.4.3 implies that all snippets of $\beta'[1 : -1]$ are carried. As $\text{len}(\beta'[1 : -1]) \leq s - 1$ and $\beta'[-1]$ is not carried if it lies in α'_{trim} , this implies that $\text{carr}(\alpha'_{\text{trim}}) \leq \text{carr}(\alpha_{\text{trim}}) + s_N - 1$, giving 3.

Since no duals of α are affected by or created under the homotopy, statement 4 follows.

Lastly, we note that $\text{len}_{\text{corn}}(\beta[0]) = \text{len}_{\text{corn}}(\beta'[0])$ since both snippets lie inside a switch rectangle. By definition of the corner length of the snippet $\beta[1]$ we know that $\text{len}_{\text{corn}}(\beta[1]) = \text{len}_{\text{corn}}(\beta'[1 : -1])$. We further know that

$$\text{len}_{\text{corn}}(\beta[-1]) = \text{len}_{\text{corn}}(\beta'[-1])$$

and that no duals of α are affected by or created under the homotopy, so statement 5 follows. \square

3.4.6 Trigon homotopies - a summary

We can summarise our findings of trigon homotopies in the following lemma:

Lemma 3.4.11. *Suppose that $\alpha \subset S$ is an almost efficient arc or curve of trigon type. Set $0 \leq k \leq \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . Set $\alpha' = \text{Hom}(\alpha, k)$. If $\alpha[k]$ turns right, then the following statements hold.*

1. α'_{trim} contains at most one bad snippet. This is a right-turning trigon.
2. If α'_{trim} contains a bad snippet, then $\text{len}_{red}(\alpha'_{trim}) \leq \text{len}_{red}(\alpha_{trim})$ and one of the following statements holds.
 - $\text{dual}_R(\alpha_{trim}) = \text{dual}_R(\alpha'_{trim})$ and $\text{carr}(\alpha'_{trim}) < \text{carr}(\alpha_{trim})$.
 - $\text{dual}_R(\alpha'_{trim}) < \text{dual}_R(\alpha_{trim})$ and $\text{carr}(\alpha'_{trim}) \leq \text{carr}(\alpha_{trim})$.
 - $\alpha[k]$ is a trigon of type $\mathbb{R}(h, v)$. This implies that $\text{dual}_R(\alpha_{trim}) = \text{dual}_R(\alpha'_{trim})$ and $\text{carr}(\alpha'_{trim}) \leq \text{carr}(\alpha_{trim}) + s - 1$.
3. If α'_{trim} contains a trigon of type $\mathbb{R}(h, v)$, then $\text{dual}_R(\alpha'_{trim}) < \text{dual}_R(\alpha_{trim})$.
4. If α'_{trim} is in efficient position, then $\text{len}_{red}(\alpha'_{trim}) \leq \text{len}_{red}(\alpha_{trim}) + 2s$.

Proof. This is an immediate consequence of the Lemmas 3.4.6, 3.4.7, 3.4.8, 3.4.9 and 3.4.10. □

Remark 3.4.12. We note that the statement of Lemma 3.4.11 also holds for left-turning trigons if we replace all occurrences of right-turning trigons and duals by left turning trigons and duals.

To make the content of Lemmas 3.4.6, 3.4.7, 3.4.8, 3.4.9, 3.4.10 and 3.4.11 more accessible, the directed graph in Figure 3.20 visualizes the connections between the different types of trigons. We note that we abbreviated $\text{carr}(\alpha_{trim})$ by c , $\text{carr}(\alpha'_{trim})$ by c' , $\text{dual}_R(\alpha_{trim})$ by d_R , and $\text{dual}_R(\alpha'_{trim})$ by d'_R . This graph displays the changes in the number of duals of α_{trim} in the case of a right-turning trigon snippet. Next to its five vertices corresponding to the five different types of trigon snippets, there should be a vertex corresponding to arcs that are in efficient position. For clarity, this has been omitted. Directed edges between a source and a target vertex indicate that under a trigon homotopy, trigon snippets of the source type might turn into trigon snippets of the target type. Situations in which neither the number of carried snippets nor the number of right duals decreases have been highlighted in red. We notice that these only occur for trigon snippets of type $\mathbb{R}(h, v)$ and that one only reaches the vertex labelled $\mathbb{R}(h, v)$ if one travels along an edge that indicates a decrease in right duals as highlighted in the second to last point of Lemma 3.4.11.

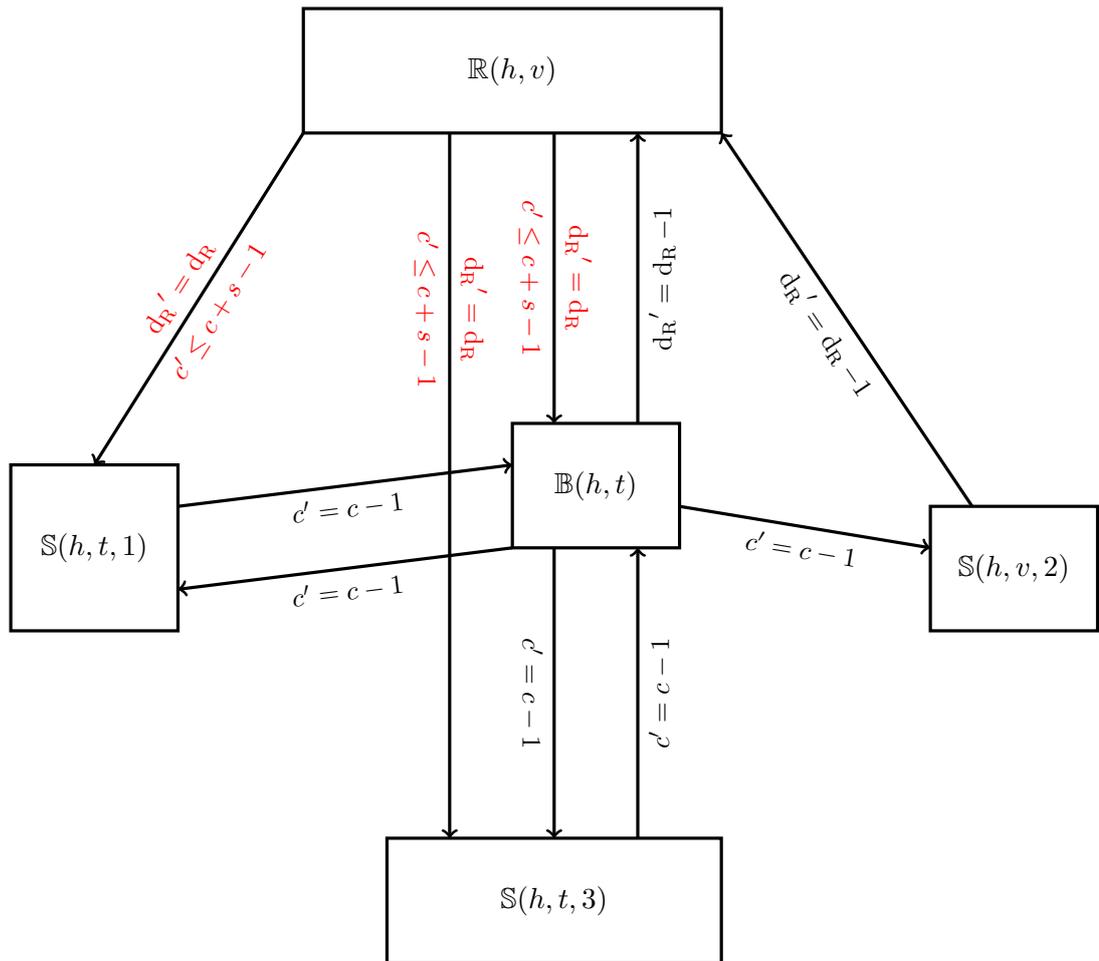


Figure 3.20: The trigon homotopy graph for right-turning trigons.

3.5 The algorithm TrigArc

Building on the previous sections, we now present the algorithm `TrigArc`. This algorithm

- takes as input a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an arc $\alpha \subset S$ which is almost efficient of trigon type or does not contain any bad snippets in its inside, and
- outputs an arc α' that is homotopic to α and does not contain any bad snippets in $\alpha'[1 : -1]$.

Lemma 3.4.11 shows that applying one trigon homotopy might not be sufficient to achieve efficient position for almost efficient arcs. The goal of this section is to prove that the process of repeatedly applying trigon homotopies terminates and yields an arc that is in efficient position in its inside. The formal statement of `TrigArc` is given in Algorithm 3.5.1.

Algorithm 3.5.1: `TrigArc` - Homotoping the inside of arcs with at most one bad snippet into efficient position, where the bad snippet is of trigon type.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an arc $\alpha \subset S$ which is almost efficient of trigon type or does not contain any bad snippets in its inside.

Output: An arc α' homotopic, relative its endpoints, to α such that α'_{trim} is in efficient position.

```

1  $\alpha' = \alpha$ 
2 while there is some  $k$  satisfying  $0 < k < \text{len}(\alpha') - 1$  such that  $\alpha'[k]$  is a
   bad snippet do
3   |  $\alpha' = \text{Hom}(\alpha', k)$ 
4 return  $\alpha'$ 

```

Lemma 3.5.1. *The algorithm `TrigArc` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2(\text{len}(\alpha_{trim}) + 1))$ time. For $\alpha' = \text{TrigArc}(S, N, \alpha)$, we have that*

$$\text{len}_{red}(\alpha'_{trim}) \leq \text{len}_{red}(\alpha_{trim}) + 2s.$$

Proof. By assumption on the input of the algorithm `TrigArc`, we are given an arc α whose inside either is in efficient position and, therefore, meets the requirements on the output of `TrigArc`, or contains a unique bad snippet $\alpha[k]$. In the latter case

we further have that $\alpha[k]$ is a trigon snippet. Without loss of generality, we may assume that $\alpha[k]$ is turning right. By Lemma 3.4.11, applying a local homotopy to α at $\alpha[k]$ yields an arc whose inside contains at most one bad snippet of trigon type that is again turning right. By induction, repeatedly applying a local homotopy to the unique right-turning trigon snippet of the arc yields again an arc with at most one right-turning trigon snippet in its inside. We claim that after having applied at most $(\text{len}(\alpha_{\text{trim}}) + 1) \cdot (s + 2)$ many homotopies, the inside of the arc will be in efficient position.

This can be seen as follows: by Lemma 3.4.11, one of the following two statements is true when applying a local trigon homotopy to an arc α :

- the number of carried snippets or right duals of α_{trim} decreases.
- the trigon homotopy is of type $\mathbb{R}(h, v)$, the number of carried snippets of α_{trim} increases by at most $s - 1$, and the number of right duals of α_{trim} does not change.

Moreover, Lemma 3.4.11 implies that α_{trim} contains a trigon of type $\mathbb{R}(h, v)$ only if α is the original input arc or α is the image under a local homotopy of an arc whose inside has one right dual more than α_{trim} . Thus, there can be at most $\text{dual}_R(\alpha_{\text{trim}}) + 1$ many homotopies of type $\mathbb{R}(h, v)$. Therefore, under the process of applying trigon homotopies repeatedly, the total number of carried snippets is bounded by

$$\begin{aligned} & \text{carr}(\alpha_{\text{trim}}) + (\text{dual}_R(\alpha_{\text{trim}}) + 1) \cdot (s - 1) \\ & \leq \text{len}(\alpha_{\text{trim}}) + (\text{len}(\alpha_{\text{trim}}) + 1) \cdot (s - 1) \\ & \leq (\text{len}(\alpha_{\text{trim}}) + 1) \cdot s, \end{aligned}$$

and any local homotopy which is not of type $\mathbb{R}(h, v)$ reduces this number or the number of right duals by at least one.

In summary, if we take into account one final homotopy which might be required once there are no carried snippets or right duals left, the total number of local homotopies that can be applied to an arc α before efficient position in the inside must be achieved is bounded by

$$\begin{aligned} & (\text{dual}_R(\alpha_{\text{trim}}) + 1) + \text{dual}_R(\alpha_{\text{trim}}) + (\text{len}(\alpha_{\text{trim}}) + 1) \cdot s + 1 \\ & \leq (\text{len}(\alpha_{\text{trim}}) + 1) + \text{len}(\alpha_{\text{trim}}) + (\text{len}(\alpha_{\text{trim}}) + 1) \cdot s + 1 \\ & \leq (\text{len}(\alpha_{\text{trim}}) + 1) \cdot (s + 2). \end{aligned}$$

Hence, the algorithm exits the while-loop after at most $(\text{len}(\alpha_{\text{trim}}) + 1) \cdot (s + 2)$ many iterations and the output arc α' must be in efficient position in its inside by Lemma 3.4.11. This implies correctness of the algorithm **TrigArc**.

We now analyse the running time of the algorithm **TrigArc** by going through its pseudocode line by line. If we want to determine whether there is indeed a bad snippet in the inside of the arc α in the very first iteration of the while-loop, we potentially have to check each snippet of α_{trim} and determine if it is in efficient position. This takes $O(|\chi(S)| \text{len}(\alpha_{\text{trim}}))$ time. However, in further iterations of the while-loop, we only have to check at most s many snippets. This follows from the fact that all snippets but the subarc $\alpha[k - 1 : k + 2]$ remain unchanged when applying a local homotopy to α at $\alpha[k]$ and that $\text{len}(\text{Hom}(\alpha'[k - 1 : k + 2], k)) \leq s$ by Lemma 3.4.11. Thus, for all but one iteration of the while-loop, checking the while-condition takes $O(|\chi(S)|)$ time. Following Remark 3.4.5, $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time. Therefore, the homotopy in an iteration of the while-loop can be computed in $O(|\chi(S)|)$ time as well. Thus, the algorithm **TrigArc** executes at most one iteration of the while-loop in time $O(|\chi(S)| \text{len}(\alpha_{\text{trim}}))$ and at most $(\text{len}(\alpha_{\text{trim}}) + 1) \cdot (s + 1)$ many iterations of the while loop in $O(|\chi(S)|)$ time. Hence, the algorithm **TrigArc** terminates in $O(\chi(S)^2(\text{len}(\alpha_{\text{trim}}) + 1))$ time.

By Lemma 3.4.11, the reduced corner length of the arc does not increase under a local trigon homotopy unless one obtains efficient position for the inside of the arc. For $\alpha' = \text{TrigArc}(S, N, \alpha)$, this implies that $\text{len}_{\text{red}}(\alpha'_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s$. \square

Corollary 3.5.2. *For any input (S, N, α) , the algorithm **TrigArc** halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha_{\text{trim}}) + 1))$ time.*

Proof. This follows from Lemma 3.5.1 and 3.2.13 if we keep in mind that α_{trim} contains at most one bad snippet, so $\text{len}(\alpha_{\text{trim}}) + 1 \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2$. Since $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2)) = O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha_{\text{trim}}) + 1))$, the desired bound follows. \square

Suppose that S is a surface of positive complexity and $N \subset S$ is a tie neighbourhood of a large train track in S . The following two corollaries will be of use later on:

Corollary 3.5.3. *Suppose that $\alpha \subset S$ is an almost efficient arc of trigon type. Set $0 < k < \text{len}(\alpha) - 1$ such that $\alpha[k]$ is the unique bad snippet of α_{trim} . If $\text{Hom}(\alpha, k)[0]$ or $\text{Hom}(\alpha, k)[-1]$ is not weakly snippet homotopic to $\alpha[0]$ or $\alpha[-1]$ respectively, then $\text{Hom}(\alpha, k)_{\text{trim}}$ is in efficient position. Hence, the algorithm **TrigArc** terminates immediately once the weak snippet homotopy type of the first or last snippet of the given arc has changed.*

Proof. We recall that any local homotopy applied to $\alpha[k]$ affects the subarc $\alpha[k-1 : k+2]$ only. By Lemma 3.4.3 we further know that the homotopy alters the weak snippet homotopy type of either $\alpha[k-1]$ or $\alpha[k+1]$ and that all snippets in $\text{Hom}(\alpha[k-1 : k+2], 1)[1 : -1]$ are in efficient position. Hence, if $\text{Hom}(\alpha, k)[0]$ is not weakly snippet homotopic to $\alpha[0]$, we know that $k = 1$ and that $\text{Hom}(\alpha, 1)[1 : -1]$ must be in efficient position. If $\text{Hom}(\alpha, k)[-1]$ is not weakly snippet homotopic to $\alpha[-1]$, we know that $k = \text{len}(\alpha) - 2$ and that again $\text{Hom}(\alpha, 1)[1 : -1]$ must be in efficient position. Thus, we see that $\text{Hom}(\alpha, k)_{\text{trim}}$ is in efficient position in both cases. As the algorithm **TrigArc** terminates as soon as the inside of the underlying arc is in efficient position, the second claim of the lemma follows. \square

Corollary 3.5.4. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc. Suppose further that $\text{len}(\alpha) > 2$, that $\alpha[0]$ is a left vertical dual and, that $\alpha[1]$ and $\alpha[-1]$ are right-turning trigon snippets of type $\mathbb{B}(h, t)$. Then $\alpha' = \text{TrigArc}(S, N, \alpha)$ contains at most one bad snippet, $\alpha'[-1]$, which is a snippet of type $\mathbb{B}(h, t)$ or $\mathbb{B}(h, h)$. If $\alpha'[-1]$ is a bigon snippet, then $\alpha'[0]$ and $\alpha'[-2]$ are left vertical duals.*

Proof. For clarity, we set $\text{TrigArc}(\alpha) := \text{TrigArc}(S, N, \alpha)$ for the remainder of this proof. By Corollary 3.5.3 we know that either both or exactly one of the pairs $\{\text{TrigArc}(\alpha)[0], \alpha[0]\}$ and $\{\text{TrigArc}(\alpha)[-1], \alpha[-1]\}$ are weakly snippet homotopic.

If both are weakly snippet homotopic, then $\text{TrigArc}(\alpha)[-1]$ is in efficient position and $\text{TrigArc}(\alpha)[-1] = \alpha[-1]$ is a right-turning trigon snippet inside a branch rectangle.

If $\text{TrigArc}(\alpha)[0]$ and $\alpha[0]$ are not weakly snippet homotopic, then $\text{TrigArc}(\alpha)[0]$ cuts off a region of index $-1/4$ on its left-hand side by Lemma 3.4.3. Thus, it is in efficient position following an index-argument. As $\text{TrigArc}(\alpha)[-1]$ and $\alpha[-1]$ are weakly snippet homotopic in this case, $\text{TrigArc}(\alpha)$ contains one bad snippet, which is the trigon snippet $\text{TrigArc}(\alpha)[-1] = \alpha[-1]$.

If $\text{TrigArc}(\alpha)[0]$ and $\alpha[0]$ are weakly snippet homotopic but $\text{TrigArc}(\alpha)[-1]$ and $\alpha[-1]$ are not, then $\text{TrigArc}(\alpha)[-1]$ is in efficient position and $\text{TrigArc}(\alpha)[-1]$ is a bigon snippet of type $\mathbb{B}(h, h)$. The last claim follows from the fact that $\alpha[-1]$ is embedded and cuts off a trigon on its right-hand side. Thus, Lemma 3.4.3 implies that $\text{TrigArc}(\alpha)[-1]$ is embedded and cuts off a bigon on its right-hand side. As $\alpha[-1](1) = \text{TrigArc}(\alpha)[-1](1) \subset \partial_h N$, the snippet $\text{TrigArc}(\alpha)[-1]$ is a bigon snippet of type $\mathbb{B}(h, h)$. We recall that $\alpha[1]$ is adjacent to the vertical dual $\alpha[0]$. Hence, as long as the homotopy does not change the weak snippet homotopy type of $\alpha[0]$, the subarc $\text{Hom}(\alpha[0 : 2], 1)[-1]$ consists of snippets that cut off regions of index zero on their left-hand side. So, if $\text{Hom}(\alpha[0 : 2], 1)[1 : -1]$ is

empty, then the “new” trigon $\text{Hom}(\alpha[0 : 2], 1)[-1]$ is adjacent to a snippet that is weakly snippet homotopic to $\alpha[0]$, thus is a left vertical dual. Else, it is adjacent to the snippet $\text{Hom}(\alpha[0 : 2], 1)[-2]$, which cuts off a region of index zero on its left-hand side. By induction, if $\text{TrigArc}(\alpha)[-1]$ and $\alpha[-1]$ are not weakly snippet homotopic, then $\text{TrigArc}(\alpha)[-2]$ cuts off a region of index zero on its left-hand side. As $\text{TrigArc}(\alpha)[-1](0) \subset \partial_h N$, $\text{TrigArc}(\alpha)[-2]$ is a left vertical dual. \square

3.6 The algorithm TrigCurve

In the last section we saw that the process of repeatedly applying trigon homotopies to an almost efficient arc α of trigon type eventually yields a homotopic arc α' whose inside is in efficient position. In this chapter, we are applying a similar line of reasoning to see that the repeated application of local homotopies to the unique trigon snippet of an almost efficient curve of trigon type yields efficient position or shortens the curve until it consists of a single snippet only. For a formal statement of the corresponding algorithm `TrigCurve` we refer the reader to Algorithm 3.6.1.

Algorithm 3.6.1: TrigCurve - Homotoping a curve with at most one bad snippet into efficient position or shortening it to consist of a single snippet. The bad snippet of the input curve is required to be a trigon snippet.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and a curve $\alpha \subset S$ with at most one bad snippet, which must be a trigon snippet.

Output: A curve α' homotopic to α such that α' is in efficient position or $\text{len}(\alpha') = 1$.

```

1  $\alpha' = \alpha$ 
2 while  $\text{len}(\alpha') > 1$  and there is some  $k$  satisfying  $0 \leq k \leq \text{len}(\alpha') - 1$  such that  $\alpha[k]$  is a bad snippet do
3   |  $\alpha' = \text{Hom}(\alpha', k)$ 
4 return  $\alpha'$ 

```

Lemma 3.6.1. *The algorithm TrigCurve is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2 \cdot \text{len}(\alpha))$ time.*

Proof. By assumption on the input of the algorithm `TrigCurve`, we are given a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and a curve α that contains at most one bad snippet $\alpha[k]$. This is a trigon snippet. We note that $\text{len}(\alpha_{\text{trim}}) = \text{len}(\alpha)$ as $\alpha = \alpha_{\text{trim}}$. Arguing as in the case of the algorithm `TrigArc`, Lemma 3.4.11 tells us that the algorithm exits the while-loop

after at most $(\text{len}(\alpha) + 1) \cdot (s + 2)$ many iterations. Thus, checking the validity of the while-loop condition and applying the trigon homotopy in each iteration of the while-loop sums up to at most $O(\chi(S)^2 \cdot (\text{len}(\alpha) + 1))$ operation being required before efficient position is achieved or the curve is shortened to consist of a single snippet. As $\text{len}(\alpha) \geq 1$, also $2 \cdot \text{len}(\alpha) \geq (\text{len}(\alpha) + 1)$, so the algorithm `TrigCurve` halts in $O(\chi(S)^2 \cdot \text{len}(\alpha))$ time. \square

Corollary 3.6.2. *For any input (S, N, α) , the algorithm `TrigCurve` terminates in $O(\chi(S)^2 \cdot (\text{len}_{red}(\alpha) + 1))$ time.*

Proof. Again, this follows from Lemma 3.5.1 and 3.2.13 if we keep in mind that α contains at most one bad snippet. \square

Chapter 4

Local bigon homotopies

In the previous chapter, we studied one family of bad snippets in detail: trigon snippets. There is one other important family of bad snippets: bigon snippets. These are the focus of this chapter. We will see that we have to alter our approach from the trigon snippet case: Suppose that we are given an almost efficient arc or curve of bigon type. Applying one local homotopy might yield another almost efficient arc or curve of bigon type. However, it might also yield an arc or curve that contains up to two trigon snippets. Therefore, we can not simply apply bigon homotopies repeatedly, and if we wish to make use of the algorithm **TrigArc** we have to be careful to select appropriate almost efficient subarcs.

These issues can be bypassed as follows: suppose that α is an almost efficient arc of bigon type, with $\alpha[k]$ being the unique bad snippet in α_{trim} . As in the case of trigon snippets we will see that $\text{Hom}(\alpha[k-1 : k+2], 1)_{\text{trim}}$ is in efficient position. Thus, if $k = -2$, that is, if the bigon snippet $\alpha[k]$ is the second to last snippet of the arc, then applying one local homotopy yields an arc that contains at most one bad snippet in its inside. A careful analysis will show that this snippet has to be a trigon snippet. This allows us to employ the algorithm **TrigArc** to achieve efficient position for the inside of the arc.

The layout of this chapter is as follows: First, we discuss the effects that local homotopies have on almost efficient arcs and curves of bigon type. Again, we put special emphasis on the various notions of length and their changes under the bigon homotopies. Secondly, we present the algorithm **BigArc** which yields efficient position for the inside of an arc α containing a unique bad snippet, $\alpha[-2]$, in its inside. We close this chapter by giving bounds on the running time of **BigArc** and the snippet length of its output. We remark that most of the very detailed observations made about bigon homotopies are not required to obtain the bounds

on the running time and length of the output of `BigArc`. However, they will be essential when working with curves containing a single bad snippet (see Chapter 6).

4.1 Local homotopies for bigon arcs

Throughout this section, we fix a surface $S = S_{g,b}$ satisfying $\xi(S) = 3g - 3 + b \geq 1$. We further fix a large train track $\tau \subset S$ and a tie neighbourhood $N = N(\tau) \subset S$. As usual, we do not distinguish between snippets and their strong snippet homotopy classes unless otherwise stated. For any snippet $a \subset R \in \mathcal{R}$, we always assume that a has minimal self-intersection, intersects the one-skeleton of \mathcal{R} perpendicularly and misses the corners of \mathcal{R} . We further recall that arcs and curves in S are assumed to be self-transverse and transverse to $\partial\mathcal{R}$. Moreover, if α is an arc, we assume that $\alpha(0)$ and $\alpha(1)$ lie in $\partial\mathcal{R}$. Thus, arcs and curves in S admit canonical decompositions into snippets.

Lemma 4.1.1. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose that $\alpha[k] \subset \alpha_{trim}$ is a bad snippet of bigon type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, or $\mathbb{R}(v, v)$. If $\text{len}(\alpha) = 2$, then $\text{len}(\text{Hom}(\alpha, k)) = 1$. Else, that is if $\text{len}(\alpha) \geq 3$, we have that $\text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha) - 2$. Furthermore, if $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position, then one of the following three statements holds.*

1. $\text{Hom}(\alpha[k-1 : k+2], 1)$ is a bigon snippet or a trigon snippet and

$$\text{len}_{red}(\text{Hom}(\alpha, k)) \leq \text{len}_{red}(\alpha) - 2.$$

2. $\text{Hom}(\alpha, k)$ is in efficient position.
3. $\text{Hom}(\alpha, k)$ is a peripheral curve consisting of a single snippet only.

Proof. We prove the statements of the lemma in order. As $\alpha[k]$ is a bigon, $\alpha[k](0)$ and $\alpha[k](1)$ lie on the same component of $\partial_h R$ or $\partial_v R$ of some rectangle $R \in \mathcal{R}_{tie}$. By assumption on the specific bigon snippet types, the boundary of the bigon does not contain any points of $\partial^2\mathcal{R}$ (see Figures 4.4-4.5). Hence, $\alpha[k-1]$ and $\alpha[k+1]$ lie inside the same region $R' \in \mathcal{R}$ and the arc $\alpha[k-1 : k+1]$ is replaced by a single snippet $\alpha'[k-1]$.

To prove the remaining claims of the lemma, we split the five possible types of bigon snippets into two groups and consider them separately.

Firstly, let us assume that $\alpha[k]$ is a bigon snippet of type $\mathbb{S}(t, t, 0)$, $\mathbb{B}(t, t)$ or $\mathbb{R}(v, v)$. If $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position, we claim that $\text{Hom}(\alpha[k-1 :$

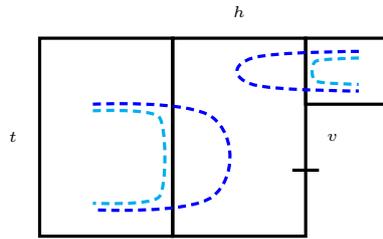


Figure 4.1: Homotopies of type $\mathbb{S}(t, t, 0)$.

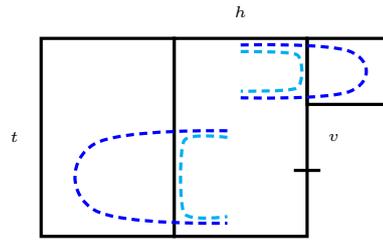


Figure 4.2: Homotopies of type $\mathbb{B}(t, t)$.

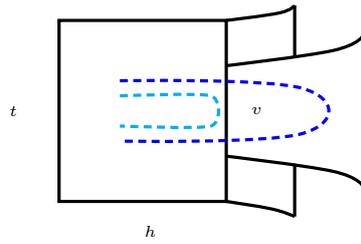


Figure 4.3: A homotopy of type $\mathbb{R}(v, v)$.

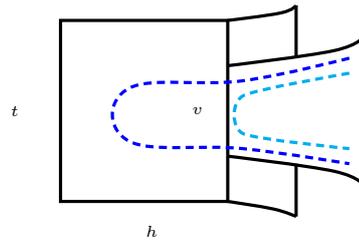


Figure 4.4: A homotopy of type $\mathbb{S}(v, v, 0)$.

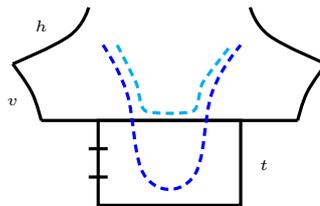


Figure 4.5: A homotopy of type $\mathbb{S}(h, h, 0)$.

$k + 2], 1)$ is a bigon snippet and satisfies $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 2$. This can be seen as follows: since $\alpha[k - 1]$ and $\alpha[k + 1]$ are efficient snippets inside a branch or switch rectangle, we know that $\alpha[k - 1] \neq \alpha[k + 1]$ and that both snippets must be carried and have corner length one or three. Thus, $\alpha[k - 1](0)$ and $\alpha[k + 1](1)$ lie on the same side of a branch or switch rectangle. This implies that $\text{Hom}(\alpha[k - 1 : k + 2], 1)$ is a bigon snippet. As the corner length of snippets inside a branch or switch rectangle is independent of their form, we further see that $\text{len}_{\text{corn}}(\text{Hom}(\alpha[k - 1 : k + 2], 1)) = \text{len}_{\text{corn}}(\alpha[k - 1])$. As $\text{len}_{\text{corn}}(\alpha[k + 1]) \geq 1$ and $\text{len}_{\text{corn}}(\alpha[k]) = 0$ only if $\text{len}_{\text{corn}}(\alpha[k + 1]) = 3$, this implies that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha[k - 1 : k + 2], 1)) \leq \text{len}_{\text{corn}}(\alpha[k - 1 : k + 2]) - 2.$$

Since no dual snippets are affected by the homotopy, the number of blockers of α and $\text{Hom}(\alpha, k)$ coincides. Thus, we see that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 2$, giving 1.

Secondly, let us assume that $\alpha[k]$ is a bigon snippet of type $\mathbb{S}(h, h, 0)$ or $\mathbb{S}(v, v, 0)$. If $\alpha[k - 1] = \alpha[k + 1]$, then $\text{Hom}(\alpha, k)$ consists of a single snippet that satisfies $\text{wind}(\text{Hom}(\alpha, k)) = \text{wind}(\alpha[k - 1])$. Thus, if $\alpha[k - 1] \subset R \in \mathcal{R}_{\text{comp}}$ is in efficient position, it has a winding number which is a non-trivial multiple of $|\partial^2 R|$. As $\text{wind}(\text{Hom}(\alpha, k)) = \text{wind}(\alpha[k - 1])$, this implies that $\text{Hom}(\alpha, k)$ is a peripheral curve consisting of a single snippet only, giving 3.

For the remainder of this proof we assume that $\alpha[k - 1] \neq \alpha[k + 1]$ and that both of these snippets are in efficient position. We distinguish the following two cases: either at least one of the snippets $\alpha[k - 1]$ and $\alpha[k + 1]$ has corner length $2s$, or both have a corner length that is strictly smaller than $2s$. We remark that in the latter case, efficient position implies that both snippets must be horizontal or vertical duals.

Let us first assume that at least one of the snippets $\alpha[k - 1]$ and $\alpha[k + 1]$ has corner length $2s$. Without loss of generality, we assume that this is the snippet $\alpha[k - 1]$. Then $\text{len}_{\text{corn}}(\text{Hom}(\alpha[k - 1 : k + 2], 1)) \leq 2s \leq \text{len}_{\text{corn}}(\alpha[k - 1])$. We note that $\text{len}_{\text{corn}}(\alpha[k]) = 3$ and $\text{len}_{\text{corn}}(\alpha[k + 1]) \geq 1$. Thus, we see that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha[k - 1 : k + 2], 1)) \leq \text{len}_{\text{corn}}(\alpha[k - 1 : k + 2]) - 4.$$

As snippets of corner length $2s$ cannot be part of any blocker, the homotopy reduces the number of blockers of the underlying arc or curve by at most one. This implies that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 2$, giving 1 and 2 if $\text{Hom}(\alpha[k - 1 : k + 2], 1)$ is in efficient position.

Let us now assume that the corner length of $\alpha[k - 1]$ and $\alpha[k + 1]$ is strictly

smaller than $2s$. As remarked previously, efficient position implies that both of these snippets cut off regions of index zero on one of their sides. If they cut off a region of index zero on the same side, an index-argument shows that $\text{Hom}(\alpha[k-1 : k+2], 1)$ is in efficient position, giving 2. Therefore, let us assume that they cut off the regions of index zero on different sides. In this case $\text{Hom}(\alpha[k-1 : k+2], 1)$ is a bigon of type $\mathbb{R}(h, h)$ or $\mathbb{R}(v, v)$. Without loss of generality, we may further assume that $\text{len}_{\text{corn}}(\alpha[k-1]) \leq \text{len}_{\text{corn}}(\alpha[k+1])$. Then the bigon cut off by $\text{Hom}(\alpha[k-1 : k+2], 1)$ is contained inside the region of index zero cut off by $\alpha[k+1]$. However, the boundary of the bigon contains two components of $\partial\mathcal{R} - \partial^2\mathcal{R}$ less than the boundary of the region of index zero cut off by $\alpha[k+1]$. Hence, we see that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha[k-1 : k+2], 1)) \leq \text{len}_{\text{corn}}(\alpha[k+1]) - 2.$$

Since $\text{len}_{\text{corn}}(\alpha[k]) = 3$ and $\text{len}_{\text{corn}}(\alpha[k+1]) \geq 1$, this implies that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha[k-1 : k+2], 1)) \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - 6.$$

As the homotopy reduces the number of blockers by at most two, we can again conclude that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 2$, which completes the proof of the lemma. \square

As in the case of trigon homotopies, we see that homotopies for bigon snippets of type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, and $\mathbb{R}(v, v)$ can be computed in $O(|\chi(S)|)$ time:

Lemma 4.1.2. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose that $\alpha[k] \subset \alpha_{\text{trim}}$ is a bad snippet of bigon type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, or $\mathbb{R}(v, v)$. Then $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time.*

Proof. By assumption, α is a snippet-decomposed arc or curve containing a bigon snippet $\alpha[k]$ of the required type. We recall that α is determined by its cutting sequence and the winding numbers of the respective snippets. Lemma 4.1.1 implies that the subarc $\alpha[k-1 : k+2]$ is replaced by the single snippet $a = \text{Hom}(\alpha, k)[k-1]$ under the homotopy. Thus, the cutting sequence of $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time.

Assuming that arithmetic on winding numbers can be done in constant time, we claim that the winding number of a can be calculated in $O(|\chi(S)|)$ time given the winding numbers of α . For snippets of type $\mathbb{S}(t, t, 0)$, $\mathbb{B}(t, t)$ or $\mathbb{R}(v, v)$ this is immediate since a lies inside the tie neighbourhood.

Thus, suppose that $\alpha[k]$ is of type $\mathbb{S}(h, h, 0)$ or $\mathbb{S}(v, v, 0)$. If $\text{len}(\alpha) = 2$ and $\alpha[k-1]$ lies inside a peripheral annulus, then $\text{Hom}(\alpha, k)$ consists of a single snippet only and $\text{wind}(\text{Hom}(\alpha, k)) = \text{wind}(\alpha[k-1])$. Suppose that $\text{len}(\alpha) > 2$ and that $\alpha[k-1]$ and thus $\alpha[k+1]$ lie inside a peripheral annulus. We can distinguish the following two cases: either $\alpha[k-1]$ and $\alpha[k+1]$ have empty intersection with ∂S , or at least one of their boundary points lies in ∂S . In the first case we see that

$$\text{wind}(a) = \text{wind}(\alpha[k-1]) + \text{wind}(\alpha[k+1]).$$

In the second case, at least one of the boundary points of a lies in ∂S . Thus, $\text{wind}(a) = 0$. This implies that the winding numbers of $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time in all cases. \square

Lemma 4.1.3. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose that $\alpha[k] \subset \alpha_{\text{trim}}$ is a bad snippet of bigon type $\mathbb{S}(t, v, 1)$. Then $\text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha) - 1$. Furthermore, if $\alpha[k-1]$ is in efficient position, then $\text{Hom}(\alpha, k)[k-1]$ is in efficient position or a trigon snippet of type $\mathbb{B}(h, t)$ or $\mathbb{R}(h, v)$. If $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position, then one of the following two statements holds.*

1. $\text{Hom}(\alpha[k-1 : k+2], 1)$ consists of one trigon snippet of type $\mathbb{B}(h, t)$ and one trigon snippet of type $\mathbb{R}(h, v)$. These are turning into the same direction. Further, $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 4$.
2. $\text{Hom}(\alpha[k-1 : k+2], 1)$ consist of one trigon snippet of type $\mathbb{B}(h, t)$ and one snippet in efficient position and $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) + 2s - 5$.

Proof. We first observe that $\text{len}(\alpha) \geq 3$ as $\alpha[k-1]$ and $\alpha[k+1]$ are contained in a branch rectangle and a complementary region of N respectively. As $\alpha[k]$ is of weight one, the bigon homotopy replaces the subarc $\alpha[k-1 : k+2]$ by a subarc whose inside intersects $\partial \mathcal{R}$ exactly one (see Figure 4.6). Thus, we see that $\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) = 2$. This implies that $\text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha) - 1$.

Without loss of generality, we may assume that $\alpha[k]$ is turning right. If $\alpha[k-1]$ is in efficient position and embedded, it cuts off a region of non-positive index on either side. Under the homotopy, the index of this region increases by at most $1/4$. Hence, $\text{Hom}(\alpha, k)[k-1]$ is a right-turning trigon snippet or in efficient position. If $\alpha[k-1]$ is in efficient position but not embedded, its winding number changes by at most one under the homotopy. As non-embedded snippets have a winding number whose modulus is strictly greater than two, the snippet $\text{Hom}(\alpha, k)[k-1]$ will have a winding number whose modulus is greater than or equal to two. Thus,

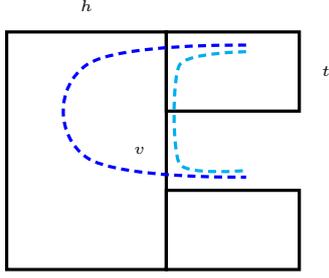


Figure 4.6: A homotopy of type $\mathbb{S}(t, v, 1)$.

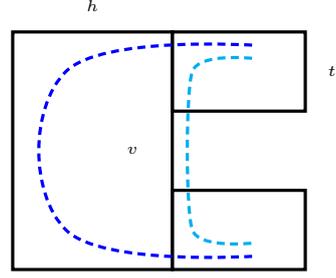


Figure 4.7: A homotopy of type $\mathbb{S}(t, t, 2)$.

$\text{Hom}(\alpha, k)[k-1]$ is in efficient position. We recall that $\alpha[k-1]$ lies inside a branch rectangle or complementary region. Thus, if $\alpha[k-1]$ is in efficient position, then $\text{Hom}(\alpha, k)[k-1]$ is in efficient position or a trigon snippet of type $\mathbb{B}(h, t)$ or $\mathbb{R}(h, v)$.

Let us now assume that $\alpha[k-1]$ and $\alpha[k+1]$ are both in efficient position. Without loss of generality, we may assume that $\alpha[k-1]$ is the snippet that lies inside a branch rectangle. Following previous arguments, $\text{Hom}(\alpha, k)[k-1]$ is a trigon snippet inside the branch rectangle, and therefore of type $\mathbb{B}(h, t)$. The previous paragraph implies that $\text{Hom}(\alpha, k)[k]$ is a trigon snippet or in efficient position. We note that $\text{Hom}(\alpha, k)[k-1]$ and $\text{Hom}(\alpha, k)[k]$ turn the same way.

Let us first assume that $\text{Hom}(\alpha, k)[k]$ is a trigon snippet. Then $\alpha[k+1]$ is a horizontal dual and $\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k]) = \text{len}_{\text{corn}}(\alpha[k+1]) - 1$. We know that $\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k-1]) = \text{len}_{\text{corn}}(\alpha[k-1])$ and that $\text{len}_{\text{corn}}(\alpha[k]) = 3$. Since no vertical duals of α are affected by the homotopy, this implies that

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 4.$$

This gives 1. If $\text{Hom}(\alpha, k)[k]$ is in efficient position, its length is bounded by $2s$. As $\text{len}_{\text{corn}}(\alpha[k+1]) \geq 1$ and the number of blockers of the arc or curve does not decrease under the homotopy, we conclude that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) + 2s - 5$, obtaining 2. \square

Remark 4.1.4. As in the case of the previously discussed types of local homotopies, we see that $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time if $\alpha[k]$ is a snippet of type $\mathbb{S}(t, v, 1)$. This follows from the following two observations: firstly, Lemma 4.1.3 states that $\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) = 2$. Thus, the cutting sequence can be adjusted in constant time. Secondly, at most one of the snippets of $\text{Hom}(\alpha[k-1 : k+2], 1)$ can lie inside a peripheral annulus. Let us assume that this is the snippet $\text{Hom}(\alpha, k)[k]$. Then $\text{wind}(\text{Hom}(\alpha, k)[k]) = \text{wind}(\text{Hom}(\alpha[k+1])) \pm 1$, so assuming that arithmetic on winding numbers can be done in constant time, this implies that $\text{Hom}(\alpha, k)$ can

be computed in $O(|\chi(S)|)$.

Lemma 4.1.5. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose that $\alpha[k] \subset \alpha_{\text{trim}}$ is a bad snippet of bigon type $\mathbb{S}(t, t, 2)$. Then $\text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha)$. Moreover, the following statements hold.*

1. *If $\alpha[k]$ turns right, then $\text{Hom}(\alpha, k)[k]$ is a left vertical dual.*
2. *If $\text{len}(\alpha) = 2$, then $\text{Hom}(\alpha, k)[k - 1]$ is a bigon snippet of type $\mathbb{B}(h, h)$.*
3. *If $\text{len}(\alpha) > 2$, $\alpha[k]$ turns right, and $\alpha[k - 1]$ is in efficient position, then $\text{Hom}(\alpha, k)[k - 1]$ is a right-turning trigon snippet inside a branch rectangle.*
4. *If $\text{len}(\alpha) > 2$ and $\alpha[k - 1]$ and $\alpha[k + 1]$ are in efficient position, then*

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 2.$$

Proof. We prove the statements of the lemma in order. As $\alpha[k]$ has weight two, the subarc $\alpha[k - 1/3, k + 4/3]$ is replaced by a subarc intersecting $\partial\mathcal{R}$ exactly twice. This implies that $\text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha)$. Further, the snippet lying between these two intersection points is parallel to a component of $\partial_v N$, thus cuts off a region of index zero on the side where the bigon cut off by $\alpha[k]$ lies. Hence, if $\alpha[k]$ turns right, then $\text{Hom}(\alpha, k)[k]$ is a left vertical dual, giving 1.

If $\text{len}(\alpha) = 2$, then $\alpha[k - 1]$ is a carried snippet inside a branch rectangle (see Figure 4.8). The homotopy increases the index of the region cut off by $\alpha[k - 1]$ by $1/2$, so $\text{Hom}(\alpha, k)[k - 1]$ is a bigon snippet of type $\mathbb{B}(h, h)$, giving 2.

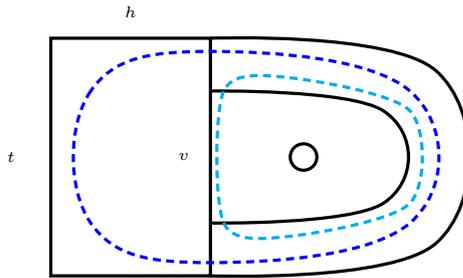


Figure 4.8: A homotopy of type $\mathbb{S}(t, t, 2)$ applied to a curve of snippet length two.

If $\text{len}(\alpha) > 2$ and $\alpha[k - 1]$ is in efficient position, then $\alpha[k - 1]$ must be a carried snippet inside a branch rectangle. Thus, if $\alpha[k]$ turns right, the index of the region cut off by $\alpha[k - 1]$ on its right-hand side increases by $1/4$. This implies that $\text{Hom}(\alpha, k)[k - 1]$ is a right-turning trigon of type $\mathbb{B}(h, t)$, giving 3.

To prove the last claim let us assume that $\text{len}(\alpha) > 2$ and that $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position. Then

$$\begin{aligned}\text{len}_{\text{corn}}(\alpha[k-1]) &= \text{len}_{\text{corn}}(\alpha[k+1]) = \text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k-1]) \\ &= \text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k+1]) = 1.\end{aligned}$$

Moreover, we know that $\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k]) = \text{len}_{\text{corn}}(\alpha)[k] - 2$. Thus, if $\text{len}(\alpha) > 2$ and $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position, we see that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{corn}}(\alpha) - 2.$$

As no vertical duals are affected by the homotopy, this gives 4. \square

Remark 4.1.6. Again, we see that $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time if $\alpha[k]$ is a snippet of type $\mathbb{S}(t, t, 2)$. Firstly, Lemma 4.1.5 states that

$$\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) = 3.$$

Thus, the cutting sequence can be adjusted in constant time. Secondly, the only snippet of $\text{Hom}(\alpha[k-1 : k+2], 1)$ that can lie inside a peripheral annulus is the snippet $\text{Hom}(\alpha[k-1 : k+2], 1)[1]$. Lemma 4.1.5 implies that this snippet is a vertical dual snippet. Thus, the modulus of its winding number is two, which implies that the winding numbers of $\text{Hom}(\alpha, k)$ can be computed in constant time as well.

Lemma 4.1.7. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose that $\alpha[k] \subset \alpha_{\text{trim}}$ is a bad snippet of bigon type $\mathbb{R}(h, h)$. Then the following statements hold.*

1. $\text{len}(\text{Hom}(\alpha, k)) \leq \text{len}(\alpha) + s - 2$.
2. If $\text{len}(\alpha) = 2$, then $\text{Hom}(\alpha, k)$ is an inessential curve of length one or a curve containing a unique bad snippet of bigon type. In the latter case we know that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha)$ and that the bigon snippet does not intersect $\partial_h N$.
3. If $\text{len}(\alpha) > 2$, then $\text{Hom}(\alpha[k-1 : k+2], 1)_{\text{trim}}$ consists of carried snippets only.
4. Assume that $\text{len}(\alpha) > 2$, that $\alpha[k]$ turns right, that $\alpha[k-1]$ is in efficient position, and that $\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) > 1$. Then $\text{Hom}(\alpha, k)[k-1]$ is a right-turning trigon of type $\mathbb{B}(h, t)$, $\mathbb{S}(h, t, 1)$ or $\mathbb{S}(h, t, 3)$.

Moreover, if $\text{len}(\alpha) > 2$ and $\alpha[k-1]$ and $\alpha[k+1]$ are both in efficient position, then one of the following two statements holds.

I. $\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) = 1$ and $\text{Hom}(\alpha, k)[k-1]$ is a bigon snippet of type $\mathbb{B}(h, h)$ or $\mathbb{S}(h, h, 0)$. Furthermore, it holds that

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 1.$$

II. $\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) > 1$ and $\text{Hom}(\alpha[k-1 : k+2], 1)$ contains exactly two bad snippets which are trigon snippets turning into the same direction. Furthermore, it holds that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha)$.

Proof. We prove the statements of the lemma in order. Suppose that $B \subset R \in \mathcal{R}_{\text{comp}}$ is the bigon cut off by the snippet $\alpha[k]$. Then the homotopy increases the number of snippets by $|\partial B \cap \partial^2 \mathcal{R}| - 1 \leq s - 2$. Thus, we see that $\text{len}(\text{Hom}(\alpha, k)) \leq \text{len}(\alpha) + s - 2$, giving 1.

Let us now assume that $\text{len}(\alpha) = 2$. Then either $\alpha[k](0)$ and $\alpha[k](1)$ lie on the same or on two different horizontal sides of a branch or switch rectangle (see Figure 4.9). In the first case, $\text{Hom}(\alpha, k)$ is an inessential curve of snippet length one. For the second case, that is if $\alpha[k-1]$ is a dual snippet, let us assume that $\alpha[k]$ is turning right. Then the region cut off by $\alpha[k-1]$ on the right-hand side loses two outward-pointing corners under the homotopy. Thus, $\text{Hom}(\alpha, k)[k-1]$ is a bigon snippet whose boundary points lie inside the tie neighbourhood and

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k-1]) = \text{len}_{\text{corn}}(\alpha[k-1]).$$

Further, by definition of the corner length of the snippet $\alpha[k]$, we have that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)) - \text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k-1]) = \text{len}_{\text{corn}}(\alpha[k]).$$

Since no vertical duals are affected by the homotopy, this gives 2.

For the remainder of this proof we assume that $\text{len}(\alpha) > 2$. We distinguish two cases: either $\alpha[k](0)$ and $\alpha[k](1)$ lie in the same component of $\partial \mathcal{R} - \partial^2 \mathcal{R}$, or they do not (see Figure 4.10).

If $\alpha[k](0)$ and $\alpha[k](1)$ lie in the same component of $\partial \mathcal{R} - \partial^2 \mathcal{R}$, then $\text{Hom}(\alpha[k-1 : k+2], 1)$ consists of a single snippet only, giving 3. If, in addition, $\alpha[k-1]$ and $\alpha[k+1]$ are both in efficient position, they must be ties of the same branch or switch rectangle. Thus, $(\text{Hom}(\alpha[k-1 : k+2], 1))$ is a bigon snippet of type $\mathbb{B}(h, h)$ or

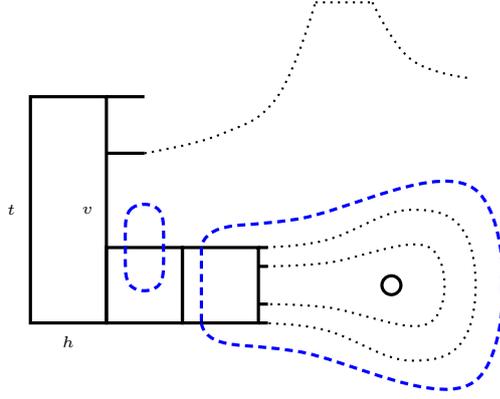


Figure 4.9: Homotopies of type $\mathbb{R}(h, h)$ for a curve of snippet length two.

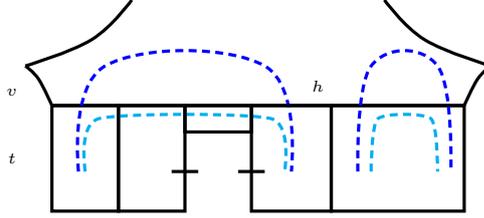


Figure 4.10: Two different kinds of trigon snippets of type $\mathbb{R}(h, h)$.

$\mathbb{S}(h, h, 0)$. Since

$$\begin{aligned} \text{len}_{\text{corn}}(\text{Hom}(\alpha[k-1 : k+2], 1)) &= \text{len}_{\text{corn}}(\alpha[k-1]) \\ &\leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - \text{len}_{\text{corn}}(\alpha[k+1]), \end{aligned}$$

and no blockers are affected by the homotopy, we see that

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 1$$

or $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 3$ if $\text{Hom}(\alpha, k)[k-1]$ is a snippet of type $\mathbb{B}(h, h)$ or $\mathbb{S}(h, h, 0)$ respectively. This gives I .

If $\alpha[k](0)$ and $\alpha[k](1)$ lie in different components of $\partial\mathcal{R} - \partial^2\mathcal{R}$, then $\text{Hom}(\alpha[k-1 : k+2], 1)_{\text{trim}}$ is empty or consists of snippets that are all parallel to $\partial_h N$. Thus, they are all carried, giving \mathcal{J} . If $\alpha[k-1]$ is in efficient position, it must be a tie of a branch or switch rectangle. Assuming that $\alpha[k]$ is turning right, the index of the region cut off by $\alpha[k-1]$ on its right-hand side increases by $1/4$. This implies that $\text{Hom}(\alpha, k)[k-1]$ is a right-turning trigon snippet inside a branch or switch rectangle. We note that this trigon snippet cannot be of type $\mathbb{S}(h, v, 2)$ since the boundary of the trigon contains exactly one point of $\partial^2\mathcal{R}$ less than the boundary of the region cut

off by $\alpha[k-1]$. As $\alpha[k-1]$ is a tie inside a branch or switch rectangle, the boundary of the region cut off by $\alpha[k-1]$ contains two or four points of $\partial^2\mathcal{R}$. This gives 4.

We further note that $\text{len}_{\text{corn}}(\alpha[k-1]) = \text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k-1])$. Recalling the definition of the corner length of the snippet $\alpha[k]$, we see that

$$\text{len}_{\text{corn}}(\alpha[k-1 : k+2], 1) = \text{len}_{\text{corn}}(\alpha[k-1 : k+2]).$$

As no dual verticals of α are affected by the homotopy, this gives II. \square

Remark 4.1.8. Again, we see that $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time if $\alpha[k]$ is a snippet of type $\mathbb{R}(h, h)$. Lemma 4.1.7 implies that

$$\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) \leq s$$

and that $\text{Hom}(\alpha[k-1 : k+2], 1) \subset N$. Thus, the cutting sequence can be adjusted in $O(|\chi(S)|)$ time, whereas the winding numbers remain unchanged.

Lemma 4.1.9. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve satisfying $\text{len}(\alpha) \geq 2$. Suppose that $\alpha[k] \subset \alpha_{\text{trim}}$ is a bad snippet of bigon type $\mathbb{B}(h, h)$. If $\text{len}(\alpha) = 2$, then $\text{len}(\text{Hom}(\alpha, k)) = 1$. Else, that is if $\text{len}(\alpha) \geq 3$, we have that $\text{len}(\text{Hom}(\alpha, k)) = \text{len}(\alpha) - 2$. If $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position, then one of the following three statements holds.*

1. $\text{Hom}(\alpha[k-1 : k+2], 1)$ is of type $\mathbb{R}(h, v)$, $\mathbb{R}(h, h)$, or $\mathbb{R}(v, v)$ and

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) + 1.$$

2. $\text{Hom}(\alpha, k)$ is in efficient position.
3. $\text{Hom}(\alpha, k)$ is a peripheral curve consisting of a single snippet only.

Proof. We prove the statements of the lemma in order. Since $\alpha[k]$ is a bad snippet of type $\mathbb{B}(h, h)$, we know that $\alpha[k-1]$ and $\alpha[k+1]$ are contained in the same complementary region $R \in \mathcal{R}_{\text{comp}}$ (see Figure 4.11). This implies that the arc $\alpha[k-1 : k+1]$ is replaced by a single snippet $\alpha'[k-1]$. If $\alpha[k-1] = \alpha[k+1]$ and this snippet is in efficient position, then $|\text{wind}(\alpha[k-1])| = l \cdot n$ where $n = |\partial^2 R|$ and $l \in \mathbb{Z} - \{0\}$. As $\text{wind}(\text{Hom}(\alpha, k)) = \text{wind}(\alpha[k-1])$, we see that $\text{Hom}(\alpha, k)$ is a peripheral curve consisting of a single snippet only, giving 3.

For the remainder of this proof we assume that $\alpha[k-1] \neq \alpha[k+1]$ and that $\alpha[k-1]$ and $\alpha[k+1]$ are in efficient position. As in the case of bigon snippets of type $\mathbb{S}(h, h, 0)$ in the proof of Lemma 4.1.1, we distinguish the following two cases:

Since the homotopy decreases the number of blockers by at most two, we conclude that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) + 1$, giving 1. \square

Remark 4.1.10. Arguing as in the case of snippets of type $\mathbb{S}(h, h, 0)$, we see that $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$ time if $\alpha[k]$ is a snippet of type $\mathbb{B}(h, h)$.

Combining this with Lemma 4.1.2 and Remarks 3.4.5, 4.1.4, 4.1.6, and 4.1.8, we conclude the following: suppose that α is a snippet-decomposed arc or curve with a bad snippet $\alpha[k] \subset \alpha_{\text{trim}}$ of arbitrary type. Then $\text{Hom}(\alpha, k)$ can be computed in $O(|\chi(S)|)$.

From the previous six lemmas about local bigon homotopies we deduce the following.

Corollary 4.1.11. *Suppose that $\alpha \subset S$ is a snippet-decomposed arc satisfying $\text{len}(\alpha) \geq 3$. We further assume that α_{trim} contains exactly one bad snippet, $\alpha[-2]$, which is a bigon snippet of arbitrary type. Then the following statements hold.*

1. $\text{Hom}(\alpha, -2)_{\text{trim}}$ contains at most one bad snippet, which must be of trigon type.
2. $\text{len}(\text{Hom}(\alpha, -2)_{\text{trim}}) \leq \text{len}(\alpha_{\text{trim}}) + s - 2$.
3. $\text{len}_{\text{red}}(\text{Hom}(\alpha, -2)_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s$.

Proof. The first two statements follow directly from Lemmas 4.1.1, 4.1.3, 4.1.5, 4.1.7 and 4.1.9 when setting $k = -2$ and observing that $\alpha[k+1] \not\subset \alpha_{\text{trim}}$.

To prove that $\text{len}_{\text{red}}(\text{Hom}(\alpha, -2)_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s$, we note that $\text{Hom}(\alpha, -2)_{\text{trim}}$ contains at most one blocker less than α_{trim} . This follows from the fact that $\alpha[k+1]$ is not contained in α_{trim} . If $\alpha[-3] \subset \alpha_{\text{trim}}$, we further know that $\text{len}_{\text{corn}}(\alpha[-3]) \geq 1$. If $\text{len}(\text{Hom}(\alpha[-3:], -2)) = 1$, this implies that

$$\begin{aligned} \text{len}_{\text{red}}(\text{Hom}(\alpha, -2)_{\text{trim}}) &= \text{len}_{\text{red}}(\alpha[1 : -3]) \\ &\leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) - 1 + 2 \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 1. \end{aligned}$$

If $\text{len}(\text{Hom}(\alpha[-3:], -2)) \neq 1$, $\alpha[-2]$ must be a bigon of type $\mathbb{S}(t, v, 1)$, $\mathbb{S}(t, t, 2)$, or $\mathbb{R}(h, h)$. Thus, no vertical duals of α are affected by the homotopy. As

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha[-3:], -2)) \leq \text{len}_{\text{corn}}(\alpha[-3:]) + 2s,$$

we see that $\text{len}_{\text{red}}(\text{Hom}(\alpha, -2)_{\text{trim}}) \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s$ in this case. \square

4.2 The algorithm BigArc

Building on the results from the previous section, we now present the algorithm BigArc. This algorithm

- takes as input a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an arc $\alpha \subset S$ whose inside contains at most one bad snippet, which must be its penultimate snippet, and
- outputs an arc α' that is homotopic to α and in efficient position in $\alpha'[1 : -1]$.

The formal statement of BigArc is given in Algorithm 4.2.1. Corollary 4.1.11 shows that if α is an arc with a unique bad snippet, $\alpha[-2]$, of bigon type, then applying one local bigon homotopy is sufficient to obtain an arc that is a valid input for the algorithm TrigArc. Thus, efficient position for the inside of α is achieved by employing the algorithm TrigArc.

Algorithm 4.2.1: BigArc - Achieving efficient position for the inside of an arc that contains at most one bad snippet, which must be its penultimate snippet.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an arc $\alpha \subset S$ such that $\alpha[1 : -1]$ contains at most one bad snippet, which must be the snippet $\alpha[-2]$.

Output: An arc α'' which is homotopic, relative its endpoints, to α and is in efficient position in its inside.

```

1  $\alpha' = \alpha$ 
2 if  $\text{len}(\alpha') > 2$  and  $\alpha'[-2]$  is a bigon then
3   |  $\alpha' = \text{Hom}(\alpha', -2)$ 
4 return  $\text{TrigArc}(S, N, \alpha')$ 

```

Lemma 4.2.1. *The algorithm BigArc is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2(\text{len}(\alpha_{trim}) + s - 1))$ time. For $\alpha'' = \text{BigArc}(S, N, \alpha)$, we have that*

$$\text{len}_{red}(\alpha''_{trim}) \leq \text{len}_{red}(\alpha_{trim}) + 4s.$$

Proof. The correctness of this algorithm follows from Corollary 4.1.11 and Lemma 3.5.1. Remark 4.1.10 implies that examining the last three snippets of the input arc and replacing them, if required, by $\text{Hom}(\alpha[-3 :], 1)$, takes $O(|\chi(S)|)$ time. Further, any local bigon homotopy increases the snippet count of the underlying arc by at most $s - 2$. Thus, Lemma 3.5.1 implies that the algorithm halts within a further

$O(\chi(S)^2(\text{len}(\alpha_{\text{trim}}) + s - 1))$ operations and that the reduced corner length of the output is bounded by $\text{len}_{\text{red}}(\alpha_{\text{trim}}) + 2s + 2s$. \square

Corollary 4.2.2. *For any input (S, N, α) , the algorithm *BigArc* halts in*

$$O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha_{\text{trim}}) + 1))$$

time.

Proof. As α_{trim} contains at most one bad snippet, Lemma 3.2.13 implies that

$$\text{len}(\alpha_{\text{trim}}) + s - 1 \leq \text{len}_{\text{red}}(\alpha_{\text{trim}}) + s,$$

which is sufficient to prove the claim. \square

Chapter 5

Obtaining efficient position for all snippets but one

Building on the results from the previous two chapters, we now present the algorithm `ReduceToTwoBadSnippets` that “almost” yields efficient position for properly immersed arcs and curves. For a formal statement of this algorithm we point the reader to Algorithm 5.1.1. Suppose that α is a snippet-decomposed arc or curve. On input α , `ReduceToTwoBadSnippets` outputs an arc or curve α' homotopic to α such that $\alpha'[1 : -1]$ is in efficient position. If α is a properly immersed, essential arc, this implies that α' is in efficient position. If α is a curve, then α' can have up to two bad snippets: its very first and very last snippet. To reduce the number of bad snippets of the curve α' further, some additional work is needed. To decrease the number of bad snippets to be less than two, a simple modification of our previous discussions is sufficient. We present the corresponding algorithm `ReduceToOneBadSnippet` in this chapter. For a formal statement of this algorithm we point the reader to Algorithm 5.2.1. However, eliminating the last bad snippet turns out to be quite complex: When applying local homotopies as before, the number of bad snippets alternates between one and two. Moreover, these snippets might change their turning direction, so monitoring the number of right or left duals does not work. These issues will be resolved by applying local homotopies in a specific order, ensuring that the reduced corner length of the underlying curve decreases sufficiently often. As this discussion is rather delicate, it is postponed to the subsequent chapter.

Throughout this chapter, we assume $S = S_{g,b}$ to be a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. We further assume that $\tau \subset S$ is a large train track and that $N = N(\tau) \subset S$ is a tie neighbourhood of τ in S . As usual, we do not distinguish between snippets and their strong snippet homotopy classes unless otherwise stated.

For any snippet $a \subset R \in \mathcal{R}$, we always assume that a has minimal self-intersection, intersects the one-skeleton of \mathcal{R} perpendicularly and misses the corners of \mathcal{R} . We further recall that arcs and curves in S are assumed to be self-transverse and transverse to $\partial\mathcal{R}$. Moreover, if α is an arc, we assume that $\alpha(0)$ and $\alpha(1)$ lie in $\partial\mathcal{R}$. Thus, arcs and curves in S admit canonical decompositions into snippets.

5.1 Reducing the number of bad snippets to two

Algorithm 5.1.1: ReduceToTwoBadSnippets - Achieving efficient position for all but the first and last snippet of a given arc or curve.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an arc or curve $\alpha \subset S$.

Output : An arc or curve α'' which is homotopic to α (relative its endpoints if applicable) such that $\alpha''[1 : -1]$ is in efficient position.

```

1  $\alpha' = \alpha$ 
2 if  $\text{len}(\alpha) \leq 2$  then
3   | return  $\alpha$ 
4 for  $k$  in  $[3, \dots, \text{len}(\alpha) - 1]$  do
5   |  $\alpha' = \text{BigArc}(S, N, \alpha'[-\text{len}(\alpha) + k]) \cdot \alpha[-\text{len}(\alpha) + k : ]$ 
6 return  $\text{BigArc}(S, N, \alpha')$ 

```

Lemma 5.1.1. *The algorithm `ReduceToTwoBadSnippets` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time. For*

$$\alpha'' = \text{ReduceToTwoBadSnippets}(S, N, \alpha),$$

we have that $\text{len}_{\text{red}}(\alpha''_{\text{trim}}) \leq (\text{len}(\alpha) - 1) \cdot 6s$.

Proof. We prove the correctness of the algorithm by going through the pseudocode line by line. By assumption on the input of the algorithm `ReduceToTwoBadSnippets`, we are given an arc or curve α . If $\text{len}(\alpha) \leq 2$, then $\alpha[1 : -1]$ is empty. Hence, α is a valid output for `ReduceToTwoBadSnippets`.

Let us now assume that $\text{len}(\alpha) > 2$. The statement of the for-loop is executed $\text{len}(\alpha) - 3$ many times. In the first iteration, that is for $k = 3$, we apply `BigArc` to the subarc $\alpha[0 : 3] = \alpha'[0 : 3]$, whose inside consists of the single snippet $\alpha[1]$. Thus, $\alpha'[0 : 3]$ is a valid input for `BigArc`. By Lemma 4.2.1, the arc $\text{BigArc}(S, N, \alpha'[0 : 3])$ is homotopic, relative its endpoints, to $\alpha[0 : 3]$ and is in efficient position in its inside. Therefore, the arc or curve $\alpha' = \text{BigArc}(S, N, \alpha'[0 : 3]) \cdot \alpha[3 :]$ constitutes

an improvement compared to the input arc or curve α as follows: even though the first snippet of α' might still be bad, we can now be sure that all the bad snippets of $\alpha'[1 :]$ are contained in the subarc $\alpha'[-\text{len}(\alpha) + 2 :]$. For α , this is not true. There, we only know that all the bad snippets of $\alpha[1 :]$ are contained in the subarc $\alpha[-\text{len}(\alpha) + 1 :]$. Thus, after one iteration of the for-loop, we have shortened the subarc that is guaranteed to hold all bad snippets of α' but $\alpha'[0]$ by one snippet. In particular, we see that $\alpha'[: -\text{len}(\alpha) + 4]$ now contains at most one bad snippet in its inside, which is the snippet $\alpha'[-\text{len}(\alpha) + 2] = \text{BigArc}(S, N, \alpha[0 : 3])[-1]$. This implies that $\alpha'[: -\text{len}(\alpha) + 4]$ is a valid input for **BigArc** in the next iteration of the for-loop. By induction, all further arcs $\alpha'[: -\text{len}(\alpha) + k]$ for k in $[3, \dots, \text{len}(\alpha) - 1]$ are valid inputs for **BigArc**. We remark that we exit the for-loop once α' has at most three bad snippets: $\alpha'[0]$, $\alpha'[-2]$, and $\alpha'[-1]$. Thus, α' at this stage is a valid input for **BigArc**, which concludes the proof of the correctness of the algorithm **ReduceToTwoBadSnippets**. We remark that we excluded the last iteration of **BigArc** from the for-loop as $\alpha'[: 0]$ equals, by Python convention, the empty subarc.

To analyse the running time of the algorithm **ReduceToTwoBadSnippets** we note that it takes at most $O(|\chi(S)|)$ time to advance to the for-loop in the pseudocode. To analyse the number of operations required to execute the for-loop and the final call of **BigArc**, we recall that

$$\text{len}_{\text{red}}(\alpha_{\text{trim}}) \leq \text{len}_{\text{corn}}(\alpha_{\text{trim}}) \leq 2s \cdot \text{len}(\alpha_{\text{trim}}).$$

Corollary 4.2.2 implies that the first iteration of the for-loop is executed within $O(|\chi(S)|^3 \cdot (2s + 1))$ time, while increasing the reduced corner length of $\alpha'[: -\text{len}(\alpha) + 3]_{\text{trim}}$ by at most $4s$. Assuming the maximum corner length of $2s$ for the last snippet of $\alpha'[: -\text{len}(\alpha) + 3]$, we see that

$$\text{len}_{\text{red}}(\alpha'[: -\text{len}(\alpha) + 4]_{\text{trim}}) \leq 2s + 6s.$$

From this we conclude that the next iteration of the for-loop is executed within $O(|\chi(S)|^3 \cdot (8s + 1))$ time and yields a subarc that satisfies

$$\text{len}_{\text{red}}(\alpha'[: -\text{len}(\alpha) + 5]_{\text{trim}}) \leq 2s + 2 \cdot (6s).$$

By induction, the last iteration of the for-loop terminates within

$$O(|\chi(S)|^3 \cdot (2s + (\text{len}(\alpha) - 3 - 1) \cdot 6s))$$

time and yields an arc $\alpha'[: -1] = \alpha'[: -\text{len}(\alpha) + \text{len}(\alpha) - 1]$ that satisfies

$$\text{len}_{\text{red}}(\alpha'[: -1]_{\text{trim}}) \leq 2s + (\text{len}(\alpha) - 3) \cdot 6s.$$

One final application of **BigArc** then returns an arc or curve α'' of the desired form within $O(|\chi(S)| \cdot (2s + (\text{len}(\alpha) - 3) \cdot 6s))$ time, and we see that

$$\text{len}_{\text{red}}(\alpha''[1 : -1]) \leq 2s + (\text{len}(\alpha) - 2) \cdot 6s.$$

Summarizing, this gives us a total of $\text{len}(\alpha) - 2$ many applications of **BigArc**, each being computed in $O(|\chi(S)|^3 \cdot (2s + (\text{len}(\alpha) - 3) \cdot 6s))$ time. Thus, the algorithm **ReduceToTwoBadSnippets** halts within

$$O(|\chi(S)|^3 \cdot (2s + (\text{len}(\alpha) - 3) \cdot 6s) \cdot (\text{len}(\alpha) - 2))$$

time, which is bounded by $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ as $\text{len}(\alpha) \geq 1$.

Set $\alpha'' = \text{ReduceToTwoBadSnippets}(S, N, \alpha)$. If α is an arc, then

$$\alpha''_{\text{trim}} = \alpha''[1 : -1],$$

so $\text{len}_{\text{red}}(\alpha''_{\text{trim}}) \leq 2s + (\text{len}(\alpha) - 2) \cdot 6s$. If α is a curve, then $\alpha''_{\text{trim}} = \alpha''$, so

$$\begin{aligned} \text{len}_{\text{red}}(\alpha''_{\text{trim}}) &\leq \text{len}_{\text{red}}(\alpha''[1 : -1]) + 4s \\ &\leq 2s + (\text{len}(\alpha) - 2) \cdot 6s + 4s \\ &\leq (\text{len}(\alpha) - 1) \cdot 6s. \end{aligned}$$

□

Corollary 5.1.2. *Suppose that $S = S_{g,b}$ is a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. Suppose that $\tau \subset S$ is a large train track and $N = N(\tau)$ is a tie neighbourhood of τ in S . Suppose further that $\alpha \subset S$ is a properly immersed arc. Then $\text{ReduceToTwoBadSnippets}(S, N, \alpha)$ is in efficient position with respect to N or consists of a single snippet. Moreover, $\text{ReduceToTwoBadSnippets}(S, N, \alpha)$ can be computed in polynomial time.*

Proof. Lemma 2.6.4 implies that there are no properly immersed arcs $\alpha \subset S$ whose snippet decomposition consists of exactly two snippets. As the first and last snippet of every arc α satisfying $\text{len}(\alpha) \geq 3$ are in efficient position, the claim follows from Lemma 5.1.1. □

5.2 Reducing the number of bad snippets from two to one

Suppose that S is a surface of positive complexity, that $\tau \subset S$ is a large train track, and that $N = N(\tau)$ is a tie neighbourhood of τ in S . Suppose further that $\alpha \subset S$ is a properly immersed curve. In the last section we saw that $\text{ReduceToTwoBadSnippets}(S, N, \alpha)$ might still contain up to two bad snippets. In this section we explain how to obtain a curve that contains at most one bad snippet.

5.2.1 The idea behind $\text{ReduceToOneBadSnippet}$

Suppose that α is a properly immersed curve in S . If the curve

$$\alpha' = \text{ReduceToTwoBadSnippets}(S, N, \alpha)$$

contains two bad snippets, they will be the very first and very last snippet of α' . Thus, if we cut α' open between its first and last snippet, we obtain an arc α'' that is in efficient position in its inside. We then glue a copy of $\alpha[0]$ to the end of α'' . The resulting arc $\alpha''' = \alpha'' \cdot \alpha[0]$ contains a unique bad snippet in its inside. As this is the penultimate snippet, the algorithm BigArc yields a homotopic arc which is in efficient position in its inside. By Remark 3.3.2 we know that the first two-thirds of $\alpha'''[0]$ and the last two-thirds of $\alpha'''[-1]$ remain unchanged when applying the algorithm BigArc to α''' . Thus, we can remove the first and last half of the first and last snippet of $\text{BigArc}(S, N, \alpha''')$ respectively and glue the remaining arc along its boundary points to obtain a smooth curve that is homotopic to α . This curve now contains at most one bad snippet. To formalize this process, we introduce some further notation.

5.2.2 Subsets of snippets and gluings of arcs

Suppose that $\alpha \subset S$ is a snippet-decomposed arc. Recall that for $0 \leq k < \text{len}(\alpha)$ and $0 \leq \epsilon \leq \delta \leq 1$, we set

$$\alpha[k] = \alpha[k : k + \epsilon] \cdot \alpha[k + \epsilon : k + \delta] \cdot \alpha[k + \delta : k + 1].$$

Suppose that $\alpha[\epsilon : \delta]$ and $\alpha[|\alpha| - 1 + \epsilon : |\alpha| - 1 + \delta]$ are identical as parametrized arcs for some $0 \leq \epsilon < \delta \leq 1$. Let $\epsilon < \zeta < \delta$. By $\alpha[\zeta : |\alpha| - 1 + \zeta]/\sim$ we denote the snippet-decomposed curve that we obtain by identifying the boundary points of $\alpha[\zeta : |\alpha| - 1 + \zeta]$ (see Figure 5.1). After a suitable reparametrisation, this is a

smooth, properly immersed, snippet-decomposed curve in S .

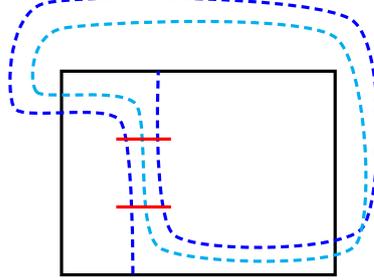


Figure 5.1: Gluing of an arc whose first and last snippet have identical middle thirds. For clarity, small push-offs have been performed in the figure.

5.2.3 The algorithm `ReduceToOneBadSnippet`

As described earlier, the algorithm `ReduceToOneBadSnippet` reduces the number of bad snippets of a curve from two to at most one. For a formal statement of `ReduceToOneBadSnippet` we refer the reader to Algorithm 5.2.1.

Algorithm 5.2.1: `ReduceToOneBadSnippet` - Reducing the number of bad snippets of a given curve to at most one.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and a curve $\alpha \subset S$ which contains at most two bad snippets $\alpha[0]$ and $\alpha[-1]$.

Output : A curve α'' that is homotopic to α such that α'' contains at most one bad snippet.

```

1 if  $\text{len}(\alpha) < 2$  then
2   | return  $\alpha$ 
3  $\alpha' = \text{BigArc}(S, N, \alpha \cdot \alpha[0])$ 
4 return  $\alpha'[1/2 : \text{len}(\alpha') - 1/2]/\sim$ 

```

Lemma 5.2.1. *The algorithm `ReduceToOneBadSnippet` is correct. On an input (S, N, α) , the algorithm terminates in $O(\chi(S)^2(\text{len}(\alpha) + s))$ time. For*

$$\alpha'' = \text{ReduceToOneBadSnippet}(S, N, \alpha),$$

we have that $\text{len}_{\text{red}}(\alpha'') \leq \text{len}_{\text{red}}(\alpha) + 7s$.

Proof. By our assumptions on the input α , the arc $\alpha' = \alpha \cdot \alpha[0]$ is a valid input for the algorithm `BigArc` with a single bad snippet, $\alpha'[\text{len}(\alpha) - 1]$, as its penultimate snippet. We remark that $\alpha'[0] = \alpha'[\text{len}(\alpha') - 1]$. Remark 3.3.2 implies that the

subarcs $\alpha'[: 2/3]$ and $\alpha'[-2/3 :]$ remain fixed under the algorithm `BigArc`. Set $\alpha' = \text{BigArc}(S, N, \alpha \cdot \alpha[0])$. Lemma 4.2.1 implies that $\alpha'[0]$ and $\alpha'[-1]$ are the only snippets of α' that can be bad. We further know that their middle thirds coincide. Therefore, $\alpha'[1/2 : \text{len}(\alpha') - 1/2]/\sim$ is a well-defined, properly immersed, snippet-decomposed curve that contains at most one bad snippet. This proves the correctness of the algorithm `ReduceToOneBadSnippet`.

The bound on its running time follows from Lemma 4.2.1: since $\text{len}(\alpha'_{\text{trim}}) = \text{len}(\alpha) - 1$, we know that the algorithm `BigArc` on input $(S, N, \alpha \cdot \alpha[0])$ terminates within $O(\chi(S)^2 \cdot (\text{len}(\alpha) + s - 2))$ time. We further know that

$$\text{len}_{\text{red}}(\text{BigArc}(S, N, \alpha \cdot \alpha[0])[1 : -1]) \leq \text{len}_{\text{red}}(\alpha[1 :]) + 4s.$$

As $\text{len}_{\text{red}}(\alpha[1 :]) \leq \text{len}_{\text{red}}(\alpha) + 4 \leq \text{len}_{\text{red}}(\alpha) + s$, this implies that

$$\begin{aligned} & \text{len}_{\text{red}}(\text{ReduceToOneBadSnippet}(\alpha)) \\ & \leq \text{len}_{\text{red}}(\text{BigArc}(S, N, \alpha \cdot \alpha[0])[1 : -1]) + \text{len}_{\text{corn}}(\text{BigArc}(S, N, \alpha \cdot \alpha[0])[0]) \\ & \leq \text{len}_{\text{red}}(\alpha[1 :]) + 4s + 2s \\ & \leq \text{len}_{\text{red}}(\alpha) + s + 4s + 2s \\ & \leq \text{len}_{\text{red}}(\alpha) + 7s. \end{aligned}$$

□

Corollary 5.2.2. *For any input (S, N, α) of the required form, the algorithm `ReduceToOneBadSnippet` terminates in $O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time.*

Proof. As α contains at most two bad snippets, Lemma 3.2.13 implies that

$$\text{len}(\alpha) + s \leq \text{len}_{\text{red}}(\alpha) + s + 2 \leq \text{len}_{\text{red}}(\alpha) + 2s,$$

which is sufficient to prove the claim. □

Chapter 6

Efficient position for curves with a single bad snippet

In this chapter, we focus on the last major piece still missing for the proof for Theorem 7.0.2. Throughout this chapter, we assume that $S = S_{g,b}$ is a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. We further assume that $\tau \subset S$ is a large train track and that $N = N(\tau) \subset S$ is a tie neighbourhood of τ in S . Suppose that $\alpha \subset S$ is a properly immersed, almost efficient curve. If α contains a trigon snippet, we already know that we can achieve efficient position or shorten the curve to consist of a single snippet in $O(\chi(S)^2 \cdot \text{len}(\alpha))$ time. However, if the bad snippet of α is not a trigon snippet, we have to undertake a delicate case analysis taking into account the various bigon snippet types.

In the following, we define a collection of algorithms whose input is an almost efficient curve α of a specified bigon type. Their output will be a curve α' homotopic to α such that one of the following three statements holds.

- α' is in efficient position or has snippet length one.
- α' contains a unique bad snippet of trigon type.
- α' contains a unique bad snippet of bigon type and satisfies

$$\text{len}_{\text{red}}(\alpha') < \text{len}_{\text{red}}(\alpha).$$

Thus, we can obtain efficient position for an almost efficient curve α or shorten it to consist of a single snippet only by applying at most $\text{len}_{\text{red}}(\alpha) + 1$ many of these algorithms.

Lemma 4.1.1 implies that for bigon snippets of type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, or $\mathbb{R}(v, v)$, these algorithms only have to execute one bigon homotopy. However, the remaining four types of bigon snippets require some subtle case analyses. Applying a single bigon homotopy of one of these types might yield

- a curve containing two trigon snippets (see page 69, Lemma 4.1.5) or
- a curve with a unique bigon snippet that has a greater reduced corner length (see page 73, Lemma 4.1.9).

Thus, in the case of these four bigon snippet types, the algorithm will have to perform more than a single bigon homotopy to yield the desired results. We begin our discussion with the more accessible cases of bigon snippets of type $\mathbb{S}(t, v, 1)$ and $\mathbb{S}(t, t, 2)$. Afterwards, we address the delicate cases of bigon snippets of type $\mathbb{R}(h, h)$ and $\mathbb{B}(h, h)$. We conclude this chapter by presenting the algorithm `SingleBadSnippet` which takes as input an almost efficient curve and outputs a homotopic curve that consists of a single snippet only or is in efficient position.

6.1 WeightOneBigon-algorithm for almost efficient curves

We begin by studying almost efficient curves that contain a snippet of type $\mathbb{S}(t, v, 1)$. The corresponding algorithm `WeightOneBigon`

- takes as input a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an almost efficient curve α with a bad snippet of type $\mathbb{S}(t, v, 1)$, and
- outputs a curve homotopic to α which either contains a unique bad snippet of trigon type or contains a unique bad snippet of bigon type and has a shorter reduced corner length than α .

For a formal statement of the algorithm `WeightOneBigon` we refer the reader to Algorithm 6.1.1.

Lemma 6.1.1. *The algorithm `WeightOneBigon` is correct. On an input (S, N, α) , the algorithm halts in $O(|\chi(S)|(\text{len}_{red}(\alpha) + 1))$ time.*

Proof. By assumption on the input of the algorithm `WeightOneBigon`, we are given an almost efficient curve α with a unique bad snippet $\alpha[k]$ of type $\mathbb{S}(t, v, 1)$. As $\alpha[k]$ is a bigon of type $\mathbb{S}(t, v, 1)$ we know that $\text{len}(\alpha) > 2$. As all other snippets of α are in efficient position, we know that $\alpha[k - 1]$ and $\alpha[k + 1]$ are in efficient position, too.

Algorithm 6.1.1: WeightOneBigon - Homotoping an almost efficient curve with a bad snippet of type $\mathbb{S}(t, v, 1)$ into an input for `TrigCurve` or into an almost efficient curve of shorter reduced corner length.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an almost efficient curve $\alpha \subset S$ that contains a bad snippet of type $\mathbb{S}(t, v, 1)$.

Output : A curve α' homotopic to α such that α' contains at most one bad snippet. If this is a trigon, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) + 2s - 5$. If this is a bigon, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) - 4$.

```

1 Set  $k$  such that  $\alpha[k]$  is bad
2  $\alpha' = \text{Hom}(\alpha, k)$ 
3 if  $\alpha'[k - 1]$  is a trigon of type  $\mathbb{R}(h, v)$  then
4 |   return  $\text{Hom}(\alpha', k - 1)$ 
5 if  $\alpha'[k]$  is a trigon of type  $\mathbb{R}(h, v)$  then
6 |   return  $\text{Hom}(\alpha', k)$ 
7 return  $\alpha'$ 

```

Set $\alpha' = \text{Hom}(\alpha, k)$. Lemma 4.1.3 implies that $\text{Hom}(\alpha[k - 1 : k + 2], 1)$ consists of two snippets $\alpha'[k - 1]$ and $\alpha'[k]$. Either both of these are trigon snippets, one of type $\mathbb{B}(h, t)$ and one of type $\mathbb{R}(h, v)$, or exactly one of them is a bad snippet, which then must be of type $\mathbb{B}(h, t)$.

Let us first assume that exactly one of the snippets $\alpha'[k - 1]$ and $\alpha'[k]$ is a bad snippet. As all other snippets of α' are in efficient position, `WeightOneBigon` then returns a curve containing a unique trigon. Lemma 4.1.3 implies that

$$\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) + 2s - 5.$$

Thus, the algorithm terminates with a valid output.

Secondly, let us assume that both snippets $\alpha'[k - 1]$ and $\alpha'[k]$ are bad. Lemma 4.1.3 implies that the curve α' satisfies $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) - 4$. Lemma 3.4.10 implies that applying one local homotopy to the trigon of type $\mathbb{R}(h, v)$ does not increase the reduced corner length of the underlying curve. Suppose that $\alpha'[k - 1]$ is the trigon of type $\mathbb{B}(h, t)$. Suppose further that $\alpha[k]$ turns right. Hence, $\alpha'[k - 1]$ and $\alpha'[k]$ are both right-turning trigons. Applying one local homotopy of type $\mathbb{R}(h, v)$ to α at $\alpha'[k]$ increases the index of the region cut off by the trigon $\alpha'[k - 1]$ by $1/4$ (see Figure 6.1). This follows from Lemma 3.4.3 and Lemma 3.4.10. Thus, we obtain an

Algorithm 6.2.1: WeightTwoBigon - Homotoping an almost efficient curve with a bigon of type $\mathbb{S}(t, t, 2)$ into an input for **TrigCurve**.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an almost efficient curve $\alpha \subset S$ that contains a bad snippet of type $\mathbb{S}(t, t, 2)$.

Output : A curve α''' homotopic to α such that α''' contains at most one bad snippet. If there is a bad snippet, then this is a trigon snippet and $\text{len}_{\text{red}}(\alpha''') \leq \text{len}_{\text{red}}(\alpha) + 2s$.

```

1 Set  $k$  such that  $\alpha[k]$  is bad
2 if  $\text{len}(\alpha) = 2$  then
3   |  $\alpha' = \text{Hom}(\alpha, k)$ 
4   | return  $\text{Hom}(\alpha', k - 1)$ 
5  $\alpha' = \text{Hom}(\alpha, k)$ 
6  $\alpha'' = \text{TrigArc}(S, N, \alpha'[k:] \cdot \alpha'[:k])$ 
7 if  $\alpha''[-1]$  is a bigon then
8   | return  $\text{Hom}(\alpha''/\sim, -1)$ 
9 return  $\alpha''/\sim$ 

```

Proof. By assumption on the input of the algorithm **WeightOneBigon**, we are given an almost efficient curve α with a unique bad snippet $\alpha[k]$ of type $\mathbb{S}(t, t, 2)$. Without loss of generality, we assume that $\alpha[k]$ turns right. As all other snippets of α are in efficient position, we know that $\alpha[k - 1]$ and $\alpha[k + 1]$ are in efficient position. Set $\alpha' = \text{Hom}(\alpha, k)$.

We have to distinguish two cases: either $\text{len}(\alpha) = 2$ or $\text{len}(\alpha) > 2$. If $\text{len}(\alpha) = 2$, then Lemma 4.1.5 implies that $\alpha'[k]$ is a left vertical dual and $\alpha'[k - 1]$ is a bigon of type $\mathbb{B}(h, h)$ (see page 69, Figure 4.8). Thus, $\text{Hom}(\alpha', k - 1)$ is a peripheral curve of snippet length one and therefore a valid output for **WeightTwoBigon**. We remark that **WeightTwoBigon** terminates in $O(|\chi(S)|)$ time in this case.

If $\text{len}(\alpha) > 2$, we know that $\alpha'[k]$ is a left vertical dual and that $\alpha'[k - 1]$ and $\alpha'[k + 1]$ are right-turning trigons. Thus, the arc that starts with the snippet $\alpha'[k]$ and ends with the snippet $\alpha'[k - 1]$, that is, the arc $\alpha'[k:] \cdot \alpha'[:k]$, has a unique bad snippet of trigon type in its inside. Corollary 3.5.4 implies that $\alpha'' = \text{TrigArc}(S, N, \alpha'[k:] \cdot \alpha'[:k])$ contains a unique bad snippet $\alpha''[-1]$, which is a trigon snippet or a bigon snippet. If $\alpha''[-1]$ is a bigon snippet, the adjacent two snippets must be left vertical duals, so an index-argument shows that $\text{Hom}(\alpha''/\sim, -1)$ is in efficient position. If $\alpha''[-1]$ is a trigon snippet, then α''/\sim is a smooth curve with a unique bad snippet of trigon type.

To analyse the running time and length of the output curve when $\text{len}(\alpha) > 2$, we note that Lemma 4.1.5 implies that $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) - 2$. It requires at most

$O(|\chi(S)|)$ many operations to replace α by α' . Since $\alpha'[k-1]$ is a trigon, the reduced corner length of α' equals the reduced corner length of the arc $\alpha'[k:] \cdot \alpha'[:k]$. Lemma 3.5.2 implies that we obtain the arc α'' within another $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations. The arc α'' satisfies

$$\text{len}_{\text{red}}(\alpha'') \leq \text{len}_{\text{red}}(\alpha'[k:] \cdot \alpha'[:k]) + 2s \leq \text{len}_{\text{red}}(\alpha') + 2s \leq \text{len}_{\text{red}}(\alpha) + 2s.$$

As $\alpha''[-1]$ is a bad snippet, $\text{len}_{\text{red}}(\alpha'') = \text{len}_{\text{red}}(\alpha''/\sim)$. Thus, if the algorithm returns α''/\sim , that is, when α''/\sim contains a unique trigon, then the reduced corner length of the output curve is bounded by $\text{len}_{\text{red}}(\alpha) + 2s$. Else, the algorithm returns a curve in efficient position within a further $O(|\chi(S)|)$ operations.

Summarizing, if $\text{len}(\alpha) > 2$, the algorithm `WeightTwoBigon` terminates within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations, which concludes the proof of the lemma. \square

6.3 AllButHorBigInCompOrBranch-algorithm for almost efficient curves

To simplify later discussions, we combine the algorithms `WeightOneBigon` and `WeightTwoBigon` with the trivial algorithms corresponding to the homotopies of type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, and $\mathbb{R}(v, v)$ into a single algorithm called `AllButHorBigInCompOrBranch`.

Lemma 6.3.1. *The algorithm `AllButHorBigInCompOrBranch` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time.*

Proof. By assumption on the input of the algorithm `AllButHorBigInCompOrBranch`, we are given a surface S of positive complexity, a tie neighbourhood N of a large train track in S , and an almost efficient curve α with a unique bad snippet $\alpha[k]$ of type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, $\mathbb{R}(v, v)$, $\mathbb{S}(t, t, 2)$, or $\mathbb{S}(t, v, 1)$. It requires at most $O(|\chi(S)|(\text{len}_{\text{red}}(\alpha) + 1))$ many operations to determine the index of this bad snippet. If this snippet is of type $\mathbb{S}(t, v, 1)$ or $\mathbb{S}(t, t, 2)$, the correctness of the algorithm and the bounds on the running time follow from Lemma 6.1.1 and Lemma 6.2.1. If the bad snippet is of type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, or $\mathbb{R}(v, v)$, Lemma 4.1.1 implies that a single homotopy is sufficient to obtain a curve of the desired shape that satisfies the required bounds on the reduced corner length. \square

Algorithm 6.3.1: AllButHorBigInCompOrBranch - Homotoping an almost efficient curve of bigon type that does not contain a bigon snippet of type $\mathbb{R}(h, h)$ or $\mathbb{B}(h, h)$ into an input for **TrigCurve** or into an almost efficient curve of shorter reduced corner length.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an almost efficient curve $\alpha \subset S$ that contains a bad snippet of type $\mathbb{B}(t, t)$, $\mathbb{S}(h, h, 0)$, $\mathbb{S}(t, t, 0)$, $\mathbb{S}(v, v, 0)$, $\mathbb{R}(v, v)$, $\mathbb{S}(t, t, 2)$, or $\mathbb{S}(t, v, 1)$.

Output: A curve α' homotopic to α such that α' contains at most one bad snippet. If this unique bad snippet is a bigon snippet, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) - 2$. If it is a trigon snippet, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) + 2s$.

```

1 Set  $k$  such that  $\alpha[k]$  is bad
2 if  $\alpha[k]$  is of type  $\mathbb{S}(t, v, 1)$  then
3   | return WeightOneBigon( $S, N, \alpha$ )
4 if  $\alpha[k]$  is of type  $\mathbb{S}(t, t, 2)$  then
5   | return WeightTwoBigon( $S, N, \alpha$ )
6 return Hom( $\alpha, k$ )

```

6.4 TwoTrigons-algorithm for almost efficient curves

It remains to understand how to handle curves that contain a unique bad snippet of type $\mathbb{R}(h, h)$ or $\mathbb{B}(h, h)$. As seen in the previous chapter, applying a single homotopy might not only increase the number of bad snippets, but can also increase the reduced corner length of the underlying curve. To solve these issues, a subtle case analysis is required. To reduce the number of cases appearing at one time, we first discuss how to proceed in the special case of curves that have exactly two bad snippets which are adjacent trigon snippets in the tie neighbourhood. The corresponding algorithm **TwoTrigons**

- takes as input a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as a curve with exactly two bad snippets, which are adjacent trigon snippets in the tie neighbourhood and turn into the same direction, and
- outputs a curve that is homotopic to the input and contains at most one bad snippet. If this is a bigon snippet, then the reduced corner length has decreased by at least two.

For a formal statement of the algorithm **TwoTrigons** we refer the reader to Algorithm 6.4.1.

Algorithm 6.4.1: TwoTrigons - Homotoping a curve with exactly two bad snippets which must be adjacent trigon snippets in the tie neighbourhood that turn into the same direction into an input for `TrigCurve` or into a curve of shorter reduced corner length containing a single bigon.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and a curve $\alpha \subset S$ that contains exactly two bad snippets $\alpha[0]$ and $\alpha[1]$. We require that $\alpha[0]$ is of type $\mathbb{S}(h, t, 3)$ or $\mathbb{S}(h, t, 1)$ and that $\alpha[1]$ is of type $\mathbb{B}(h, t)$. The two trigon snippets must turn into the same direction.

Output: A curve α' homotopic to α such that α' contains at most one bad snippet. If this unique bad snippet is a bigon snippet, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) - 2$. If it is a trigon snippet, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) + 2s + 1$.

```

1 if  $\alpha[0]$  is of type  $\mathbb{S}(h, t, 3)$  then
2    $\alpha' = \text{Hom}(\alpha', 1)$ 
3   if  $\alpha'[1]$  is in efficient position then
4     return  $\alpha'$ 
5    $\alpha' = \text{Hom}(\alpha', 1)$ 
6   if  $\alpha'[0]$  is a bigon of type  $\mathbb{S}(t, t, 0)$  then
7     return  $\text{Hom}(\alpha', 0)$ 
8    $\alpha' = (\alpha'[0] \cdot \text{TrigArc}(S, N, \alpha'[1 : ])) / \sim$ 
9   if  $\alpha'[-1]$  is a trigon of type  $\mathbb{R}(h, v)$  then
10    return  $\text{Hom}((\text{Hom}(\alpha', -1)), 0)$ 
11  if  $\alpha'[1]$  is in efficient position then
12    return  $\alpha'$ 
13 return  $\text{Hom}(\text{Hom}(\alpha', 1), 0)$ 

```

Lemma 6.4.1. *The algorithm `TwoTrigons` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time.*

Proof. By assumption on the input of the algorithm `TwoTrigons`, we are given a curve α that contains exactly two bad snippets, $\alpha[0]$ and $\alpha[1]$. We further know that $\alpha[0]$ is a bad snippet of type $\mathbb{S}(h, t, 3)$ or $\mathbb{S}(h, t, 1)$ and $\alpha[1]$ is of type $\mathbb{B}(h, t)$. Both turn into the same direction. Without loss of generality, we assume that they turn right. We remark that $\text{len}(\alpha) \geq 3$ as $\alpha[0]$ and $\alpha[1]$ are contained in different types of neighbourhood rectangles.

We begin by proving the correctness of the algorithm `TwoTrigons`. Let us first assume that $\alpha[0]$ is of type $\mathbb{S}(h, t, 3)$. Set $\alpha' = \text{Hom}(\alpha, 1)$. Since $\alpha[0]$ is of type $\mathbb{S}(h, t, 3)$ and $\alpha[0]$ and $\alpha[1]$ turn right, the boundary of the trigon cut off by $\alpha[1]$ contains a corner of the adjacent complementary region. Thus, Lemma 3.4.3 implies that $\alpha[1]$ is not weakly snippet homotopic to $\alpha'[1]$, but $\alpha[0]$ is weakly snippet homotopic to $\alpha'[0]$. We note that $\alpha'[0]$ then must be of type $\mathbb{S}(h, v, 2)$. As $\alpha[1]$ was in efficient position, we know that $\alpha'[1]$ is in efficient position or a right-turning trigon snippet of type $\mathbb{R}(h, v)$. We note that applying a local trigon homotopy to α at $\alpha[1]$ does not increase the reduced corner length of the underlying curve unless $\alpha'[1]$ is in efficient position. If $\alpha'[1]$ is in efficient position, the reduced corner length increases by at most $2s$ following Lemma 3.4.6. Thus, the algorithm either returns a curve α' with a unique bad snippet of type $\mathbb{S}(h, v, 1)$ satisfying $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) + 2s$ or proceeds by applying a local trigon homotopy to α at $\alpha'[1]$.

In the following, we set $\alpha'' = \text{Hom}(\alpha', 1)$. Lemma 3.4.10 implies that

$$\text{len}_{\text{red}}(\alpha'') = \text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha).$$

If $\text{len}(\alpha') = 2$, then α'' contains a unique bad snippet $\alpha''[0]$ of type $\mathbb{S}(t, t, 0)$ (see Figure 6.2). Applying one local homotopy to the snippet of type $\mathbb{S}(t, t, 0)$ reduces the reduced corner length by at least two according to Lemma 4.1.1 and yields a curve containing at most one bad snippet.

If $\text{len}(\alpha') > 2$, then $\alpha''[1 :]$ is an arc that contains at most one bad snippet, which must be of trigon type (see Figure 6.3). We note that this trigon can be the snippet $\alpha''[1]$. Applying the algorithm `TrigArc` to $\alpha''[1 :]$ yields an arc which is in efficient position in its inside. As a single trigon homotopy changes at most one weak snippet homotopy type of the adjacent snippets, `TrigArc` $(S, N, \alpha''[1 :])[0]$ or `TrigArc` $(S, N, \alpha''[1 :])[-1]$ are in efficient position. This implies that either `TrigArc` $(S, N, \alpha''[1 :])$ is in efficient position, or that `TrigArc` $(S, N, \alpha''[1 :])[-1]$ is a trigon inside a complementary region, or that `TrigArc` $(\alpha''[1 :])[0]$ is a trigon in a

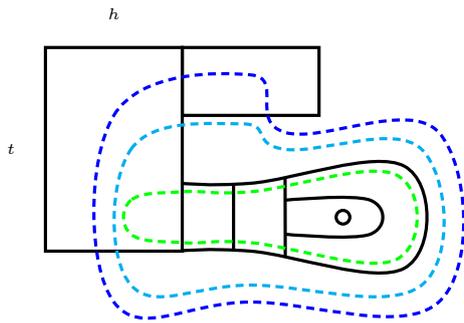


Figure 6.2: Applying the algorithm `TwoTrigons` to a curve of snippet length three.

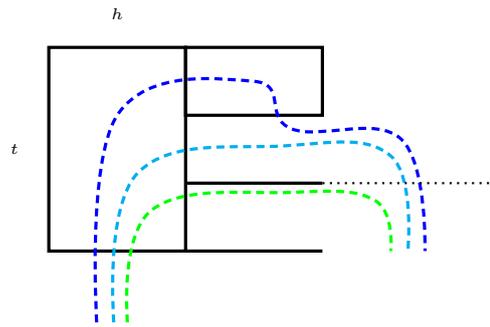


Figure 6.3: Applying two trigon homotopies on an input of `TwoTrigons` which has snippet length at least four.

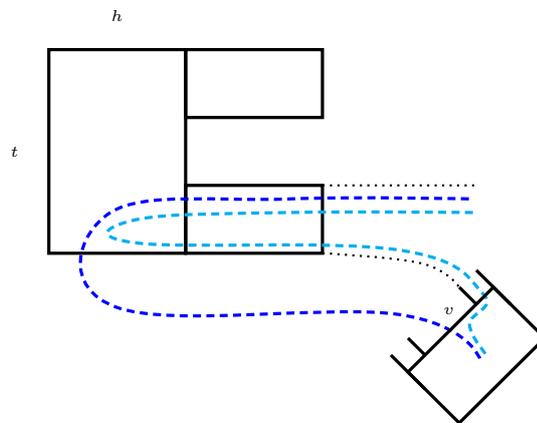


Figure 6.4: A curve with two adjacent trigons of which one lies in a complementary region and one inside a switch rectangle.

branch rectangle. We note that the reduced corner length of the arc $\alpha''[1:]$ increases under **TrigArc** only if $\text{TrigArc}(S, N, \alpha''[1:])$ is in efficient position, and then by at most $2s$. This can be seen as follows: if $\alpha''[1]$ or $\alpha''[-1]$ are the unique bad snippet of the arc $\alpha''[1:]$, then **TrigArc** simply returns $\alpha''[1:]$ and thus does not increase the reduced corner length. Else, the arc $\alpha''[1:]$ can be artificially prolonged by one snippet at the beginning and one snippet at the end to yield an arc that contains a unique trigon snippet in its inside and contains the entire arc $\alpha''[1:]$ in its inside. Thus, applying **TrigArc** to this prolonged arc increases the reduced corner length only if efficient position is achieved on the inside or if the weak snippet homotopy type of the first or last snippet is altered. As the algorithm **TrigArc** applied to the arc $\alpha''[1:]$ would have terminated in the latter two cases already, this implies that the reduced corner length of $\text{TrigArc}(S, N, \alpha''[1:])$ has increased only if $\text{TrigArc}(S, N, \alpha''[1:])$ is in efficient position.

If $\text{TrigArc}(S, N, \alpha''[1:])$ is in efficient position, the curve

$$\alpha''[0] \cdot \text{TrigArc}(S, N, \alpha''[1:])/_{\sim}$$

contains a unique trigon snippet of type $\mathbb{S}(h, t, 1)$. Its reduced corner length is bounded by $\text{len}_{\text{red}}(\alpha'') + 2s$, hence also by $\text{len}_{\text{red}}(\alpha) + 2s$.

In the second case, that is, if $\text{TrigArc}(S, N, \alpha''[1:])[-1]$ is a trigon snippet inside a complementary region, applying on trigon homotopy to

$$\text{TrigArc}(S, N, \alpha''[1:])[-1]$$

yields a curve that contains a unique bigon of type $\mathbb{S}(t, t, 0)$ (see Figure 6.4). Applying one further homotopy to the snippet $\mathbb{S}(t, t, 0)$ reduces the reduced corner length by at least two and yields a curve containing at most one bad snippet. Thus, the algorithm outputs a curve with a unique bad snippet whose reduced corner length is less than or equal to $\text{len}_{\text{red}}(\alpha) - 2$ in this case.

Else, that is if $\text{TrigArc}(S, N, \alpha''[1:])[0]$ is a trigon snippet in a branch rectangle, the first snippet of the curve $\alpha''' = \alpha''[0] \cdot \text{TrigArc}(S, N, \alpha''[1:])/_{\sim}$ must be a right-turning snippet of type $\mathbb{S}(h, t, 0)$, whereas its second snippet, the snippet $\text{TrigArc}(S, N, \alpha''[1:])[0]$ is a trigon snippet inside a branch rectangle. We note that the algorithm exists the first if-statement and the given curve is of the second possible input type for the algorithm **TwoTrigons**. We recall that $\text{len}_{\text{red}}(\alpha''') \leq \text{len}_{\text{red}}(\alpha)$. Then $\text{Hom}(\alpha''', 1)$ contains a unique bad snippet of type $\mathbb{S}(h, h, 0)$ and satisfies

$$\text{len}_{\text{red}}(\text{Hom}(\alpha''', 1)) \leq \text{len}_{\text{red}}(\alpha''').$$

Lemma 4.1.1 implies that the curve $\alpha''' = \text{Hom}(\text{Hom}(\alpha'', 1), 0)$ contains at most one bad snippet. If it does, the curve satisfies

$$\text{len}_{\text{red}}(\alpha''') \leq \text{len}_{\text{red}}(\text{Hom}(\alpha'', 1)) - 2 \leq \text{len}_{\text{red}}(\alpha'') - 2 \leq \text{len}_{\text{red}}(\alpha) - 2.$$

Thus, the correctness of the algorithm `TwoTrigons` follows.

Lastly, let us analyse the running time of the algorithm `TwoTrigons`. Checking whether $\alpha[0]$ is of type $\mathbb{S}(h, t, 3)$ and, if applicable, replacing the subarc $\alpha[: 3]$ by the subarc $\text{Hom}(\alpha[: 3], 1)$ can be done within $O(|\chi(S)|)$ many operations. Similarly, evaluating any further if-statements in the pseudocode of the algorithm `TwoTrigons` and executing one or two local homotopies can be done within $O(|\chi(S)|)$ many operations. As $\text{len}_{\text{red}}(\alpha''[1 :]) \leq \text{len}_{\text{red}}(\alpha)$, executing the algorithm `TrigArc` on the input $(S, N, \alpha''[1 :])$ requires at most $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations. Hence, the entire algorithm terminates within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations. \square

6.5 HorBigonInCompWide-algorithm for almost efficient curves

In the following we say that a snippet of type $\mathbb{R}(h, h)$ is *wide* or *narrow* if the boundary of the bigon contains at least one point or does not contain any points of $\partial^2\mathcal{R}$ respectively. For examples of wide and narrow bigon snippets of type $\mathbb{R}(h, h)$ we refer the reader to Figure 4.10 on page 72.

We now use the results from the previous section to define the algorithm `HorBigonInCompWide` that

- takes as input a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an almost efficient curve with a wide snippet of type $\mathbb{R}(h, h)$, and
- outputs a curve that is homotopic to the input and contains at most one bad snippet. If this is a bigon snippet, then the reduced corner length has decreased by at least two.

For a formal statement of the algorithm `HorBigonInCompWide` we refer the reader to Algorithm 6.5.1.

Strictly speaking it is not necessary to differentiate between wide and narrow horizontal bigons and to present an extra algorithm which deals with wide horizontal bigons. However, it reduces the number of case distinctions that appear simultaneously in the arguments.

To simplify notation, we introduce the notion of reverse snippets and reverse arcs or curves. We recall that snippets and snippet-decomposed arcs and curves carry an orientation induced by the standard parametrization of the interval $[0, 1]$.

Definition 6.5.1. Suppose that $a \subset R \in \mathcal{R}$ is a snippet. By $\text{rev}(a)$ we denote the snippet a with its orientation reversed. Similarly, for any snippet-decomposed arc or curve $\alpha \subset S$, we denote by $\text{rev}(\alpha)$ the arc or curve α with its orientation reversed.

Suppose that $\alpha \subset S$ is a snippet-decomposed arc or curve of length $\text{len}(\alpha) = k$. Then $\text{rev}(\alpha)[i] = \text{rev}(\alpha[k - i - 1])$.

Algorithm 6.5.1: HorBigonInCompWide - Homotoping an almost efficient curve with a wide bigon of type $\mathbb{R}(h, h)$ into an input for **TrigCurve** or into an almost efficient curve of bigon type and shorter reduced corner length.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an almost efficient curve $\alpha \subset S$ that contains a wide bigon of type $\mathbb{R}(h, h)$. We further require that $\text{len}(\alpha) > 2$.

Output : A curve α''' homotopic to α such that α''' contains at most one bad snippet. If this unique bad snippet is a bigon snippet, then $\text{len}_{\text{red}}(\alpha''') \leq \text{len}_{\text{red}}(\alpha) - 2$. If it is a trigon snippet, then $\text{len}_{\text{red}}(\alpha''') \leq \text{len}_{\text{red}}(\alpha) + 2s + 1$.

```

1 Set  $k$  such that  $\alpha[k]$  is bad
2  $\alpha' = \text{Hom}(\alpha, k)$ 
3  $\alpha'' = \text{TrigArc}(S, N, \alpha'[k : ] \cdot \alpha'[: k - 1])$ 
4 if  $\alpha''$  is in efficient position then
5 |   return  $\alpha'' \cdot \alpha'[k - 1] / \sim$ 
6 if  $\alpha''[-1]$  is of type  $\mathbb{R}(h, v)$  then
7 |    $\alpha''' = \text{Hom}(\alpha'' \cdot \alpha'[k - 1] / \sim, -2)$ 
8 |   return  $\text{AllButHorBigInCompOrBranch}(S, N, \alpha''')$ 
9 if  $\alpha'[k - 1]$  is of type  $\mathbb{S}(h, t, 1)$  or  $\mathbb{S}(h, t, 3)$  then
10 |   $\alpha''' = \alpha'[k - 1] \cdot \alpha'' / \sim$ 
11 |  return  $\text{TwoTrigons}(S, N, \alpha''')$ 
12  $\alpha''' = \text{rev}(\alpha''[0]) \cdot \text{rev}(\alpha'[k - 1]) \cdot \text{rev}(\alpha''[1 : ]) / \sim$ 
13 return  $\text{rev}(\text{TwoTrigons}(S, N, \alpha'''))$ 

```

Lemma 6.5.2. *The algorithm **HorBigInCompWide** is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time.*

Proof. By assumption on the input of the algorithm **HorBigonInCompWide**, we are given a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train

track in S , as well as an almost efficient curve α with a wide horizontal bigon snippet $\alpha[k]$ of type $\mathbb{R}(h, h)$. We can assume that $\text{len}(\alpha) > 2$. We remark that it requires at most $O(|\chi(S)|(\text{len}_{\text{red}}(\alpha) + 1))$ many operations to determine the index of this bad snippet.

We prove the correctness of the algorithm by going through its pseudocode line by line. As $\text{len}(\alpha) > 2$, the curve $\alpha' = \text{Hom}(\alpha, k)$ has two trigon snippets inside the tie neighbourhood which turn the same way. Lemma 4.1.7 implies that

$$\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha).$$

Since $\text{Hom}(\alpha[k-1 : k+2], 1)[1 : -1]$ is in efficient position, we further know that one of these trigon snippets is the snippet $\alpha'[k-1]$, whereas the other trigon snippet might be adjacent to $\alpha'[k-1]$ or separated from it by a number of carried snippets of α' (see page 72, Figure 4.10). The previous section provides an algorithm to handle curves that contain two adjacent trigon snippets inside the tie neighbourhood. Thus, the next step in the algorithm aims to homotope the curve into such a form. As $\alpha'[k-1]$ is one of the two bad snippets of trigon type of α' , the arc $\alpha'[k:] \cdot \alpha':[k-1]$ contains exactly one bad snippet. This is of trigon type. We note that the inside of $\alpha'[k:] \cdot \alpha':[k-1]$ might already be in efficient position. In any case, the arc $\alpha'' = \text{TrigArc}(S, N, \alpha'[k:] \cdot \alpha':[k-1])$ then is in efficient position in its inside. If the entire arc α'' is in efficient position, then $\alpha'' \cdot \alpha'[k-1]/\sim$ contains a unique bad snippet of trigon type. Else, either the snippet $\alpha''[0]$ or the snippet $\alpha''[-1]$ are a bad snippet of trigon type. We note that in this case, the reduced corner length of α'' equals the reduced corner length of $\alpha'[k:] \cdot \alpha':[k-1]$ following previously applied arguments as in the proof of Lemma 6.4.1.

If $\alpha''[-1]$ is a trigon, it must be a trigon of type $\mathbb{R}(h, v)$ (see page 94, Figure 6.4). This follows from the fact that the snippet $\alpha'[k-2]$ intersects the trigon $\alpha'[k-1]$ along $\partial_h N$. Hence, $\alpha'[k-2]$ and therefore $\alpha''[-1]$ lie inside a complementary region. Applying one homotopy to the snippet $\alpha''[-1]$ of type $\mathbb{R}(h, v)$ does not increase the reduced corner length but yields a curve with a unique snippet of bigon type that lies inside the tie neighbourhood. This implies that

$$\text{AllButHorBigInCompOrBranch}(S, N, \text{Hom}(\alpha'' \cdot \alpha'[k-1]/\sim, -2))$$

is a curve that contains at most one bad snippet. If this bad snippet is a bigon, its reduced corner length has decreased by at least two. If the curve contains a unique trigon, the reduced corner length has increased by at most $2s + 1$.

If $\alpha''[0]$ is a trigon, it must be a trigon inside the tie neighbourhood which

is turning into the same direction as $\alpha'[k-1]$. Hence, we can use the algorithm `TwoTrigons` to obtain the desired results. Due to our special set-up of the algorithm `TwoTrigons` we have to make sure to feed it a curve whose first snippet lies inside a switch rectangle. Hence, if $\alpha'[k-1]$ lies inside a switch rectangle, we can immediately apply the algorithm `TwoTrigons` and `TwoTrigons`($S, N, \alpha'[k-1] \cdot \alpha''$) yields a curve whose reduced corner length has decreased (respectively increased) by at least two (respectively at most $2s+1$) if it contains a bigon (respectively trigon). If, on the other hand, the snippet $\alpha'[k-1]$ lies inside a branch rectangle, $\alpha''[0]$ is a snippet inside a switch rectangle. Thus, the curve $\alpha''' = \text{rev}(\alpha''[0]) \cdot \text{rev}(\alpha'[k-1]) \cdot \text{rev}(\alpha''[1:]) / \sim$ is a valid input for `TwoTrigons`, and `rev`(`TwoTrigons`(S, N, α''')) is homotopic to α and of the required form. This concludes the proof of the correctness of the algorithm `HorBigInComplWide`.

To analyse the running time of the algorithm `HorBigInComplWide` on an input (S, N, α) , we note that any evaluation of the if-statements and algorithms can be executed within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\tilde{\alpha}) + 1))$ many operations, where $\tilde{\alpha}$ is the relevant version of the curve homotopic to α . As $\alpha'[k-1]$ is a bad snippet, we know that

$$\text{len}_{\text{red}}(\alpha') = \text{len}_{\text{corn}}(\alpha'[k-1]) + \text{len}_{\text{red}}(\alpha'[k:] \cdot \alpha'[:k-1]).$$

Together with the fact that the reduced corner length of $\tilde{\alpha}$ never exceeds the reduced corner length of α unless efficient position is achieved or a unique bad snippet of trigon type is created, this implies that the entire algorithm terminates within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations. \square

6.6 HorBigonInComp-algorithm for almost efficient curves

Building on the results from the previous section, we now present the algorithm `HorBigonInComp` that handles almost efficient curves with a wide or narrow horizontal bigon. For a formal statement of `HorBigonInComp` we refer the reader to Algorithm 6.6.1.

Lemma 6.6.1. *The algorithm `HorBigInCompl` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time.*

Proof. By assumption on the input of the algorithm `HorBigonInComp`, we are given a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an almost efficient curve $\alpha \subset S$ with a bad snippet $\alpha[k]$ of type $\mathbb{R}(h, h)$. We remark that it requires at most $O(|\chi(S)|(\text{len}_{\text{red}}(\alpha) + 1))$ many operations to determine the index of this bad snippet.

Algorithm 6.6.1: HorBigonInComp - Homotoping an almost efficient curve with a snippet of type $\mathbb{R}(h, h)$ into an input for **TrigCurve** or into an almost efficient curve of bigon type and shorter reduced corner length.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an almost efficient curve $\alpha \subset S$ that contains a snippet of type $\mathbb{R}(h, h)$.

Output : A curve α' homotopic to α such that α' contains at most one bad snippet. If this unique bad snippet is a bigon snippet, then $\text{len}_{\text{red}}(\alpha') < \text{len}_{\text{red}}(\alpha)$. If it is a trigon snippet, then $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) + 2s + 1$.

```

1 Set  $k$  such that  $\alpha[k]$  is bad
2 if  $\text{len}(\alpha) = 2$  then
3   | return AllButHorBigInCompOrBranch( $S, N, \text{Hom}(\alpha, k)$ )
4 if  $\text{Hom}(\alpha, k)[k - 1]$  is a bigon of type  $\mathbb{B}(h, h)$  or  $\mathbb{S}(h, h, 0)$  then
5   | return  $\text{Hom}(\alpha, k)$ 
6 return HorBigInComplWide( $S, N, \alpha$ )

```

If $\text{len}(\alpha) = 2$, then $\alpha[k - 1]$ must be a tie of branch or switch rectangle (see page 72, Figure 4.9). We see that $\text{Hom}(\alpha, k)[k - 1]$ is a bigon inside the tie neighbourhood that does not intersect $\partial_h N$. Thus, it is a valid input for the algorithm **AllButHorBigInCompOrBranch**. Since $\text{len}_{\text{red}}(\alpha) = \text{len}_{\text{red}}(\text{Hom}(\alpha, k))$, we know that

$$\text{len}_{\text{red}}(\text{AllButHorBigInCompOrBranch}(S, N, \text{Hom}(\alpha, k))) \leq \text{len}_{\text{red}}(\alpha) - 2$$

and that the output curve contains a bigon.

If $\text{len}(\alpha) > 2$ and $\alpha[k]$ is a narrow bigon, then $\alpha[k - 1]$ and $\alpha[k + 1]$ are both ties of the same branch or switch rectangle. Hence, $\text{Hom}(\alpha, k)$ contains a unique bigon of type $\mathbb{B}(h, h)$ or $\mathbb{S}(h, h, 0)$ and satisfies $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 1$ by Lemma 4.1.7.

Else, we have that $\text{len}(\alpha) > 2$ and $\alpha[k]$ is a wide bigon. Thus, **HorBigInComplWide**(S, N, α) is of the desired form following the results in the previous section.

As all algorithms employed terminate within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time, the algorithm **HorBigInCompl** terminates within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations. \square

6.7 HorBigonInBranch-algorithm for almost efficient curves

In this section we see that the discussion about almost efficient curves containing a snippet of type $\mathbb{B}(h, h)$ naturally reduces to the case of almost efficient curve containing a snippet of type $\mathbb{R}(h, h)$. For a formal statement of the corresponding algorithm `HorBigonInBranch` we refer the reader to Algorithm 6.7.1.

Algorithm 6.7.1: `HorBigonInBranch` - Homotoping an almost efficient curve with a bigon snippet of type $\mathbb{B}(h, h)$ into an input for `TrigCurve` or into a curve of shorter reduced corner length containing a single bigon snippet.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an almost efficient curve $\alpha \subset S$ that contains a snippet of type $\mathbb{B}(h, h)$.

Output: A curve α'' homotopic to α such that α'' contains at most one bad snippet. If this bad snippet is a bigon snippet, then $\text{len}_{\text{red}}(\alpha'') < \text{len}_{\text{red}}(\alpha)$. If it is a trigon snippet, then $\text{len}_{\text{red}}(\alpha'') \leq \text{len}_{\text{red}}(\alpha) + 2s + 1$.

```

1 Set  $k$  such that  $\alpha[k]$  is bad
2 if  $\text{len}(\alpha) = 2$  or  $\text{len}_{\text{corn}}(\alpha[k - 1]) = 2s$  or  $\text{len}_{\text{corn}}(\alpha[k + 1]) = 2s$  then
3   | return  $\text{Hom}(\alpha, k)$ 
4 if  $\alpha[k - 1]$  and  $\alpha[k + 1]$  are vertical duals that turn the same way then
5   | return  $\text{Hom}(\alpha, k)$ 
6 if  $\text{len}_{\text{corn}}(\alpha[k - 1]) > 1$  and  $\text{len}_{\text{corn}}(\alpha[k + 1]) > 1$  then
7   | return  $\text{Hom}(\alpha, k)$ 
8 if  $\text{len}_{\text{corn}}(\alpha[k - 1]) = 1$  and  $\text{len}_{\text{corn}}(\alpha[k + 1]) = 1$  then
9   | if  $\alpha[k - 3 : k]$  or  $\alpha[k + 1 : k + 4]$  is not a blocker then
10  |   | return  $\text{Hom}(\alpha, k)$ 
11  |    $\alpha' = \text{Hom}(\text{Hom}(\alpha, k), k - 1)$ 
12  |   return  $\text{Hom}(\alpha', k - 2)$ 
13 return  $\text{HorBigonInComp}(S, N, \text{Hom}(\alpha, k))$ 

```

Lemma 6.7.1. *The algorithm `HorBigInBranch` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time.*

Proof. By assumption on the input of the algorithm `HorBigInBranch`, we are given a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an almost efficient curve α that contains a bigon snippet $\alpha[k]$ of type $\mathbb{B}(h, h)$. We remark that it requires at most $O(|\chi(S)|(\text{len}_{\text{red}}(\alpha) + 1))$ many operations to determine the index of this bad snippet.

The structure of the algorithm and of our proof is arranged according to the different types of the snippets $\alpha[k - 1]$ and $\alpha[k + 1]$. We recall that all snippets

of α but $\alpha[k]$ are in efficient position. The following list provides a complete case distinction.

- $\text{len}(\alpha) = 2$.
- $\text{len}_{\text{corn}}(\alpha[k-1])$ or $\text{len}_{\text{corn}}(\alpha[k+1])$ equal $2s$.
- $\alpha[k-1]$ and $\alpha[k+1]$ are vertical duals. In this case they are
 - vertical duals turning into the same direction (see Figure 6.5) or
 - vertical duals turning into different direction (see Figures 6.6-6.8).

If $\alpha[k-1]$ and $\alpha[k+1]$ are vertical duals turning into different directions, we can further distinguish the following disjoint cases.

- $\text{len}_{\text{corn}}(\alpha[k-1]) > 1$ and $\text{len}_{\text{corn}}(\alpha[k+1]) > 1$ (see Figure 6.6).
- $\text{len}_{\text{corn}}(\alpha[k-1]) = 1$ and $\text{len}_{\text{corn}}(\alpha[k+1]) = 1$ (see Figure 6.7).
- Exactly one of the snippets $\alpha[k-1]$ and $\alpha[k+1]$ has a corner length greater than one (see Figure 6.8).

We note that the corner length of any vertical dual snippet must be greater than zero as vertical duals are in efficient position. We go through each of these cases and the corresponding if-statements of the pseudocode step by step. We remark that each of the if-statements concludes with a return-command. Thus, earlier tested properties of the curve can be assumed to be false.

If $\text{len}(\alpha) = 2$, then $\alpha[k-1]$ is an efficient snippet inside a peripheral complementary region winding non-trivially around a boundary component. Hence, $\text{Hom}(\alpha, k)$ is a peripheral curve of snippet length one.

If $\text{len}_{\text{corn}}(\alpha[k-1]) = 2s$ or $\text{len}_{\text{corn}}(\alpha[k+1]) = 2s$, then $\alpha[k-1]$ or $\alpha[k+1]$ are in efficient position but not part of any blockers of the curve α . We recall that any snippet in efficient position has non-zero corner length. Lemma 4.1.9 implies that $\text{len}(\text{Hom}(\alpha[k-1 : k+2], 1)) = 1$. Thus, we see that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha[k-1 : k+2], 1)) \leq 2s < 2s + 1 + 1 \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]).$$

Unless $\text{len}_{\text{corn}}(\alpha[k-1]) = 1$ or $\text{len}_{\text{corn}}(\alpha[k+1]) = 1$, this implies that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k))$ is strictly smaller than $\text{len}_{\text{red}}(\alpha)$ as the number of blockers decreases by at most one under the homotopy. Thus, of whatever type the snippet $\text{Hom}(\alpha[k-1 : k+2], 1)$ is in this case, the curve $\text{Hom}(\alpha, k)$ is a valid output for the algorithm `HorBigonInBranch`. On the other hand, if $\text{len}_{\text{corn}}(\alpha[k-1]) = 1$ or $\text{len}_{\text{corn}}(\alpha[k+1]) = 1$, we see that

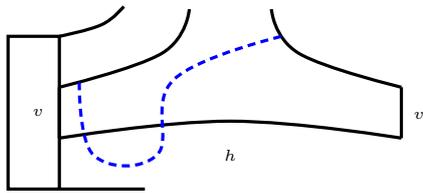


Figure 6.5: A trigon snippet of type $\mathbb{B}(h, h)$ that is adjacent to two vertical duals that turn the same way.

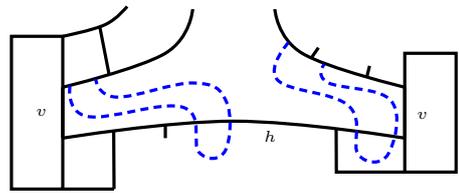


Figure 6.6: Examples of trigon snippets of type $\mathbb{B}(h, h)$ that are adjacent to two vertical duals which turn into different directions and both have corner length greater than one.

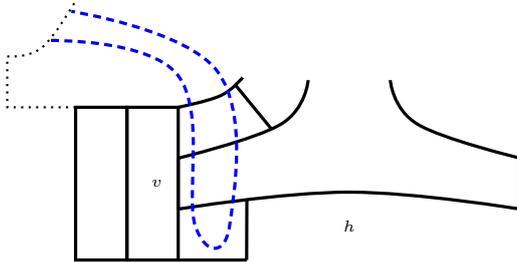


Figure 6.7: A trigon snippet of type $\mathbb{B}(h, h)$ that is adjacent to two vertical duals which turn into different directions and which both have corner length one.

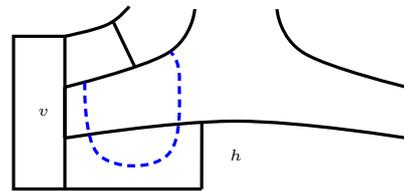


Figure 6.8: A trigon snippet of type $\mathbb{B}(h, h)$ that is adjacent to two vertical duals which turn into different directions and of which exactly one has corner length one.

$\text{Hom}(\alpha[k-1 : k+2], 1)$ must be a trigon or in efficient position following an index argument, and

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha).$$

Thus, the curve $\text{Hom}(\alpha, k)$ is a valid output for the algorithm `HorBigonInBranch` in this case, too.

If $\alpha[k-1]$ and $\alpha[k+1]$ are vertical duals turning into the same direction, and index argument shows that $\text{Hom}(\alpha, k)$ is in efficient position (see Figure 6.5).

From this point onwards we can assume that $\alpha[k-1]$ and $\alpha[k+1]$ are vertical duals that turn into different directions (see Figures 6.6-6.8). The points $\alpha[k-1](0)$ and $\alpha[k+1](1)$ lie on the same component of $\partial_h N$. Thus, $\text{Hom}(\alpha, k)$ is an almost efficient curve containing a snippet of type $\mathbb{R}(h, h)$. Set $\beta = \text{Hom}(\alpha[k-1 : k+2], 1)$. For the remainder of this proof we assume that $\text{len}_{\text{corn}}(\alpha[k-1]) \leq \text{len}_{\text{corn}}(\alpha[k+1])$. This implies that

$$\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - \text{len}_{\text{corn}}(\alpha[k-1]) - 1 - 1.$$

This can be seen as follows: Let us denote the boundary of the index-zero region cut off by $\alpha[k+1]$ by $B_{\alpha[k+1]}$ and the boundary of the region cut off by the bigon snippet β by B_β . Then $(B_{\alpha[k+1]} \cap \partial \mathcal{R}) - \partial_v N$ consists of two components, of which one contains the point $\alpha[k-1](0)$ as well as the set $B_\beta \cap \partial \mathcal{R}$. We see that

$$\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k+1]) - 1.$$

As $\text{len}_{\text{corn}}(\alpha[k]) = 1$, this implies that

$$\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - \text{len}_{\text{corn}}(\alpha[k-1]) - 2.$$

If $\text{len}_{\text{corn}}(\alpha[k+1]) > 1$, then one of the components of $(B_{\alpha[k+1]} \cap \partial \mathcal{R}) - \partial_v N$ must contain a component of the horizontal boundary of a branch rectangle. If this is the component that does not contain $B_\beta \cap \partial \mathcal{R}$, then

$$\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k+1]) - 2.$$

This implies that

$$\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - \text{len}_{\text{corn}}(\alpha[k-1]) - 3.$$

Similarly, if this component of the horizontal boundary of a branch rectangle lies in the component of $B_\beta \cap \partial \mathcal{R}$ that contains $\alpha[k-1](0)$, then $B_\beta \cap \partial \mathcal{R}$, which is

confined by the points $\alpha[k+1](1)$ and $\alpha[k-1](0)$, contains one less component of the horizontal boundary of a branch rectangle, namely the one that intersects $\partial_v N$. This, too, implies that $\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k+1]) - 2$, thus

$$\text{len}_{\text{corn}}(\beta) \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - \text{len}_{\text{corn}}(\alpha[k-1]) - 3.$$

If $\text{len}_{\text{corn}}(\alpha[k-1]) > 1$ and $\text{len}_{\text{corn}}(\alpha[k+1]) > 1$, this discussion implies that $\text{len}_{\text{corn}}(\beta) < \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - 4$ (see Figure 6.6). As the homotopy decreases the number of blockers by at most two, we see that $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) < \text{len}_{\text{red}}(\alpha)$.

If $\text{len}_{\text{corn}}(\alpha[k-1]) = 1$ and $\text{len}_{\text{corn}}(\alpha[k+1]) > 1$, the previous discussion implies that $\text{len}_{\text{corn}}(\text{Hom}(\alpha[k-1 : k+2], 1)) \leq \text{len}_{\text{corn}}(\alpha[k-1 : k+2]) - 4$. Since at most two blockers are affected by the homotopy, we obtain that

$$\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha),$$

where $\text{Hom}(\alpha, k)$ is a curve containing a wide horizontal bigon. Lemma 6.6.1 then implies that $\text{HorBigonInComp}(S, N, \text{Hom}(\alpha, k))$ is a curve which contains at most one bad snippet. If this is a bigon, then the reduced corner length decreased by at least one compared to the reduced corner length of the original curve α .

Lastly, let us assume that $\text{len}_{\text{corn}}(\alpha[k-1]) = 1$ and $\text{len}_{\text{corn}}(\alpha[k+1]) = 1$ (see Figure 6.7). Then β is a narrow bigon of type $\mathcal{R}(h, h)$ whose boundary points lie in the same component of the horizontal boundary of a branch rectangle. We note that

$$\text{len}_{\text{corn}}(\text{Hom}(\alpha, k)) = \text{len}_{\text{corn}}(\alpha) - 3.$$

Hence, if at most one of the subarcs $\alpha[k-3 : k]$ or $\alpha[k+1 : k+4]$ was a blocker, then $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) - 1$ and $\text{Hom}(\alpha, k)$ is a valid output for HorBigonInBranch . Else, that is if the subarcs $\alpha[k-3 : k]$ and $\alpha[k+1 : k+4]$ are both blockers, then $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha) + 1$. As $\alpha[k-2]$ and $\alpha[k+2]$ are ties of the same branch rectangle, applying one further local homotopy does not reduce the number of blockers of the underlying curve but reduces the corner length by one. This implies that

$$\text{len}_{\text{red}}(\text{Hom}(\text{Hom}(\alpha, k), k-1)) \leq \text{len}_{\text{red}}(\alpha).$$

Since $\alpha[k-1]$ and $\alpha[k+1]$ were part of some blocker, we know that $\alpha[k-3]$ and $\alpha[k+3]$ are not the same snippet as they are vertical duals turning in different directions. Moreover, Lemma 3.2.9 implies that $\text{len}_{\text{corn}}(\alpha[k-3]) \geq 5$ and $\text{len}_{\text{corn}}(\alpha[k+3]) \geq 5$.

Set $\alpha' = \text{Hom}(\text{Hom}(\text{Hom}(\alpha, k), k - 1), k - 2)$. Then

$$\text{len}_{\text{red}}(\alpha') = \text{len}_{\text{red}}(\alpha'[k - 2 :] \cdot \alpha'[: k - 3]) + \text{len}_{\text{corn}}(\alpha'[k - 3]).$$

Reasoning as in the second to last paragraph we see that

$$\begin{aligned} \text{len}_{\text{corn}}(\alpha'[k - 3]) &\leq \text{len}_{\text{corn}}(\text{Hom}(\text{Hom}(\alpha, k), k - 1)[k - 3 : k]) - (5 + 1 + 5) \\ &\leq \text{len}_{\text{corn}}(\text{Hom}(\alpha, k)[k - 3 : k + 2]) - 1 - 11 \\ &\leq \text{len}_{\text{corn}}(\alpha[k - 3 : k + 4]) - 3 - 12. \end{aligned}$$

As $\alpha'[k - 2 :] \cdot \alpha'[: k - 3] = \alpha[k + 4 :] \cdot \alpha[: k - 3]$, we know that

$$\text{len}_{\text{red}}(\alpha'[k - 2 :] \cdot \alpha'[: k - 3]) = \text{len}_{\text{red}}(\alpha[k + 4 :] \cdot \alpha[: k - 3]).$$

Therefore,

$$\begin{aligned} \text{len}_{\text{red}}(\alpha') &\leq \text{len}_{\text{red}}(\alpha[k + 4 :] \cdot \alpha[: k - 3]) + \text{len}_{\text{corn}}(\alpha[k - 3 : k + 4]) - 15 \\ &\leq \text{len}_{\text{red}}(\alpha[k + 4 :] \cdot \alpha[: k - 3]) + \text{len}_{\text{red}}(\alpha[k - 3 : k + 4]) + 4 - 15 \\ &\leq \text{len}_{\text{red}}(\alpha) + 4 + 4 - 15 \\ &\leq \text{len}_{\text{red}}(\alpha) - 7. \end{aligned}$$

Thus, we see that $\alpha' = \text{Hom}(\text{Hom}(\text{Hom}(\alpha, k), k - 1), k - 2)$ contains a unique bad snippet of type $\mathbb{R}(h, h)$ and satisfies $\text{len}_{\text{red}}(\alpha') \leq \text{len}_{\text{red}}(\alpha) - 7$. Therefore, it is a valid output for the algorithm `HorBigonInBranch`. This completes the proof of the correctness of the algorithm `HorBigonInBranch`.

To analyse the running time of the algorithm `HorBigonInBranch` on an input (S, N, α) , we note that within $O(|\chi(S)| \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time all if-statements in the pseudocode can be checked and all single local homotopies can be executed. If $\text{len}_{\text{corn}}(\alpha[k - 1]) = 1$ and $\text{len}_{\text{corn}}(\alpha[k + 1]) > 1$, then $\text{len}_{\text{red}}(\text{Hom}(\alpha, k)) \leq \text{len}_{\text{red}}(\alpha)$. Therefore, `HorBigonInComp` $(S, N, \text{Hom}(\alpha, k))$ terminates within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operation. Thus, on input α , the entire algorithm `HorBigonInBranch` halts in $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time. \square

6.8 The algorithm `SingleBadSnippet`

Combining the algorithms from the previous sections, we can finally present an algorithm that achieves efficient position for an almost efficient curve or shortens it until it consists of a single snippet.

Algorithm 6.8.1: SingleBadSnippet - Homotoping a curve with at most one bad snippet into efficient position or shortening it to consist of a single snippet.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and a curve $\alpha \subset S$ that contains at most one bad snippet.

Output : A curve α' homotopic to α such that α' is in efficient position or consists of a single snippet.

```

1  $\alpha' = \alpha$ 
2 while  $\text{len}(\alpha') > 1$  do
3   if There is a bad snippet  $\alpha[k]$  of type  $\mathbb{B}(t, t)$ ,  $\mathbb{S}(h, h, 0)$ ,  $\mathbb{S}(t, t, 0)$ ,  $\mathbb{S}(v, v, 0)$ ,  $\mathbb{R}(v, v)$ ,  $\mathbb{S}(t, t, 2)$ , or  $\mathbb{S}(t, v, 1)$  then
4      $\alpha' = \text{AllButHorBigInCompOrBranch}(S, N, \alpha')$ 
5   else if There is a bad snippet  $\alpha[k]$  of type  $\mathbb{R}(h, h)$  then
6      $\alpha' = \text{HorBigInComp}(S, N, \alpha')$ 
7   else if There is a bad snippet  $\alpha[k]$  of type  $\mathbb{B}(h, h)$  then
8      $\alpha' = \text{HorBigInBranch}(S, N, \alpha')$ 
9   if There is a bad snippet  $\alpha[k]$  that is a trigon then
10    return  $\text{TrigCurve}(S, N, \alpha')$ 
11  if  $\alpha'$  is in efficient position then
12    return  $\alpha'$ 
13 return  $\alpha'$ 

```

Lemma 6.8.1. *The algorithm `SingleBadSnippet` is correct. On an input (S, N, α) , the algorithm halts in $O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha)^2 + 1))$ time.*

Proof. By assumption on the input of the algorithm `SingleBadSnippet`, we are given a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , as well as an almost efficient curve α . Let us first prove the correctness of the algorithm `SingleBadSnippet`. If $\text{len}(\alpha) > 1$, then the unique bad snippet must be of bigon or trigon type following the classification result of Lemma 2.6.5. If it is a bigon snippet, then exactly one of the algorithms `AllButHorBigInCompOrBranch`, `HorBigInComp`, or `HorBigInBranch` will be executed once. The resulting curve either contains a single bigon snippet and has a smaller reduced corner length than α , or contains a unique trigon snippet, is in efficient position, or of snippet length one. We furthermore know that the reduced corner length of the curve increases by at most $2s + 1$ if the curve contains a trigon snippet. Thus, if the curve does not contain a bigon snippet, the algorithm terminates within $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 2s + 2))$ further operations following Corollary 3.6.2. Else, the algorithm enters the next iteration of the while-loop. Since the reduced corner length of the underlying curves is decreasing by at least one during every iteration, there can be at most $\text{len}_{\text{red}}(\alpha) + 1$ many iterations of the while-loop before the algorithm terminates and outputs a curve that is in efficient position or consists of a single snippet only. This proves the correctness of the algorithm `SingleBadSnippet`.

To analyse the running time of the algorithm `SingleBadSnippet` on an input (S, N, α) we first remark that it requires at most $O(|\chi(S)|(\text{len}_{\text{red}}(\alpha) + 1))$ many operations to determine the index of the unique bad snippet of α . We further recall that the reduced corner length of the underlying curve decreases by at least one after every iteration of the while-loop. Thus, the while-loop is executed at most $\text{len}_{\text{red}}(\alpha) + 1$ many times. The snippet length of the underlying curve when evaluation the if-statements is bounded by $\text{len}_{\text{red}}(\alpha) + 1$ as the curve has at most one bad snippet. Hence, searching for the single bad snippet takes $O(|\chi(S)| \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time. Executing a single bigon routine requires $O(\chi(S)^2 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ time. Thus, every iteration of the while-loop requires at most $O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha) + 1))$ many operations. This implies that the entire algorithm `SingleBadSnippet` halts in $O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha)^2 + 1))$ time. \square

Chapter 7

Polynomial-time efficient position

In this chapter we present the very last algorithm of this thesis, the algorithm `EfficientPosition`. As input, `EfficientPosition` takes a surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an arc or curve α , and outputs an arc or curve α' homotopic to α such that α' is in efficient position or satisfies $\text{len}(\alpha') = 1$. In the latter case, previous results imply that α must be inessential or peripheral. For a formal statement of `EfficientPosition` we refer the reader to Algorithm 7.0.1. We will see that `EfficientPosition` halts in polynomial time and thus is sufficient to provide a proof for Theorem 7.0.2.

Lemma 7.0.1. *The algorithm `EfficientPosition` is correct. On an input (S, N, α) , the algorithm halts in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time.*

Proof. The correctness of the algorithm follows from the correctness of its subroutines and Corollary 5.1.2.

We now discuss the running time of the algorithm `EfficientPosition`. If the input α is an arc, then the bounds on the running time follow from Lemma 5.1.1. Thus, suppose that the input $\alpha \subset S$ is a curve. Lemma 5.1.1 implies that the algorithm `ReduceToTwoBadSnippets` terminates in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time on input (S, N, α) . We set $\alpha' = \text{ReduceToTwoBadSnippets}(S, N, \alpha)$. Lemma 5.1.1 further implies that

$$\text{len}_{\text{red}}(\alpha') \leq (\text{len}(\alpha) - 1) \cdot 6s.$$

Corollary 5.2.2 states that the algorithm `ReduceToOneBadSnippet` terminates within $O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha') + 1))$ time on input (S, N, α') . Thus, the algorithm `ReduceToOneBadSnippet` terminates within $O((\chi(S))^4 \cdot (\text{len}(\alpha) + 1))$ time on input

Algorithm 7.0.1: EfficientPosition - Achieving efficient position for essential and non-peripheral arcs and curves or shortening inessential or peripheral arcs and curves so that they consist of a single snippet.

Input : A surface S of positive complexity, a tie neighbourhood $N \subset S$ of a large train track in S , and an arc or curve $\alpha \subset S$.

Output: An arc or curve α' homotopic, relative to endpoints if applicable, to α such that α' is in efficient position or consists of a single snippet only.

```

1  $\alpha' = \alpha$ 
2 if  $\alpha'$  is an arc then
3   | return ReduceToTwoBadSnippets( $S, N, \alpha'$ )
4 if  $\text{len}(\alpha') > 2$  then
5   |  $\alpha' = \text{ReduceToTwoBadSnippets}(S, N, \alpha')$ 
6 if  $\text{len}(\alpha') > 1$  then
7   |  $\alpha' = \text{ReduceToOneBadSnippet}(S, N, \alpha')$ 
8   |  $\alpha' = \text{SingleBadSnippet}(S, N, \alpha')$ 
9 return  $\alpha'$ 

```

(S, N, α') .

Set $\alpha'' = \text{ReduceToOneBadSnippet}(\alpha')$. Lemma 5.2.1 implies that

$$\text{len}_{\text{red}}(\alpha'') \leq (\text{len}(\alpha) - 1) \cdot 6s + 10s \leq (\text{len}(\alpha) + 1) \cdot 6s.$$

Lastly, Lemma 6.8.1 implies that the algorithm **SingleBadSnippet** terminates within $O(|\chi(S)|^3 \cdot (\text{len}_{\text{red}}(\alpha'')^2 + 1))$ time on input (S, N, α'') . As $\text{len}_{\text{red}}(\alpha'') \leq (\text{len}(\alpha) + 1) \cdot 6s$, we see that the algorithm **SingleBadSnippet** terminates within $O(|\chi(S)|^3 \cdot (\text{len}(\alpha)^2 + 1))$ time on input (S, N, α'') . Adding the running times of the three main subroutines, this implies that the algorithm **EfficientPosition** terminates in $O(\chi(S)^4 \cdot (\text{len}(\alpha)^2 + 1))$ time. Since $\text{len}(\alpha) \geq 1$, we know that $\text{len}(\alpha)^2 + 1 \leq 2\text{len}(\alpha)^2$. Thus, **EfficientPosition** terminates in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time. \square

Theorem 7.0.2. *There is an algorithm that takes as input*

- a surface $S = S_{g,b}$ satisfying $\xi(S) = 3g - 3 + b \geq 1$,
- a tie neighbourhood $N = N(\tau)$ of a large train track τ in S , and
- a properly immersed arc or curve α given via its snippet decomposition with respect to N ,

and outputs an arc or curve α' homotopic to α , relative to endpoints in the case of arcs, such that

- α' is in efficient position with respect to N or
- α' has snippet length one.

This algorithm terminates in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time. Moreover, the output α' has snippet length one if and only if α is inessential or peripheral. If α is peripheral, the boundary component that α is homotopic to, as well as the corresponding power, can be read off from the snippet-decomposition of α' .

Proof. Lemma 2.6.6 and Lemma 2.6.7 imply that arcs and curves in efficient position are essential and non-peripheral. Further, Lemma 2.6.3 implies that properly immersed arcs and curves of snippet length one are inessential or peripheral. Thus, the existence of the desired algorithm as well as the bounds on its running time follow from Lemma 7.0.1. We recall that the cutting sequence and winding number of arcs and curves of snippet length one are sufficient to determine whether they are inessential or peripheral, and, in the latter case, to determine which power of the boundary component they are homotopic to. This concludes the proof of the theorem. \square

The mathematical content of Theorem 7.0.2 can be rephrased as follows.

Corollary 7.0.3. *Suppose that $S = S_{g,b}$ is a surface satisfying $\xi(S) = 3g - 3 + b \geq 1$. Suppose that $\tau \subset S$ is a large train track and $N = N(\tau)$ is a tie neighbourhood of τ in S . Let $\alpha \subset S$ be an essential, non-peripheral, properly immersed multiarc or multicurve given via its snippet decomposition. Then efficient position for α with respect to N exists and can be obtained in $O(\chi(S)^4 \cdot \text{len}(\alpha)^2)$ time.*

As noted previously, this is the converse to Lemmas 2.6.6 and 2.6.7.

Bibliography

- [Bel15] Mark C. Bell. *Recognising mapping classes*. PhD thesis, University of Warwick, 2015.
- [Bel18] Mark Bell. flipper (computer software). pypi.python.org/pypi/flipper, 2013–2018. Version 0.13.5.
- [BW16] Mark C. Bell and Richard C. H. Webb. Polynomial-time algorithms for the curve graph. <https://arxiv.org/abs/1609.09392>, Sep 2016.
- [CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [Deh87] Max Dehn. *Papers on group theory and topology*. Springer-Verlag, New York, 1987. Translated from the German and with introductions and an appendix by John Stillwell, With an appendix by Otto Schreier.
- [DL17] Vincent Despré and Francis Lazarus. Computing the geometric intersection number of curves. In *33rd International Symposium on Computational Geometry*, volume 77 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 35, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017.
- [EN13] Jeff Erickson and Amir Nayyeri. Tracing compressed curves in triangulated surfaces. *Discrete Comput. Geom.*, 49(4):823–863, 2013.
- [FM12] Benson Farb and Dan Margalit. *A primer on mapping class groups*, volume 49 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 2012.
- [Gué09] François Guéritaud. Deforming ideal solid tori. <https://arxiv.org/abs/0911.3067v1>, Nov 2009.

- [Ham09] Ursula Hamenstädt. Geometry of the mapping class groups. I. Boundary amenability. *Invent. Math.*, 175(3):545–609, 2009.
- [HS85] Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel J. Math.*, 51(1-2):90–120, 1985.
- [Hub06] John Hamal Hubbard. *Teichmüller theory and applications to geometry, topology, and dynamics. Vol. 1.* Matrix Editions, Ithaca, NY, 2006. Teichmüller theory, With contributions by Adrien Douady, William Dunbar, Roland Roeder, Sylvain Bonnot, David Brown, Allen Hatcher, Chris Hruska and Sudeb Mitra, With forewords by William Thurston and Clifford Earle.
- [MM99] Howard A. Masur and Yair N. Minsky. Geometry of the complex of curves. I. Hyperbolicity. *Invent. Math.*, 138(1):103–149, 1999.
- [MMS12] Howard Masur, Lee Mosher, and Saul Schleimer. On train-track splitting sequences. *Duke Math. J.*, 161(9):1613–1656, 2012.
- [Mos96] Lee Mosher. Laminations and flows transverse to finite depth foliations. <http://andromeda.rutgers.edu/~mosher/LamsAndFlows.ps.gz>, 1996.
- [Mos03] Lee Mosher. Train track expansions of measured foliations. http://andromeda.rutgers.edu/~mosher/arationality_03_02_15.pdf, 2003.
- [MS18] Joseph Maher and Saul Schleimer. The compression body graph has infinite diameter. <https://arxiv.org/abs/1803.06065>, Mar 2018.
- [Pen06] R. C. Penner. Probing mapping class groups using arcs. In *Problems on mapping class groups and related topics*, volume 74 of *Proc. Sympos. Pure Math.*, pages 97–114. Amer. Math. Soc., Providence, RI, 2006.
- [PH92] R. C. Penner and J. L. Harer. *Combinatorics of train tracks*, volume 125 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 1992.
- [Tak00] Itaru Takarajima. A combinatorial representation of curves using train tracks. *Topology Appl.*, 106(2):169–198, 2000.
- [Thu88] William P. Thurston. On the geometry and dynamics of diffeomorphisms of surfaces. *Bull. Amer. Math. Soc. (N.S.)*, 19(2):417–431, 1988.

- [Thu97] William P. Thurston. *Three-dimensional geometry and topology. Vol. 1*, volume 35 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1997. Edited by Silvio Levy.
- [Thu98] William P. Thurston. Minimal stretch maps between hyperbolic surfaces. <https://arxiv.org/abs/math/9801039>, 1998.
- [Wal13] Gregory Walters. *The Python Quick Syntax Reference*. Apress, Berkely, CA, USA, 1st edition, 2013.