Dipl.-Ing. Christian Lindorfer, BSc

A language theoretic approach to self-avoiding walks

DOCTORAL THESIS

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.rer.nat. Wolfgang Woess

Institute of Discrete Mathematics

Graz, July 2021

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date, Signature

Abstract

This thesis provides a study of self-avoiding walks on quasi-transitive graphs. The connective constant $\mu(G)$ of a graph G is the asymptotic growth rate of the number of self-avoiding walks on G starting at a given vertex. For a given graph height function mapping vertices of G to integers in a way adapted to the graph structure, a bridge is a self-avoiding walk such that the height of its vertices is bounded below by the height of the initial vertex and above by the height of the terminal vertex. If the roles of the initial and terminal vertices are reversed, we talk about reversed bridges. We show that for any graph height function, the maximum of the asymptotic growth rates of the number of bridges and the number of reversed bridges must be equal to the connective constant $\mu(G)$.

The main focus of this thesis is to apply the theory of formal languages to the study of self-avoiding walks. To this end, let G be a deterministically edge-labelled graph, that is, every (directed) edge carries a label such that any two edges starting at the same vertex have different labels. Then the set of all words which can be read along the edges of self-avoiding walks starting at o forms a language denoted by $L_{\mathrm{SAW},o}(G)$. We show that the properties of this language strongly depend on the end-structure of the graph G. It is regular if and only if all ends have size 1 and it is context-free if and only if all ends have size at most 2.

Making use of the class of multiple context-free languages, this characterisation can be extended even further. We show that $L_{\mathrm{SAW},o}(G)$ is a k-multiple context-free language if and only if the size of all ends of G is at most $2k$. Applied to Cayley graphs of finitely generated groups this says that $L_{\mathrm{SAW},o}(G)$ is multiple context-free if and only if the group is virtually free. In this setting, using our method we also show that the ordinary generating function of self-avoiding walks is algebraic and in particular, the connective constant is an algebraic number.

Acknowledgements

First and foremost, I would like to thank my supervisor Wolfgang Woess. He introduced me to the topic of this thesis and provided many interesting ideas and invaluable guidance throughout my research. On the other hand he also gave me a lot of freedom during my work and encouraged me to work on different projects independently.

Besides my supervisor I would like to express my gratitude to my colleague, friend and office mate Florian Thomas. I really enjoyed all the fruitful discussions and joint work which even continued throughout the lockdown, where we met online twice a week to discuss news and work together on various projects. During my studies he also served as a mentor and guide, and I am happy that I could learn a lot from him.

I also want to thank my very good friend and fellow student Stefan Hammer. It was very motivating and also a lot of fun to discuss problems, work on exercise sheets and study for exams together throughout my studies. He also always listened and offered me advice during the writing process of this thesis and also pointed out some mistakes in the early version of this thesis.

I am very grateful to Rögnvaldur Möller and Tullio Ceccherini-Silberstein for providing me with many helpful comments during the pre-assessment phase of this thesis.

My thanks go to all of my colleagues from the Institute of Discrete Mathematics and everyone from the DK Discrete Mathematics who provided such a friendly and inspiring environment. I really missed seeing you all during the lockdown and hope to meet you all again soon.

I thankfully acknowledge that I was able to write this thesis while being employed by Graz University of Technology, which would not have been possible without the financial support provided by FWF (Austrian Science Fund) projects P31237 and DK W1230.

Finally I wish to extend my special thanks to my parents for encouraging and supporting me in everything I achieved so far. In particular I want to thank you for the moral and financial support, which made it possible to move to Graz and allowed me to focus completely on my studies.

Contents

List of Figures

1 Introduction

Imagine you start a sightseeing tour at an intersection in a large city. Choose one street at random and follow it until reaching another intersection. By repeating this process you randomly move around the city. There is only one rule: during this tour you are not allowed to visit any location twice. In other words, your path must be self-avoiding. Clearly you might reach an intersection where each neighbour was already visited before and thus get trapped during this process. However, assume that your tour continues until you have traversed n streets and thus visited $n + 1$ intersections. One natural question arising from this process is,

"For a given number n, how many different sightseeing tours of length n are there?"

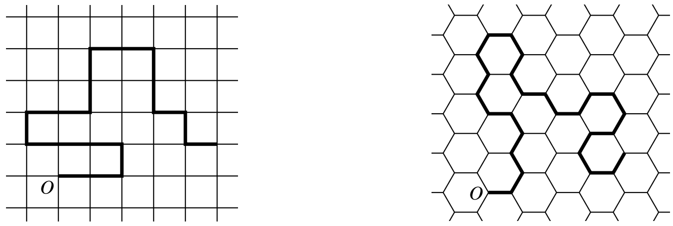
This question is simple enough, but depending on the underlying street grid it might be very difficult to obtain the number of self-avoiding sightseeing tours for large n.

Let us start by formulating the problem in mathematical terms. Start with a simple (undirected) graph G, which may (and will in most cases) contain infinitely many vertices. A walk on G is a sequence $(v_0, e_1, v_1, \dots, e_n, v_n)$ of vertices v_i and edges e_i such that e_i starts at v_{i-1} and ends at v_i for every i. It is called *self-avoiding* (or a SAW), if its vertices are pairwise different.

This notion was introduced in 1953 by the chemist Flory [16] as a model for long-chain polymer molecules and has since attracted considerable interest. Although these chains live in the continuum, in many cases a lattice approximation is good enough. The self-avoidance of the walk models the excluded-volume effect, namely that no two monomers can occupy the same position in space. The most important lattices for practical applications are the d-dimensional integer lattices $\mathbb{Z}^d$ for $d \geq 2$, where every vertex is connected to the $2d$ vertices at Euclidean distance 1. Thus a lot of research has focused on lattices, see for instance the monograph by Madras and Slade [45] and also the lecture notes by Bauerschmidt et al. [3]. Lately, self-avoiding walks have been investigated on more general classes of graphs, see for example the survey [23] of Grimmett and Li.

Denote by $c_{n,o}$ the number of self-avoiding walks with n edges starting at a fixed root vertex o. In some graphs $c_{n,o}$ can be easily calculated for every n. A very basic example is the infinite k-regular tree, where every vertex has exactly k neighbours. Independently of the choice of the starting vertex o there are k possible directions for the first step. Backtracking is not allowed, so there are at most $k-1$ choices for every consecutive step and all of them are valid because trees are cycle-free. Thus we obtain $c_{n,o} = k(k-1)^{n-1}$ for any choice of o.

While the calculations in this example are simple, they seem to be very difficult for most graphs. In these cases it makes sense to find asymptotic estimates for $c_{n,o}$.



$\mu(\mathbb{Z}^2) \approx 2.6381585$ $\qquad$ $\mu(\mathbb{H}) = \sqrt{2+\sqrt{2}}$

Figure 1.1: Examples of self-avoiding walks on the lattices $\mathbb{Z}^2$ and $\mathbb{H}$ and the connective constants of the lattices.

Hammersley [29] showed that the limit

$$\mu(G) = \lim_{n\to\infty} c_{n,o}^{1/n}$$

exists for quasi-transitive graphs, that is, graphs which allow a group action by graph automorphisms with finitely many orbits on the vertex set. Moreover, the value of $\mu(G)$ is independent of the choice of o. This number $\mu(G)$ is called the *connective constant* of the graph G. Note that by the Cauchy–Hadamard theorem, $\mu(G)$ is the reciprocal of the radius of convergence of the *SAW-generating function*

$$F_{\mathrm{SAW},o}(z) = \sum_{n=1}^{\infty} c_{n,o} z^n.$$

Explicit computation of $\mu(G)$ can however also be a very challenging task, even in seemingly harmless instances such as two-dimensional lattices (see Figure 1.1). For example, despite very precise numerical estimates, the precise value of $\mu(\mathbb{Z}^2)$ remains elusive; in fact, it is not even known whether $\mu(\mathbb{Z}^2)$ is an algebraic number.

In light of this, it is not surprising that the celebrated paper [13] by Duminil-Copin and Smirnov containing the first rigorous calculation of the connective constant of the hexagonal lattice $\mathbb{H}$ is considered a milestone in the theory. In the final part of the proof the Hammersley-Welsh method – a decomposition of walks into bridges – is applied. This method is named after the authors of [30], where it is applied to the hypercubical lattice $\mathbb{Z}^d$. The main idea is to obtain bounds for the number of self-avoiding walks by decomposing them at vertices of maximal, respectively minimal first coordinate. This process carries on until all obtained parts are bridges, which are self-avoiding walks $p = (v_0, e_1, v_1, \dots, e_n, v_n)$ on $\mathbb{Z}^d$ satisfying

$$h(v_0) < h(v_i) \le h(v_n), \quad 0 < i < n,$$

where $h(v)$ denotes the first coordinate of a vertex v. Using this decomposition, the authors showed that the number of bridges $b_{n,o}$ of length n starting at o grows with basically the same speed as the number of self-avoiding walks of length n. More precisely, they showed the existence of the bridge constant

$$\beta(\mathbb{Z}^d, h) = \lim_{n\to\infty} b_{n,o}^{1/n}$$

and its independence of the choice of o and additionally that it is equal to the connective constant $\mu(\mathbb{Z}^d)$. This result is often called the bridge theorem for the lattice $\mathbb{Z}^d$. Consequently, it is possible to determine the connective constant of $\mathbb{Z}^d$ by counting bridges instead of self-avoiding walks. Sadly, this also seems very difficult.

In Chapter 3 a generalisation of the bridge theorem to quasi-transitive graphs is discussed. The biggest motivation is the paper [22] by Grimmett and Li, where they introduced a general notion of graph height functions, which assign integers to the vertices of graph. Graph height functions have to be adapted to the graph structure according to Definition 3.1.1. They proceed to study bridges with respect to *unimodular* graph height functions and prove a bridge-theorem for quasi-transitive graphs, extending the result of Hammersley and Welsh. In this thesis we extend their results even further to obtain Theorem 3.2.2, which drops the assumption of unimodularity. The contents of Chapter 3 were first published in [43].

Let us go back to our introductory example. Suppose that after finishing your sightseeing tour you want to describe the path you have taken to your friends. A simple way to do this is by using street names. For this approach to be successful, we want every different sightseeing tour to correspond to a unique sequence of street names.

Our setting is as follows. We have a pair (G, ℓ), where G is a graph as above, and ℓ is a *labelling* assigning to every oriented edge e of G a label $\ell(e)$ of a given alphabet $\mathbf{\Sigma}$. Our assumptions are that the labelling is *deterministic,* that is, different edges with the same initial vertex have distinct labels, and that the group $\mathrm{AUT}(G, \ell)$ of all ℓ-preserving graph automorphisms of G acts quasi-transitively. The most significant class of labelled graphs are the *Cayley graphs* of finitely generated groups.

The edge-labelling is extended to walks $p = (v_0, e_1, v_1, \ldots, e_n, v_n)$ by setting

$$\ell(p) = \ell(e_1) \ldots \ell(e_n).$$

In this way, any set $\mathcal{P}$ of walks gives rise to a language $L(\mathcal{P}) = \{\ell(p) \mid p \in \mathcal{P}\}$. This identification of walks with their corresponding words allows us to study properties of $\mathcal{P}$ via properties of the corresponding language.

In the second part of this thesis we are working with the *language of self-avoiding walks* defined by

$$L_{\mathrm{SAW},o}(G) := L(\mathcal{P}_{\mathrm{SAW},o}) = \{\ell(p) \mid p \in \mathcal{P}_{\mathrm{SAW},o}\},$$

where $\mathcal{P}_{\mathrm{SAW},o}$ is the set of all self-avoiding walks of length at least 1 on G starting at o.

In the Chomsky-hierarchy of formal languages, the first basic class consists of the *regular languages,* which are accepted by a finite state automaton, or equivalently, generated by a right-linear grammar. The second class consists of the *context-free* languages (CFLs), which are accepted by a pushdown automaton, respectively, generated by a context free grammar (CFG).

Our main result in Chapter 4 is a complete characterisation of all quasi-transitive deterministically edge-labelled graphs having a regular or context-free language of self-avoiding walks. The result is joint work with Woess and has been published in [44].

An *end* of a graph G is an equivalence class of rays (one-way infinite paths) in G, where two rays are equivalent if and only if G contains a third ray intersecting both of them infinitely often. The *size* of an end is the maximum number of pairwise disjoint rays it contains; ends are *thin* if they have finite size, otherwise they are *thick*.

Theorem 1.0.1. *Let (X, ℓ) be a quasi-transitive deterministically labelled graph, connected and locally finite. For any choice of the root vertex o, the following holds.*

(i) $L_{\mathrm{SAW},o}(G)$ is regular if and only if all ends of G have size 1.

(ii) $L_{\mathrm{SAW},o}(G)$ is context-free if and only if all ends of G have size at most 2.

Relating walks in labelled graphs, in particular Cayley graphs, with formal language theory has an important history. Let $\mathcal{P}(o,o)$ be the set of all walks in X starting and ending at o, possibly with several self-intersections. For a Cayley graph of a group Γ, the language $L(\mathcal{P}(o,o))$ is called the *word problem* of Γ.

Anisimov showed in [2] that the word problem is regular if and only if the group is finite. In a ground-breaking work, Muller and Schupp [46] showed that the word problem is context-free if and only if the group is virtually free, that is the group contains a free subgroup of finite index. In particular, regularity, respectively context-freeness of the word problem are group invariants which do not depend on the specific generating set.

Sadly this is not the case for the language of self-avoiding walks, as different generating sets can produce different end-sizes in the respective Cayley graphs. This motivated us to look for a more general classes of formal languages.

Multiple context-free languages (MCFLs) were introduced by Seki et al. [52] as a generalisation of context free languages capable of modelling cross-serial dependencies occurring in some natural languages such as Swiss German. A concise definition of MCFLs will be given in Section 2.2; for now we only mention that they share many useful traits with context-free languages, including polynomial time parsability, semi-linearity and closure properties. MCFLs can be further classified depending on the largest dimension m of tuples involved to obtain m-MCFLs, which form an infinite strictly increasing hierarchy.

While MCFLs are not very well known and may seem artificial at first, they appear in some natural problems. The word problem on $\mathbb{Z}^d$ is not context free, but it was shown to be multiple context-free in seminal work by Salvati [51] for $d = 2$, and this result has since been extended by Ho [33] to all positive integers d.

For our characterisation of graphs having a multiple context-free language of self-avoiding walks we need an additional result about MCFLs. This result is introduced in Chapter 5, where we study languages defined by comparing lengths of runs of consecutive identical letters. In particular, we consider languages of the form

$$L_k = \{a_1^{n_1} a_2^{n_2} \cdots a_k^{n_k} \mid n_1 \geq n_2 \geq \cdots \geq n_k \geq 0\}$$

and generalisations thereof. It is easy to see that L_1 and L_2 are context-free, and it is a standard exercise to show that L_3 is not context-free by using the pumping lemma for CFLs. The main result of Chapter 5 generalises these observations. In particular we show that L_k is $\lceil k/2 \rceil$-multiple context-free but not $(\lceil k/2 \rceil - 1)$-multiple context-free. The result has appeared in [41] and is joint work with Lehner.

In Chapter 6 we generalise the results of Chapter 4 to graphs with thin ends of size larger than two. In particular we prove the following main result, which is contained in the joint work [40] with Lehner.

Theorem 1.0.2. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph and let $o \in V(G)$. Then $L_{\mathrm{SAW},o}(G)$ is an MCFL if and only if all ends of G are thin.*

Applied to Cayley graphs of groups, this theorem states that the language of self-avoiding walks on a Cayley graph of a group is multiple context-free if and only if the group is virtually free. In particular, the property of having a multiple context-free language of self-avoiding walks is a group invariant.

During the proof we also obtained the following result, which extends a result of Alm and Janson [1] for one-dimensional lattices.

Theorem 1.0.3. *Let G be a locally finite, connected, quasi-transitive graph having only thin ends and let $o \in V(G)$. Then the SAW-generating function $F_{\mathrm{SAW},o}(z)$ is algebraic over $\mathbb{Q}$. In particular the connective constant $\mu(G)$ is an algebraic number.*

As this thesis is comprised of four research articles, we decided to provide each chapter with its own introduction. Thus after getting familiar with the definitions and notation from Chapter 2, each chapter can be read separately.

2 Basic background and notation

Throughout this thesis, we denote by $\mathbb{N}$ the set of natural numbers starting at 1, by $\mathbb{N}_0$ the set $\mathbb{N} \cup \{0\}$, and by $[n]$ the set $\{1, \ldots, n\}$.

2.1 Graph theory

A *graph* G consists of a set $V(G)$ of vertices and a set $E(G)$ of edges. Every edge $e \in E(G)$ starts at its *initial vertex* $e^- \in V(G)$ and ends at its *terminal vertex* $e^+ \in V(G)$. The notation $v \in e$ indicates that v is one of the endpoints of e. Two different vertices of G are *adjacent*, if they are the endpoints of some edge of G. We do not allow loops, so the two endpoints e^- and e^+ of every edge $e \in E(G)$ are different. Furthermore all graphs considered are *undirected*, so all edges appear in pairs $e, \bar{e}$ having the same endpoints but different directions. In other words, for $e \in E(G)$ the edge $\bar{e} \in E(G)$ satisfies $\bar{e}^- = e^+$, $\bar{e}^+ = e^-$ and $\bar{\bar{e}} = e$. In many cases it is useful to consider pairs $(e, \bar{e})$ of edges as single undirected edges. In particular, to keep figures as simple as possible, we draw only undirected edges. A graph is called *simple*, if it contains no *multiple edges*, or in other words, if every edge e is uniquely defined by the pair (e^-, e^+) of its initial and terminal vertex. We sometimes abuse notation and write $e = e^-e^+$; if G is not simple we still use this notation, but will include further information needed to identify e among the edges with the same initial and terminal vertices if necessary. The *degree* $\deg(v)$ of a vertex v is the number of outgoing edges of v. The graph G is called *locally finite*, if all vertices have finite degree.

A *walk* in a graph is an alternating sequence $p = (v_0, e_1, v_1, \ldots, e_n, v_n)$ of vertices $v_i \in V(G)$ and edges $e_i \in E$ such that $e_i^- = v_{i-1}$ and $e_i^+ = v_i$ for every $i \in [n]$. Its *length* is the number n of edges and its *initial* and *terminal vertices* are $p^- = v_0$ and $p^+ = v_n$, respectively. This comprises the *trivial walk* (v) of length 0, starting and ending at a vertex v and also the *empty walk* $\emptyset$ consisting of no vertices and no edges. A walk p is called *self-avoiding* or a *SAW*, if the vertices in p are pairwise different. The *distance* $d_G(u, v)$ of two vertices u and v of G is the length of the shortest walk in G connecting u and v. The *diameter* of G is the maximal distance of two vertices in G.

For two vertices u and v of p we write upv for the maximal sub-walk of p starting at u and ending at v. If $u = v_0$ or $v = v_n$ we omit the corresponding vertex and denote the sub-walk by pv or up, respectively. We extend this notation even further and denote for walks $p_1, \ldots, p_n$ and vertices $v_0, \ldots, v_n$ in the respective walks the concatenation $(v_0p_1v_1)(v_1p_2v_2)\ldots(v_{n-1}p_nv_n)$ of the sub-walks $v_{i-1}p_iv_i$ by $v_0p_1v_1p_2\ldots p_nv_n$. If the terminal vertex v of p_1 coincides with the initial vertex of p_2, we write p_1p_2 instead of p_1vp_2, and similarly for concatenations of multiple walks. If e is an edge connecting the terminal vertex v_1 of p_1 to the initial vertex v_2 of p_2, then we write p_1ep_2 instead of

$p_1v_1(v_1, e, v_2)v_2p_2$, and similarly for concatenations with more parts. A walk is *closed* if its initial and terminal vertex coincide.

A graph G is *connected* if any two vertices u, v are the endpoints of some walk p in G. *Components* of G are maximal connected subgraphs. A *cycle* is a finite connected graph where each vertex has degree 2.

A *tree* is a connected and cycle-free graph. A *rooted tree* is a tree where one vertex has been designated the *root*. For vertices u and v of a rooted tree we say that u is an *ancestor* of v and v is a *descendant* of u if any walk from r to v contains u. The unique ancestor of v that is also a neighbour of v is called its *parent* and denoted by $v^\uparrow$; descendants of v in the neighbourhood of v are called its *children*. The *forefather* of a set $A \subseteq V(T)$ is the unique common ancestor of all vertices in A such that none of its children is an ancestor of all vertices in A. The *cone* at a vertex v in a rooted tree, denoted by K_v, is the subtree induced by v and its descendants. An *ordered tree* is a rooted tree with an ordering specified for the children of each vertex; in this case we denote the i-th child of a vertex v with respect to this order by $v_i^\downarrow$.

A tree consisting only of vertices of degree at most 2 is a *path*. We point out that unlike walks, paths are graphs and have no direction; a finite path can be seen as the support of a self-avoiding walk. Given two disjoint subsets A and B of vertices of a graph G, an *A–B-path* on G is a subgraph of G which is a finite path intersecting A and B only in its two endpoints. A *ray* is a one-way infinite path and a *double ray* is a two-way infinite path.

For any set $K \subseteq V(G)$ we denote by $G - K$ the subgraph obtained from G by removing K and all edges incident to K. If removing K disconnects G, then K is called a *separating set*. In this case, if $K = \{v\}$ then v is a *cut-vertex*. The vertex sets of the maximal connected subgraphs of $G - K$ are called *components* of $G - K$. Furthermore, we denote by $G[K]$ the subgraph of G *induced* by K, that is the graph $G - (V(G) \setminus K)$.

The space of ends of a connected graph was introduced by Halin in [24], and – without graph terminology – earlier by Freudenthal (see [17] and [18]).

Two rays in a graph G are said to be *equivalent*, if for every finite set $K \subseteq V(G)$ they end up in the same component of $G - K$, that is, all but finitely many of their vertices are contained in that component. An *end* of G is an equivalence class of rays with respect to this equivalence relation. If ω is an end and $K \subseteq V(G)$ is finite, then we write $C(K, \omega) = C_G(K, \omega)$ for the unique component of $G - K$ in which all representing rays of ω end up, and say that ω belongs to this component. Two ends ω_1 and ω_2 of a graph G are *separated* by K if $C(K, \omega_1) \neq C(K, \omega_2)$. Halin [25] showed that an end containing arbitrarily many disjoint rays must contain an infinite family of disjoint rays, hence the maximum number of disjoint rays contained in an end ω is well defined in $\mathbb{N} \cup \{\infty\}$. This number is called the *size* of the end ω. An end of finite size is called *thin*, an end of infinite size is called *thick*.

An *automorphism* γ of a graph G is a permutation of $V(G) \cup E(G)$, mapping vertices onto vertices and edges onto edges while preserving the incidence relations, that is, $(\gamma(e))^+ = \gamma(e^+)$ and $(\gamma(e))^- = \gamma(e^-)$ holds for every $e \in E(G)$. When working with graph automorphisms, we often omit parenthesis and write γu instead of $\gamma(u)$ for the image of a vertex u under the automorphism γ. The set of all automorphisms of G forms a group, called the *automorphism group* of G and denoted by $\mathrm{AUT}(G)$. For a subgroup $\Gamma \subseteq \mathrm{AUT}(G)$ we define an equivalence relation on $V(G)$ by calling u and v equivalent if and only if there is some $\gamma \in \Gamma$ such that $\gamma u = v$. The equivalence classes with respect to this relation are called *orbits* and denoted by Γv. We say that Γ acts *transitively*, if there is exactly one orbit, and that it acts *quasi-transitively*, if there are only finitely many orbits. In this case the graph G is also called *(quasi-)transitive*. When applying graph automorphisms to a sets of vertices or edges of G or a subgraph of G, they are applied element-wise and the result is again a set of vertices or edges or a subgraph of G, respectively. The stabiliser of a vertex v under the action of Γ is $\Gamma_v := \{\gamma \in \Gamma \mid \gamma v = v\}$; for a set S of vertices, the set-wise stabiliser of S is denoted $\Gamma_S := \{\gamma \in \Gamma \mid \gamma s \in S$ for every $s \in S\}$. Finally, a subgraph H of G is called γ*-invariant*, if γ maps H onto itself.

It is well known, that any infinite, locally finite, connected graph which is quasi-transitive has either one, two, or infinitely many ends, see [18]. If it has one end, this end is thick. If it has two ends, both are thin and must have the same size. Finally, if it has infinitely many ends, then some of its ends must be thin. These and many more results were given by Halin in [26].

The action of the automorphism group $\mathrm{AUT}(G)$ of a locally finite, connected graph extends in the obvious way to its ends. An automorphism $\gamma \in \mathrm{AUT}(G)$ is called

- *elliptic*, if its fixes a finite subset of $V(G)$,
- *parabolic*, if it is not elliptic and fixes a unique end of G, and
- *hyperbolic*, if it is not elliptic and fixes each of a unique pair of ends of G.

Halin [26] showed that any graph automorphism is either elliptic, parabolic, or hyperbolic, and additionally, that these different types of automorphisms have the following properties. Firstly, γ is elliptic if and only if for some (every) vertex v of G the sequence $v, \gamma v, \gamma^2 v, \ldots$ contains only finitely many different vertices and is periodic. Secondly, if γ is hyperbolic then the two ends fixed by γ have the same finite size k and G contains k disjoint double rays invariant under γ. Finally, if γ is parabolic then the unique end fixed by γ is thick and G contains infinitely many double rays invariant under γ.

A graph G is called *accessible* if there is a natural number k such that any two ends can be separated by a set of vertices of size at most k. Originally the notion of accessibility comes from group theory. Stalling's theorem about ends of groups states that some (every) Cayley graph of a finitely generated group Γ has more than one end if and only if Γ admits a nontrivial decomposition as an amalgamated free product or an HNN-extension over a finite subgroup. Γ is called *accessible* if the process of iterated nontrivial splitting of Γ always terminates in a finite number of steps. Thomassen and Woess [56] showed that accessibility of a group is equivalent to accessibility of some (and thus all) of its Cayley graphs.

A *quasi-isometry* between two metric spaces (X, d_X) and (Y, d_Y) is a mapping $\varphi : X \to Y$ such that there are constants $A > 0$ and $B, B' \geq 0$ such that for all $x_1, x_2 \in X$ and $y \in Y$,

$$A^{-1} d_X(x_1, x_2) - B \leq d_Y(\varphi x_1, \varphi x_2) \leq A d_X(x_1, x_2) + B \quad \text{and} \quad d_Y(y, \varphi X) \leq B'.$$

Two connected graphs G and H are called *quasi-isometric*, if the corresponding metric spaces (G, d_G) and (H, d_H) are quasi-isometric, where d_G and d_H denote the standard graph distance in G and H, respectively. Every quasi-isometry φ has a quasi-inverse $\psi : Y \to X$, which is a quasi-isometry such that $\psi\varphi$ and $\varphi\psi$ are at bounded distance from the respective identity mappings. In other words, there is some constant $M > 0$ such that $d_X(x, \psi\varphi(x)) \leq M$ and $d_Y(y, \varphi\psi(y)) \leq M$ for all $x \in X$, $y \in Y$. In particular being quasi-isometric is an equivalence relation. It is well known that any quasi-isometry between graphs G and H can be extended to the ends of G and that this extension maps thick ends to thick ends and thin ends to thin ends, see for example Lemma 21.4 in [61].

Remark 2.1.1. Every connected quasi-transitive graph is quasi-isometric to some transitive graph. To see this, pick a finite set of representatives $K \subseteq V(G)$ of the set of vertex orbits $\{\Gamma v \mid v \in V(G)\}$ such that the induced subgraph $G[K]$ is connected. Let D be the diameter of the graph $G[K]$. Given $u, v \in V(G)$, there is some $\gamma \in \Gamma$ such that we have $d_G(v, \gamma u) \leq D$. Now consider the new graph G^{2D+1} with the same vertex set $V(G)$, where two vertices u, v are connected by a non-oriented edge whenever $1 \leq d_G(u, v) \leq 2D + 1$. In G^{2D+1}, each orbit Γv induces a connected, locally finite subgraph on which Γ acts transitively. It is easy to check that this subgraph is quasi-isometric to G.

An *edge-labelled* graph is a graph G together with a *label function* ℓ assigning to every edge $e \in E(G)$ an element of some finite set $\mathbf{\Sigma}$, called *label alphabet*. The labelling is called *deterministic*, if any two edges e and f starting at the same vertex $e^- = f^-$ have different labels. For quasi-transitive graphs, we would like the edge-labelling to be compatible with the action of a quasi-transitive subgroup of $\mathrm{AUT}(G)$. To this end, we denote by $\mathrm{AUT}(G, \ell)$ the group of *label-preserving* graph automorphisms; when speaking of a quasi-transitive edge-labelled graph it will be implicitly assumed that $\mathrm{AUT}(G, \ell)$ acts quasi-transitively. Note that in the case of a deterministic labelling ℓ, the group $\mathrm{AUT}(G, \ell)$ acts freely on G, that is, the identity in $\mathrm{AUT}(G, \ell)$ is the only element fixing a vertex of G.

One well-known class of simple, connected, locally finite, transitive graphs that come with a natural deterministic edge-labelling are Cayley graphs of finitely generated groups. Starting with a symmetric generating set S of a group Γ, the *Cayley graph* $G = \mathrm{Cay}(\Gamma, S)$ has vertex set $V(G) = \Gamma$. We choose $\mathbf{\Sigma} = S$, and for each $\gamma \in \Gamma$ and $s \in S$ there is a directed edge from γ to γs with label s. The left regular action of Γ on itself extends to an action on G by label preserving automorphisms; in fact, it is not hard to see that $\Gamma = \mathrm{AUT}(G, \ell)$.

The following result originates from the theory of Cayley graphs of finitely generated groups and also holds for our more general setting.

Lemma 2.1.2. *The group $\Gamma = \mathrm{AUT}(G, \ell)$ of all graph automorphisms of the connected deterministically edge-labelled graph G preserving the edge-labels acts fixed-point freely: if $\gamma \in \Gamma$ and $\gamma v = v$ for some $v \in V(G)$ then $\gamma = 1_\Gamma$, the unit element of Γ.*

In particular, if Γ acts quasi-transitively then it is finitely generated.

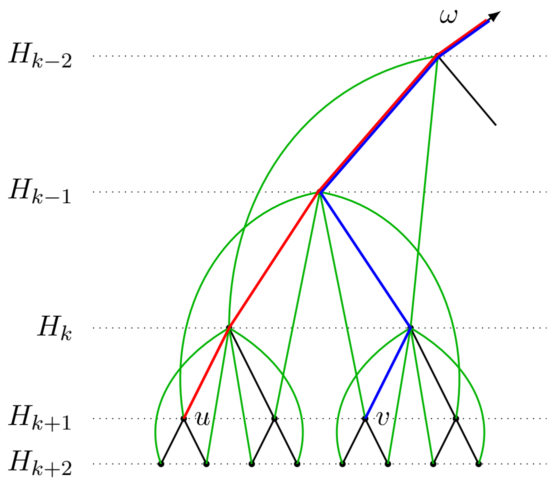


Figure 2.1: The grandparent graph G_{GP}. Edges connecting vertices to their grandparents are green. The automorphism γ maps u onto v and thus the red ray onto the blue ray.

Proof. Suppose $\gamma v = v$. Since the labelling is deterministic, $\gamma u = u$ for all neighbours u of v. By connectedness of G, we must have $\gamma = 1_G$. If Γ acts quasi-transitively, then the construction in Remark 2.1.1 yields a connected, locally finite subgraph on which Γ acts transitively and fixed-point freely. Therefore that subgraph is a Cayley graph of Γ, and Γ is finitely generated, see for instance the note [50] by Sabidussi. □

Remark 2.1.3. For any two Cayley graphs of the same finitely generated group Γ with respect to two different finite, symmetric sets of generators, the identity mapping is a quasi-isometry with $B = B' = 0$ in (4.1), that is, the mapping is *bi-Lipschitz*.

Using Lemma 2.1.2, it is not hard to show that there are transitive graphs G not admitting any deterministic labelling ℓ such that $\mathrm{AUT}(G, \ell)$ acts quasi-transitively.

Example 2.1.4. Consider the grandparent graph G_{GP}, defined as follows. Fix some end ω of the infinite 3-regular tree T_3 and let the graph "hang down" from the end ω. Then the graph can be seen as a union of horizontal layers H_k, $k \in \mathbb{Z}$. Every vertex $v \in H_k$ is adjacent to one vertex in H_{k-1}, called parent of v and two vertices in H_{k+1}, called children of v. We add additional undirected edges connecting every child of v to its grandparent, that is the parent of v. Doing this, we end up with the grandparent graph G_{GP} shown in Figure 2.1.

Clearly the group $\mathrm{AUT}(G_{\mathrm{GP}})$ acts transitively on G_{GP}. Soardi and Woess showed in [54] that $\mathrm{AUT}(G_{\mathrm{GP}})$ is the subgroup of $\mathrm{AUT}(T_3)$ fixing the end ω, that is, it maps any ray in ω onto some other ray in ω. Let $\Gamma \le \mathrm{AUT}(G_{\mathrm{GP}})$ be a subgroup of graph automorphisms acting quasi-transitively on G_{GP}. Every layer H_k contains infinitely many vertices, so there must be some $\gamma \in \Gamma$ mapping a vertex u in H_k onto some different vertex $v \neq u$ in H_k. But γ is also a graph automorphism on the subgraph T_3

and fixes the end ω of T_3, so it maps the ray in ω starting at u onto the ray in ω starting at v and thus fixes common ancestors of u and v. In particular, Γ does not act fix-point freely, so by Lemma 2.1.2 the graph G_{GP} does not admit any deterministic labelling ℓ such that $\mathrm{AUT}(G, \ell)$ acts quasi-transitively on G.

Any set of walks on an edge-labelled graph G defines a language in the following way. Extend the label function ℓ to walks $p = (v_0, e_1, \ldots, e_n, v_n)$ by setting

$$\ell(p) = \ell(e_1)\ell(e_2)\ldots\ell(e_n) \in \mathbf{\Sigma}^*.$$

Then for any set of walks $\mathcal{P}$ on G, the associated language is

$$L(\mathcal{P}) := \{\ell(p) \mid p \in \mathcal{P}\}.$$

For a given vertex o of G we denote by $\mathcal{P}_{\mathrm{SAW},o}$ the set of self-avoiding walks of length at least 1 on G starting at o and by $L_{\mathrm{SAW},o}(G) := L(\mathcal{P}_{\mathrm{SAW},o})$ the associated *language of self-avoiding walks*. Note that if the edge-labelling is deterministic, then ℓ is a bijection between $\mathcal{P}_{\mathrm{SAW},o}$ and $L_{\mathrm{SAW},o}(G)$.

2.2 Formal languages

For a finite set of letters $\mathbf{\Sigma}$, called an *alphabet*, we denote by

$$\mathbf{\Sigma}^* := \{w = a_1a_2\ldots a_n \mid n \geq 0, a_i \in \mathbf{\Sigma}\}$$

the set of all words over $\mathbf{\Sigma}$. The number n of letters in a word w is called the *length* of w and denoted by $|w|$; for the unique word of length 0 we write ϵ. A *language* over $\mathbf{\Sigma}$ is a subset of $\mathbf{\Sigma}^*$.

A *context-free grammar* is a quadruple $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{N}$ is a finite set of *non-terminals* with $\mathbf{N} \cap \mathbf{\Sigma} = \emptyset$, $S \in \mathbf{N}$ is the *start symbol* and $\mathbf{P} \subseteq \mathbf{N} \times (\mathbf{N} \cup \mathbf{\Sigma})^*$ is a finite set of *production rules*. We write $A \vdash \alpha$ for $(A, \alpha) \in \mathbf{P}$. If $\alpha \in \mathbf{\Sigma}^*$, we call $A \vdash \alpha$ a *terminal rule*.

A production rule $A \vdash \alpha$ allows us to replace the non-terminal A by the string α. More precisely, for strings $\beta, \gamma \in (\mathbf{N} \cup \mathbf{\Sigma})^*$, we say γ is obtained from β in a *single step of leftmost derivation*, and write $\beta \Rightarrow \gamma$, if there is a decomposition of the form $\beta = \beta_1 A \beta_2$ and $\gamma = \beta_1 \alpha \beta_2$ for some $\beta_1 \in \mathbf{\Sigma}^*$, $\beta_2 \in (\mathbf{N} \cup \mathbf{\Sigma})^*$ such that $A \vdash \alpha \in \mathbf{P}$. Thus, γ is a result of using the rule $A \vdash \alpha$ to replace the leftmost non-terminal in β. A *leftmost derivation* of β from α is a sequence

$$(\alpha = \alpha_0, \alpha_1, \ldots, \alpha_k = \beta)$$

such that $\alpha_{i-1} \Rightarrow \alpha_i$ for every i. We say that β is *derived* from α and write $\alpha \overset{*}{\Rightarrow} \beta$. Each non-terminal $A \in \mathbf{N}$ generates a language $L_A := \{\alpha \mid A \overset{*}{\Rightarrow} \alpha\}$ and the *language generated by* the grammar $\mathbf{G}$ is $L(\mathbf{G}) := L_S$. A *context-free language* is a language generated by a context-free grammar or equivalently, accepted by a pushdown automaton, see [32].

The grammar and its generated language are called *unambiguous* if for every $\alpha \in L(\mathbf{G})$ there is a unique leftmost derivation generating α.

A grammar and its generated language are called *linear,* if each production is of the form

$$A \vdash xBy \quad \text{or} \quad A \vdash x, \quad \text{where} \quad A, B \in \mathbf{N},\ x, y \in \mathbf{\Sigma}^*.$$

If in that situation one always has $y = \epsilon$, then the grammar and the language are called *right-linear* or *regular.* In this case, the language is accepted by a (deterministic) finite state automaton, see [32].

Recall that an ordered tree is a rooted tree with an ordering specified for the children of each vertex. For a given context-free grammar $\mathbf{G}$, a *derivation tree* is an ordered tree D together with a labelling $\lambda : V(D) \to \mathbf{N} \cup \mathbf{\Sigma}^*$ such that

– internal vertices have labels in $\mathbf{N}$,

– leaves have labels in $\mathbf{\Sigma}^*$ and

– whenever $v_1, \ldots, v_k$ are the ordered children of u in D,

$$\lambda(u) \vdash \lambda(v_1) \ldots \lambda(v_k) \in \mathbf{P}.$$

Any ordered tree induces a total order $u_1, \ldots, u_k$ on its leaves and we call D a derivation tree of $w \in \mathbf{\Sigma}^*$ if $w = \lambda(u_1) \ldots \lambda(u_k)$. It is a standard result in formal language theory that there is a bijection between leftmost derivations of $w \in \mathbf{\Sigma}^*$ from $A \in \mathbf{N}$ and derivation trees of w whose roots are labelled A.

Typical tools to show that a language is *not* regular, respectively not context-free, are the well known Pumping Lemmas.

Lemma 2.2.1 (Pumping Lemma for regular languages)**.** *Let L be a regular language over an alphabet $\mathbf{\Sigma}$. Then there is a pumping length $l_p > 0$ such that every $w \in L$ with $|w| \geq l_p$ can be written as $w = xy\tilde{x}$ for some $x, \tilde{x}, y \in \mathbf{\Sigma}^*$, such that $|y\tilde{x}| \leq l_p$, $|y| \geq 1$ and $xy^n\tilde{x} \in L$ for all $n \geq 0$.*

Lemma 2.2.2 (Pumping Lemma for context-free languages)**.** *Let L be a context-free language over an alphabet $\mathbf{\Sigma}$. Then there is a pumping length $l_p > 0$ such that every $w \in L$ with $|w| \geq l_p$ can be written as $w = xyz\tilde{y}\tilde{x}$ for some $x, \tilde{x}, y, \tilde{y}, z \in \mathbf{\Sigma}^*$, such that $|yz\tilde{y}| \leq l_p$, $|y\tilde{y}| \geq 1$ and $xy^nz\tilde{y}^n\tilde{x} \in L$ for all $n \geq 0$.*

The *(commutative) language generating function* of a given language L over the alphabet $\mathbf{\Sigma} = \{a_1, \ldots, a_m\}$ is a formal power series in the commuting variables $a_1, \ldots, a_m$ over $\mathbb{Z}$

$$F_L(a_1, \ldots, a_m) = \sum_{w \in L} c(w),$$

where $c(w) = \prod_{i=1}^m a_i^{l_i}$ for any word w containing exactly l_i copies of the letter a_i.

A famous result of Chomsky and Schützenberger [7] states that the commutative generating function of the language $L(\mathbf{G})$ generated by an unambiguous context-free grammar $\mathbf{G}$ is algebraic over $\mathbb{Q}$, meaning that there is an irreducible polynomial P in $m+1$ variables with coefficients in $\mathbb{Q}$ such that

$$P\left(F_L(a_1, \ldots, a_m), a_1, \ldots, a_m\right) = 0.$$

A proof of this statement can be found in [39].

To motivate the upcoming introduction of multiple context-free grammars as a generalisation of context-free grammars, let us briefly discuss a different notation for context-free grammars $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$. When producing words, one usually starts with the start symbol S and iteratively replaces non-terminals according to the rules given by $\mathbf{P}$. In terms of derivation trees, we build the trees starting from the top (root).

There is also an alternative way to build words using production rules. A production rule $A \vdash x_0 A_1 x_1 \ldots A_n x_n$ tells us that we can obtain an element of the language L_A generated by A by sticking together the strings $x_0, \ldots, x_n \in \mathbf{\Sigma}^*$ with strings $y_i \in L_{A_i}$, $i \in [n]$, according to the rule. This way of thinking is closely related to predicate logic. We might say that a word $w \in \mathbf{\Sigma}^*$ has property $A \in \mathbf{N}$ and write $\vdash_{\mathbf{G}} A(w)$ if $w \in L_A$. Then the rule $A \vdash x_0 A_1 x_1 \ldots A_n x_n$ is equivalent to the statement

$$\text{“If } \vdash_{\mathbf{G}} A_1(z_1), \ldots, \vdash_{\mathbf{G}} A_n(z_n), \text{ then also } \vdash_{\mathbf{G}} A(x_0 z_1 x_1 \ldots z_n x_n)\text{”.}$$

In this statement, the z_i play the role of variables and it is natural to write the production as

$$A(x_0 z_1 x_1 \ldots z_n x_n) \leftarrow A_1(z_1), \ldots, A_n(z_n).$$

With this in mind, the natural way to generate words is by starting with terminal rules and constructing the derivation from bottom to top, starting at its leaves.

Keeping this in mind, we introduce multiple context-free languages. As for context-free languages, the first two ingredients are an alphabet $\mathbf{\Sigma}$ and a set of non-terminals $\mathbf{N}$. In the intuition above we treated every non-terminal as a property applying to all strings it generates. In a similar way, a non-terminal of a multiple context-free language should be viewed as a property applying to tuples of strings. To realise this, every non-terminal is assigned an integer $r \geq 1$ counting the size of the tuples, called *rank*. In other words, $\mathbf{N}$ is a finite disjoint union $\mathbf{N} = \bigcup_{r \in \mathbb{N}} \mathbf{N}^{(r)}$ of finite sets $\mathbf{N}^{(r)}$, whose elements are called *non-terminals of rank r*. *Production rules ρ* of a multiple context-free grammar with non-terminals $\mathbf{N}$ and alphabet $\mathbf{\Sigma}$ are expressions of the form

$$A(\alpha_1, \ldots, \alpha_r) \leftarrow A_1(z_{1,1}, \ldots, z_{1,r_1}), \ldots, A_n(z_{n,1}, \ldots, z_{n,r_n}),$$

where

- $n \geq 0$,
- $A \in \mathbf{N}^{(r)}$ and $A_i \in \mathbf{N}^{(r_i)}$ for all $i \in [n]$,
- $z_{i,j}$ are variables,
- $\alpha_1, \ldots, \alpha_r$ are strings over $\mathbf{\Sigma} \cup \{z_{i,j} \mid i \in [n], j \in [r_i]\}$, such that each $z_{i,j}$ occurs at most once in $\alpha_1 \ldots \alpha_r$.

Production rules with $n = 0$ are called *terminating rules.* For convenience we sometimes use the shortened notation

$$A(\alpha_1, \dots, \alpha_r) \leftarrow (A_i(z_{i,1}, \dots, z_{i,r_i}))_{i \in [n]}$$

for the production rule ρ. For a non-terminal $A \in \mathbf{N}^{(r)}$ and words $w_1, \dots, w_r \in \mathbf{\Sigma}^*$ an expression of the form $A(w_1, \dots, w_r)$ is called a *term.* The *application* of the production rule ρ to a sequence $(A_i(w_{i,1}, \dots, w_{i,r_i}))_{i \in [n]}$ of n terms yields the term $A(w_1, \dots, w_r)$, where w_l is obtained from α_l by replacing every variable $z_{i,j}$ by the word $w_{i,j}$. The non-terminal A is called the *head* of the production and the sequence of non-terminals $A_1, \dots, A_n$ are its *tail.*

A *multiple context-free grammar* is a quadruple $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{N}$ is a finite ranked set of non-terminals, $\mathbf{\Sigma}$ is a finite alphabet, $\mathbf{P}$ is a finite set of production rules over $(\mathbf{N}, \mathbf{\Sigma})$ and $S \in \mathbf{N}^{(1)}$ is the *start symbol.* We call $\mathbf{G}$ *k-multiple context-free* or a *k-MCFG*, if the rank of all non-terminals is at most k.

A term τ is called *derivable* in $\mathbf{G}$, written $\vdash_{\mathbf{G}} \tau$ if there is a sequence $\mathcal{A}$ of derivable terms such that the application of a rule $\rho \in \mathbf{P}$ to $\mathcal{A}$ yields τ. It is implicit in this recursive definition that if $A(w_1, \dots, w_r) \leftarrow$ is a terminal rule, then the term $A(w_1, \dots, w_r)$ is derivable by letting $\mathcal{A}$ be the empty sequence. The language *generated* by $\mathbf{G}$ is the set $L(\mathbf{G}) := \{w \in \mathbf{\Sigma}^* \mid \vdash_{\mathbf{G}} S(w)\}$. We call a language *$k$-multiple context-free* or a k-MCFL if it is generated by a k-MCFG.

The following simple example of a 2-MCFG should be beneficial for a better understanding of the concepts above and a more general version of this grammar will come up again in Chapter 5.

Example 2.2.3. Consider the MCFG $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{N} = \{S, A\}$, $\mathbf{\Sigma} = \{a, b, c\}$ and the set $\mathbf{P}$ consists of the rules $\rho_1, \dots, \rho_5$ given as follows:

$$\begin{aligned}
\rho_1: &\quad S(z_1 z_2) \leftarrow A(z_1, z_2), \\
\rho_2: &\quad A(a z_1 b, z_2 c) \leftarrow A(z_1, z_2), \\
\rho_3: &\quad A(a z_1 b, z_2) \leftarrow A(z_1, z_2), \\
\rho_4: &\quad A(a z_1, z_2) \leftarrow A(z_1, z_2), \\
\rho_5: &\quad A(\epsilon, \epsilon) \leftarrow .
\end{aligned}$$

From the production rules it is immediately clear that the rank of A is 2 as A works with pairs of strings. In particular the rank of the start symbol S is 1 by definition, so that $\mathbf{G}$ is 2-multiple context-free. We use the recursive definition above to find all terms derivable in $\mathbf{G}$.

By the terminal rule ρ_5, the term $A(\epsilon, \epsilon)$ is derivable, we write $\vdash_G A(\epsilon, \epsilon)$. Applying $A(az_1, z_2) \leftarrow A(z_1, z_2)$ to the term $A(\epsilon, \epsilon)$, we replace z_1 and z_2 on the left side of the rule by the empty word ϵ and obtain the term $A(a, \epsilon)$. In a similar way, consecutive application of ρ_4 yields that all terms of the form $A(a^k, \epsilon), k \geq 0$ are derivable. Making use of rule ρ_3, we obtain $\vdash_{\mathbf{G}} A(a^{k+l} b^l, \epsilon)$ for every $k, l \geq 0$. Analogously, the rule ρ_2 provides $\vdash_{\mathbf{G}} A(a^{k+l+m} b^{l+m}, c^m)$ for $k, l, m \geq 0$. In a final step, we use the rule

$S(z_1z_2) \leftarrow A(z_1, z_2)$ containing the start symbol S to create words of the language $L(\mathbf{G})$: this rule ρ_1 is used to concatenate the pairs (w_1, w_2) of strings appearing in derivable terms $A(w_1, w_2)$ and yields $\vdash_{\mathbf{G}} S(w_1w_2)$. As a conclusion, all terms of the form $S(a^{k+l+m}b^{l+m}c^m)$ are derivable.

For the converse direction, note that any derivable term $S(w)$ must arise from an application of ρ_1, so that $w = w_1w_2$ for some derivable term $A(w_1, w_2)$. It is not hard to see that in any such term, $w_1 = a^kb^l$ and $w_2 = c^m$ holds for some $k \geq l \geq m \geq 0$: as the only term arising from a terminal rule, $A(\epsilon, \epsilon)$ satisfies this condition and the rules ρ_2, ρ_3 and ρ_4 preserve it. We conclude that the language generated by $\mathbf{G}$ is

$$L(\mathbf{G}) = \{a^{k+l+m}b^{l+m}c^m \mid k, l, m \geq 0\} = \{a^kb^lc^m \mid k \geq l \geq m \geq 0\}.$$

Remark 2.2.4. As one might already guess from the discussion right before the introduction of MCFGs, the class of context-free languages coincides with the class of 1-multiple context-free languages. A given context-free grammar $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ can be easily translated into a 1-MCFG by replacing every production rule

$$A \vdash w_0A_1w_1A_2 \dots A_nw_n \in \mathbf{P},$$

where $A, A_1, \dots, A_n \in \mathbf{N}$ and $w_1, \dots, w_n \in \mathbf{\Sigma}^*$ with the multiple-context-free production rule

$$A(w_0z_1w_1z_2 \dots z_nw_n) \leftarrow A_1(z_1), \dots, A_n(z_n)$$

over $(\mathbf{N}, \mathbf{\Sigma})$. The resulting 1-MCFG $\mathbf{G}' = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}', S)$ then generates the same language $L(\mathbf{G})$.

Remark 2.2.5. Sometimes it will be convenient to work with a slightly different definition of multiple context-free grammars allowing non-terminals A to have rank $r = 0$. For such a non-terminal A of rank 0, the only valid term is $A(\emptyset)$, where $\emptyset$ denotes the empty tuple. We point out that $\emptyset$ is different from the 1-tuple (ϵ) containing the empty string. Note that the generative ability of k-multiple context-free languages does not change and that all properties discussed here remain valid under this variation.

In the Chomsky-hierarchy of formal languages, multiple context-free languages lie strictly between context-free languages and the bigger class of context-sensitive languages. MCFLs share some important properties with context-free languages. They are closed under under homomorphisms, inverse homomorphisms, union, intersection with regular languages and Kleene closure. Furthermore they are parsable in polynomial time and semilinear.

Derivation trees for multiple context-free languages were first defined by Seki et al. [52]; we use a slight variation of their definition. Let $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ be an MCFG. A *derivation tree* of a term τ with respect to the grammar $\mathbf{G}$ is an ordered tree D whose vertices are labelled with elements of $\mathbf{P}$ satisfying the following conditions:

- The root of D has $n \geq 0$ children and is labelled with a rule $\rho \in \mathbf{P}$.
- For $i \in [n]$ the subtree D_i rooted at the i-th child of the root of D is a derivation tree of a term τ_i.

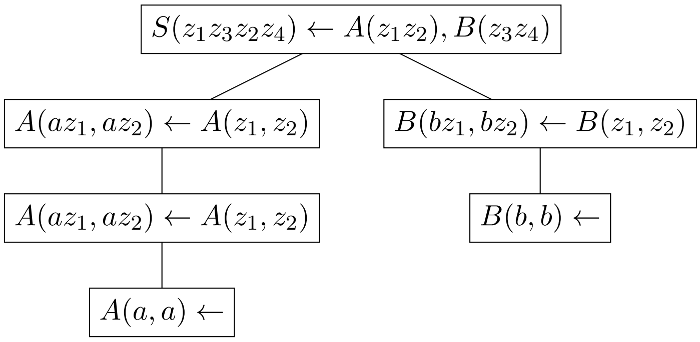


Figure 2.2: The unique derivation tree D of the word $w(D) = a^3b^2a^3b^2$ with respect to the grammar $\mathbf{G}$.

– The rule ρ applied to the sequence $(\tau_i)_{i\in[n]}$ yields τ.

It is not hard to see that $\vdash A(w_1, \ldots, w_r)$ if and only if there is a derivation tree D of $A(w_1, \ldots, w_r)$. However, in general such a derivation tree need not be unique.

An MCFG $\mathbf{G}$ is called *unambiguous*, if for every term $S(w)$ there is at most one derivation tree of $S(w)$ with respect to $\mathbf{G}$. An MCFL is called *unambiguous* if it is generated by some unambiguous MCFG. We denote by $w(D)$ the tuple of strings $w_1, \ldots, w_r$ generated by D and by $\ell(D)$ the label of the root of D.

In some sense derivations trees of MCFGs are more natural than derivation trees of CFGs. The tree corresponding to the derivation process of a term τ in an MCFG consists of a single vertex labelled ρ for every rule ρ occurring in the process.

Example 2.2.6. Consider the MCFG $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{N} = \{S, A, B\}$, $\mathbf{\Sigma} = \{a, b\}$ and $\mathbf{P}$ consists of the rules

$$\begin{aligned} S(z_1z_3z_2z_4) &\leftarrow A(z_1z_2), B(z_3z_4), \\ A(az_1, az_2) &\leftarrow A(z_1, z_2), \\ B(bz_1, bz_2) &\leftarrow B(z_1, z_2), \\ A(a, a) &\leftarrow, \\ B(b, b) &\leftarrow . \end{aligned}$$

It is a simple exercise to show that $L(\mathbf{G}) = \{a^m b^n a^m b^n \mid m, n \in \mathbb{N}\}$. The unique derivation tree D of the word $w(D) = a^3b^2a^3b^2$ can be found in Figure 2.2.

Remark 2.2.7. Let D be a derivation tree and let v be a vertex of D. Then by definition replacing the subtree D' of D rooted at v by a derivation tree D'' satisfying $\ell(D'') = \ell(D')$ yields a derivation tree.

The pumping lemma for k-MCFLs, similarly to the well known pumping lemma for CFLs, provides a convenient way to show that certain languages are not k-multiple context free.

Lemma 2.2.8 ([52, Lemma 3.2])**.** *For every infinite k-MCFL L there is some $w \in L$, which can be written in the form $w = x_1 y_1 x_2 y_2 \dots x_{2k} y_{2k} x_{2k+1}$ for some $x_i, y_i, \in \mathbf{\Sigma}^*$ such that*

- $y_1 y_2 \dots y_{2k} \neq \epsilon$ *and*
- $x_1 y_1^n x_2 y_2^n \dots x_{2k} y_{2k}^n x_{2k+1} \in L$ *for every* $n \in \mathbb{N}_0$.

Note that this lemma is weaker than the pumping lemma for CFLs: it only provides the existence of "pumpable" strings whereas the pumping lemma for CFLs states that all words exceeding a certain length are pumpable.

3 A general bridge theorem for self-avoiding walks

In [22] Grimmett and Li proved a locality theorem for connective constants, namely that the connective constants of two graphs are close in value whenever the graphs agree on a large ball around the origin and some further conditions are satisfied. To obtain this result they generalised the concept of bridges to the class of quasi-transitive graphs by introducing the notion of graph height functions, which are basically maps assigning an integer $h(v)$ to every vertex v. Graph height functions have to be adapted to the graph structure according to Definition 3.1.1. A good survey on graph height functions and in particular some existence results for Cayley graphs can be found in [20]. Similarly to the connective constant, the *bridge constant* $\beta(G, h)$ of G with respect to h is the base of the exponential growth of the number of bridges $b_{n,o}$ of length n starting at o,

$$\beta(G, h) = \lim_{n\to\infty} b_{n,o}^{1/n}.$$

The proof of its existence uses the fact that the concatenation of two bridges is again a bridge. While it is rather simple for transitive graphs, existence of the bridge constant of quasi-transitive graphs and its independence of the choice of o are shown in [22]. Furthermore, [22] also contains the first bridge-theorem for quasi-transitive graphs, stating that the bridge constant with respect to a *unimodular* graph height function is equal to its connective constant.

The goal of this chapter is to provide a version of the bridge theorem applicable to an even wider class of graphs. More precisely, we want to get rid of unimodularity in the conditions of the bridge theorem. Our main result Theorem 3.2.2 states that for any graph height function h on a graph G, the connective constant $\mu(G)$ is equal to the maximum of the bridge constant $\beta(G, h)$ with respect to h and the bridge constant $\beta(G, -h)$ with respect to the "reflected" graph height function $-h$.

As a consequence of this theorem, all results discussed by Grimmett and Li in Section 5 of [22] concerning locality of connective constants also hold in the case where the graph height functions are not unimodular. In particular they obtained conditions, under which the connective constants of sequences of graphs possessing unimodular graph height functions converges to the connective constant of a limit graph of this sequence. The proofs in the non-unimodular case work exactly the same way after replacing corresponding results by the results obtained in this chapter, so they will not be discussed here.

In Section 3.3, we provide a concrete example. The grandparent graph was given by Trofimov in [57] as an example of a connected, locally finite, transitive graph with a non-unimodular automorphism group, so it admits only non-unimodular graph height functions. We calculate the bridge constants with respect to the generic graph height function and use our version of the bridge theorem to obtain the connective constant of the grandparent graph. This connective constant can also be obtained by using a different method described in Chapter 4.

3.1 Graph height functions and bridges

All graphs discussed in this chapter are assumed to be simple, even if not mentioned explicitly. Thus it makes sense to represent any walk p by its sequence of vertices $(v_0, v_1, \dots, v_n)$ and omit its edges. As we are interested in the growth rate of the number of self-avoiding walks of length n for n going to infinity, we only consider infinite graphs.

The following definition of graph height functions on graphs is taken from [22].

Definition 3.1.1. Let G be a locally finite, connected graph. A graph height function on G is a pair (h, Γ), where

- $h : G \to \mathbb{Z}$,
- $\Gamma \leq \mathrm{AUT}(G)$ is a subgroup of graph automorphisms acting quasi-transitively on G and h is Γ-*difference-invariant* in the sense that

$$h(\gamma v) - h(\gamma u) = h(v) - h(u) \quad \text{for all } \gamma \in \Gamma,\ u, v \in V(G),$$

- for every $v \in V(G)$, there exist $u, w \in V(G)$ adjacent to v such that

$$h(u) < h(v) < h(w).$$

A graph height function (h, Γ) is called *unimodular* if the action of Γ on G is unimodular, that is if $|\Gamma_u v| = |\Gamma_v u|$ for all $u, v \in V(G)$ with $v \in \Gamma u$. Note that by definition any graph possessing a graph height function must be infinite and quasi-transitive.

Denote by $d = d(h, \Gamma)$ the smallest integer satisfying $h(u) - h(v) \leq d$ for all edges uv of G. This integer $d(h, \Gamma)$ always exists: the height h is Γ-difference-invariant and Γ acts with finitely many orbits on pairs (u, v) of adjacent vertices as G is locally finite. When talking about a graph height function (h, Γ) we often simply write h and omit Γ.

Remark 3.1.2. There are infinite, locally finite, connected, quasi-transitive graphs not supporting any graph height functions. This is still true when considering the set of all Cayley graphs of finitely generated groups: it was shown in [21], that neither the Cayley graph of the Grigorchuk group nor the Cayley graph of the Higman group admits a graph height function.

From now on let G be an infinite, locally finite, connected, quasi-transitive graph and let (h, Γ) be a graph height function on G. A self-avoiding walk $(v_0, v_1, \dots, v_n)$ is called

- a *bridge*, if $h(v_0) < h(v_i) \leq h(v_n)$ holds for every $i \in [n]$,

– a *reversed bridge*, if $h(v_0) > h(v_i) \geq h(v_n)$ holds for every $i \in [n]$,

– a *half-space-walk* (HSW), if $h(v_0) < h(v_i)$ holds for every $i \in [n]$,

– a *reversed half-space-walk*: $h(v_0) > h(v_i)$ holds for for all $i \in [n]$.

Note in particular that bridges are HSWs and HSWs are SAWs. For any $v \in V(G)$ we denote by $C_{n,v}, B_{n,v}, \overline{B}_{n,v}, H_{n,v}$ and $\overline{H}_{n,v}$ the sets of SAWs, bridges, reversed bridges, HSWs and reversed HSWs of length n starting at v, respectively. By definition all of the above sets contain the walk of length 0 consisting of the single vertex v. Furthermore, denote by $c_{n,v}, b_{n,v}, \bar{b}_{n,v}, h_{n,v}$ and $\bar{h}_{n,v}$ the cardinalities of the respective sets $C_{n,v}, B_{n,v}, \overline{B}_{n,v}, H_{n,v}$ and $\overline{H}_{n,v}$. Because of symmetry, reversed bridges and reversed half-space-walks with respect to h are bridges and HSWs with respect to the "reflected" height function $-h$, so we state most results only for bridges and HSWs.

Let p be a walk on G. The *span* of p is defined as the maximal height difference of two vertices in p:

$$\operatorname{span}(p) = h_{\max}(p) - h_{\min}(p),$$

where

$$h_{\max}(p) = \max_{v \in p} h(v), \quad h_{\min}(p) = \min_{v \in p} h(v).$$

It is clear that for any bridge p, $\operatorname{span}(p) = h(v_n) - h(v_0)$.

The group Γ acts quasi-transitively on G and is h-difference invariant, so it is possible to define

$$c_n = \max_{v \in V(G)} c_{n,v}, \quad b_n = \min_{v \in V(G)} b_{n,v} \quad \text{and} \quad \bar{b}_n = \min_{v \in V(G)} \bar{b}_{n,v}.$$

Any SAW $(v = v_0, \ldots, v_{n+m}) \in C_{n+m,v}$ can be decomposed into a pair $(v_0, \ldots, v_n) \in C_{n,v}$ and $(v_n, \ldots, v_{n+m}) \in C_{m,v_n}$ of SAWs. Picking v such that $c_{n+m,v} = c_{n+m}$ results in

$$c_{n+m} = c_{n+m,v} \leq c_{n,v} c_m \leq c_n c_m,$$

so $(c_n)_{n \geq 0}$ is a sub-multiplicative sequence. On the other hand the concatenation of the bridges $(v = v_0, \ldots, v_n) \in B_{n,v}$ and $(v_n, \ldots, v_{n+m}) \in B_{m,v_n}$ results in the bridge $(v_0, \ldots, v_{n+m}) \in B_{n+m,v}$. Picking v such that $b_{n+m,v} = b_{n+m}$ yields

$$b_n b_m \leq b_{n,v} b_m \leq b_{n+m,v} = b_{n+m}.$$

Fekete's Lemma states that for every sub-additive sequence $(a_n)_{n \geq 0}$, the limit $\lim_{n \to \infty} \frac{a_n}{n}$ exists and is equal to $\inf_{n \geq 0} \frac{a_n}{n}$. An application of this lemma to the sub-additive sequences $(\log(c_n))_{n \geq 0}$ and $(-\log(b_n))_{n \geq 0}$ provides the existence of the limits

$$\mu(G) := \lim_{n \to \infty} c_n^{1/n}, \quad \beta(G,h) := \lim_{n \to \infty} b_n^{1/n}, \quad \overline{\beta}(G,h) := \lim_{n \to \infty} \bar{b}_n^{1/n}.$$

Here $\mu(G)$ depends only on the underlying graph G and is called the connective constant of G and $\beta(G,h)$ and $\overline{\beta}(G,h)$ depend on the graph and the chosen height function h and are called the bridge constant and reversed bridge constant of G with respect to h, respectively. We usually omit the graph and the height function if they are clear and just write μ, β and $\overline{\beta}$.

Trivially $b_n \leq c_n$, so we obtain

$$b_n \leq \beta^n \leq \mu^n \leq c_n, \quad n \geq 0, \tag{3.1}$$

and the analogue statement for $\bar{b}_n$ and $\overline{\beta}$. Hammersley showed in [29] that

$$\lim_{n\to\infty} c_{n,v}^{1/n} = \mu \quad \text{for every } v \in V(G)$$

and Grimmett and Li proved in [22] the similar statement

$$\lim_{n\to\infty} b_{n,v}^{1/n} = \beta \quad \text{for every } v \in V(G).$$

Let us denote by

$$F_{\text{SAW},v}(z) = \sum_{n\geq 0} c_{n,v} z^n \quad \text{and} \quad F_{\text{bridge},v}(z) = \sum_{n\geq 0} b_{n,v} z^n$$

the generating functions of self-avoiding walks and bridges, respectively. Then the Cauchy-Hadamard theorem implies that for any vertex $v \in V(G)$ the corresponding radii of convergence are given by

$$R_{\text{SAW}} := \frac{1}{\lim_{n\to\infty} c_{n,v}^{1/n}} = \frac{1}{\mu} \quad \text{and} \quad R_{\text{bridge}} := \frac{1}{\lim_{n\to\infty} b_{n,v}^{1/n}} = \frac{1}{\beta},$$

respectively. In Section 3.3 we make use of this fact to calculate the bridge constant of the grandparent graph by finding the radius of convergence of the corresponding generating function.

3.2 The bridge theorem

As mentioned before the main motivation was the following version of the bridge theorem, holding for graphs admitting a unimodular graph height function.

Theorem 3.2.1 (Grimmett and Li [22, Theorem 4.3])**.** *Let G be a locally finite, connected graph possessing a unimodular graph height function (h, Γ). Then $\mu(G) = \beta(G, h)$.*

As a simple consequence of this theorem the bridge constant $\beta(G, h)$ does not depend on the choice of the unimodular graph height function h. However, there are simple examples showing that unimodularity is necessary in this theorem, one of them being the grandparent graph, which will be discussed in Section 3.3.

The main result of the current chapter of this thesis is the following extension of this bridge theorem, holding without the requirement of unimodularity.

Theorem 3.2.2. *Let G be a locally finite, connected graph possessing a graph height function (h, Γ). Then*

$$\mu(G) = \beta_{\max}(G, h) := \max\{\beta(G, h), \overline{\beta}(G, h)\}. \tag{3.2}$$

One inequality is clear from (3.1), we only need to show $\mu \leq \beta_{\max}$. For convenience we first provide a detailed proof of the transitive case which we then generalise to the quasi-transitive case.

From now on let G be a locally finite, connected graph, let (h, Γ) be a graph height function on G and assume that the group Γ acts transitively on G. Then the value of $c_{n,v}$ does not depend on v and is therefore equal to c_n. Moreover elements of Γ map bridges onto bridges and HSWs onto HSWs, implying that also $b_{n,v}$ is equal to b_n for every v and $h_{n,v}$ does not depend on v, so we can omit v in the notation.

For simplicity we fix some vertex o of G with $h(o) = 0$ and write C_n, B_n, H_n and $\overline{B}_n, \overline{H}_n$ for the sets of n-step SAWs, bridges, HSWs and their reversed versions starting at o, respectively. Moreover for every $v \in V(G)$ we fix some element $\gamma_v \in \Gamma$ with $\gamma_v(o) = v$. The *γ-concatenation* of two walks $p_1 = (o, v_1, \ldots, v_m)$, $p_2 = (o, w_1, \ldots, w_n)$ is defined as the walk

$$p_1 *_\gamma p_2 := (o, v_1, \ldots, v_m, \gamma_{v_m}(w_1), \ldots, \gamma_{v_m}(w_n)).$$

In other words, $p_1 *_\gamma p_2$ is the concatenation $p_1\gamma_{v_m}(p_2)$ of the walks p_1 and $\gamma_{v_m}(p_2)$ as defined in Section 2.1. Similarly, the *γ-decomposition* of p_1 at v_l is defined to provide the two walks $(o, v_1, \ldots, v_l)$ and $(\gamma_{v_l}^{-1}(v_l), \ldots, \gamma_{v_l}^{-1}(v_n))$, both of them starting at o.

Denote by $B_n(a)$ the set of bridges in B_n having span $a \geq 0$ and by $b_n(a)$ the cardinality of this set. Note that $b_0(0) = 1$ and for $d = d(h, \Gamma)$ from Definition 3.1.1 it trivially holds that $b_n(a) = 0$ for $a > dn$ because the height distance per step is at most d. It follows that

$$b_n = \sum_{a=0}^{dn} b_n(a). \tag{3.3}$$

Take any HSW $p = (v_0, v_1, \ldots, v_n)$ of length $n \geq 1$. We use the following iterative process to γ-decompose p into an alternating sequence of bridges and reversed bridges: Let $i_0 = 0$ and in step $j \geq 1$ let

$$a_j = \max_{i \in \{i_{j-1}, \ldots, n\}} \left| h(v_i) - h(v_{i_{j-1}}) \right|,$$

and let i_j be the largest index in $\{i_{j-1}, \ldots, n\}$, where this maximum is attained. Then the sub-walk $(v_{i_{j-1}}, \ldots, v_{i_j})$ of p has span a_j and is a bridge if j is odd and a reversed bridge if j is even.

By construction the span decreases in every step, so $a_1 > \cdots > a_k > 0$. We denote by $H_n(a_1, \ldots, a_k)$ the set of HSWs in H_n decomposing into an alternating sequence $p_1, p_2, \ldots, p_k$ of bridges and reversed bridges of spans $a_1, \ldots, a_k$ and by $h_n(a_1, \ldots, a_k)$ its cardinality. It is clear that for $n \geq 1$

$$h_n = \sum_{k>0} \sum_{a_1 > \cdots > a_k > 0} h_n(a_1, \ldots, a_k). \tag{3.4}$$

Moreover the equality $h_n(a_1) = b_n(a_1)$ follows directly from the definition.

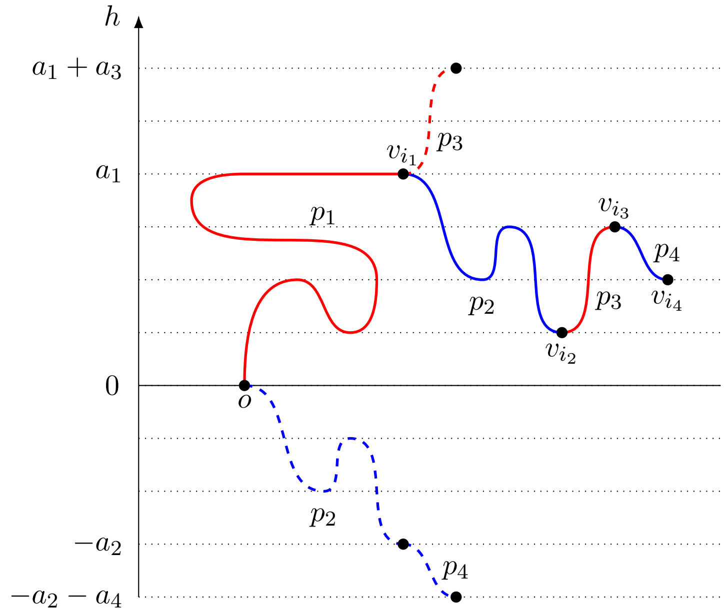


Figure 3.1: Decomposition of a HSW p into bridges (red) and reversed bridges (blue) and construction of p_+ and p_- (dashed).

Lemma 3.2.3. *Let $n, k \geq 1$ and $a_1 > a_2 > \cdots > a_k > 0$. Then*

$$h_n(a_1, \ldots, a_k) \leq \sum_{m=0}^{n} b_m(a_1 + a_3 + \ldots)\bar{b}_{n-m}(a_2 + a_4 + \ldots). \tag{3.5}$$

Proof. Let $p \in H_n(a_1, \ldots, a_k)$. Use the decomposition process described before to construct a pair (p_+, p_-) consisting of a bridge and a reversed bridge, both starting at o. Begin by γ-decomposing p into the walks $p_1, \ldots, p_k$ such that the span of p_i is a_i for every i and p_i is a bridge if i is odd and a reversed bridge otherwise. Let $p_+ = p_1 *_\gamma p_3 *_\gamma \ldots$ be the γ-concatenation of all bridges p_i (i odd) and $p_- = p_2 *_\gamma p_4 *_\gamma \ldots$ be the γ-concatenation of all reversed bridges p_i (i even). This construction can be seen in Figure 3.1.

Clearly p_+ is a bridge and its span is $a_1 + a_3 + \ldots$ and p_- is a reversed bridge and its span is $a_2 + a_4 + \ldots$. Moreover from the knowledge of the sequence $a_1, \ldots, a_k$ and the two walks p_+ and p_- the original walk p can be uniquely reconstructed, so the process of obtaining the pair (p_+, p_-) from p is injective. Let $m = |p_1| + |p_3| + \ldots$ be the sum of lengths of the odd-index sub-walks p_i. Then $p_+ \in B_m(a_1 + a_3 + \ldots)$ and $p_- \in \overline{B}_{n-m}(a_2 + a_4 + \ldots)$ and (3.5) follows. □

A *partition with distinct parts* of a positive integer A is a way to write A as a sum of distinct positive integers. Two partitions are considered the same if they differ only in the order of their summands. Denote by $p_D(A)$ the number of different partitions with distinct parts of the integer $A \geq 1$. For consistency let $p_D(0) = 1$. Hardy and

Ramanujan showed in [31] that for $A \to \infty$:

$$\log p_D(A) \sim \pi \left(\frac{A}{3}\right)^{1/2}. \tag{3.6}$$

Lemma 3.2.4. *Let $L > \pi\sqrt{d/3}$, where $d = d(h, \Gamma)$ is the maximal height distance between adjacent vertices of G. Then there is a constant $K > 0$ such that for all $n \geq 0$*

$$h_n \leq p_D(dn) \sum_{m=0}^{n} b_m \bar{b}_{n-m} \leq K e^{L\sqrt{n}} \beta_{\max}^n. \tag{3.7}$$

Proof. The statement trivially holds for $n = 0$. In the case $n > 0$ application of Lemma 3.2.3 in expression (3.4) and exchanging the finite sums yields

$$h_n \leq \sum_{m=0}^{n} \sum_{k>0} \sum_{a_1 > \cdots > a_k > 0} b_m(a_1 + a_3 + \dots)\bar{b}_{n-m}(a_2 + a_4 + \dots).$$

For given $A, B \geq 0$ we want to count the number of occurrences of the summand $b_m(A)\bar{b}_{n-m}(B)$ in the sum on the right-hand side. Clearly the total number of sequences $a_1 > \cdots > a_k > 0$ with $a_1 + a_3 + \cdots = A$ and $a_2 + a_4 + \cdots = B$ is bounded from above by the number $p_D(A + B)$. Using that the height distance per step is at most d, it follows that

$$h_n \leq \sum_{m=0}^{n} \sum_{A=0}^{dm} \sum_{B=0}^{d(n-m)} p_D(A + B) b_m(A) \bar{b}_{n-m}(B).$$

From $p_D(A + B) \leq p_D(dn)$ and (3.3) the first inequality in (3.7) follows:

$$\begin{aligned} h_n &\leq p_D(dn) \sum_{m=0}^{n} \left(\sum_{A=0}^{dm} b_m(A)\right) \left(\sum_{B=0}^{d(n-m)} \bar{b}_{n-m}(B)\right) \\ &= p_D(dn) \sum_{m=0}^{n} b_m \bar{b}_{n-m}. \end{aligned}$$

The second inequality in (3.7) follows from $b_n \leq \beta^n$ (see (3.1)) and the existence of a constant $K > 0$ such that

$$(n + 1)p_D(dn) \leq K e^{L\sqrt{n}}$$

for every $n > 0$, which is a consequence of (3.6). □

Lemma 3.2.5. *Let $M > p\sqrt{2d/3}$. Then there is an integer N such that for all $n \geq N$*

$$c_n \leq e^{M\sqrt{n+1}} \beta_{\max}^{n+1}. \tag{3.8}$$

Proof. Let $p = (v_0, \dots, v_n)$ be any SAW of length n and l the maximal index such that $h(v_l) = h_{\min}(p)$. By the definition of graph height functions there is a neighbour v' of v_l

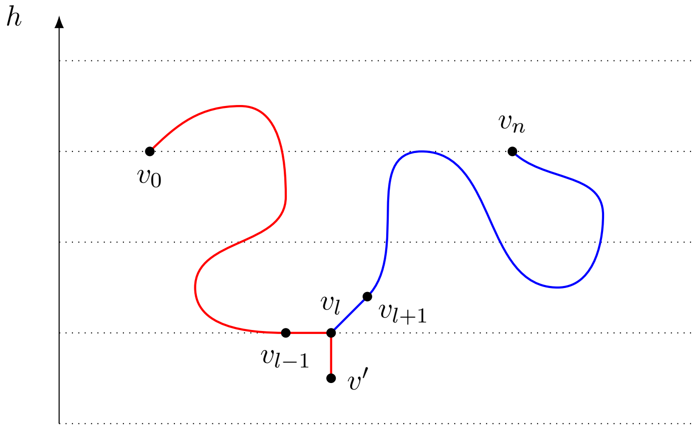


Figure 3.2: Decomposition of a SAW into a HSW starting at v' (red) and a HSW starting at v_l (blue).

with $h(v') < h(v_l)$. Hence $\gamma_{v_l}^{-1}(v_l, v_{l+1}, \dots, v_n)$ and $\gamma_{v'}^{-1}(v', v_l, v_{l-1}, \dots, v_0)$ are HSWs in H_{n-l} and H_{l+1} respectively. This construction is shown in Figure 3.2 and yields

$$c_n \leq \sum_{l=0}^{n} h_{n-l} h_{l+1}. \tag{3.9}$$

Let $\epsilon > 0$ such that $M - \epsilon > \pi\sqrt{2d/3}$. By Lemma 3.2.4 there is a $K > 0$ such that for every $n \geq 0$,

$$c_n \leq \sum_{l=0}^{n} K^2 \exp\left(\frac{M-\epsilon}{\sqrt{2}}\left(\sqrt{n-l} + \sqrt{l+1}\right)\right) \beta_{\max}^{n+1}.$$

Using this estimate and the inequality $\sqrt{a}+\sqrt{b} \leq \sqrt{2a+2b}$, which holds for all $a, b \in \mathbb{R}^+$, we obtain

$$c_n \leq (n+1)K^2 \exp\left((M-\epsilon)\sqrt{n+1}\right) \beta_{\max}^{n+1}. \tag{3.10}$$

For n large enough (3.8) follows. □

Now we have all necessary tools to finish the transitive case of the proof of Theorem 3.2.2. Using $\mu^n \leq c_n$ (see (3.1)) and Lemma 3.2.5, it follows that for $M > \pi\sqrt{2d/3}$ and n large enough

$$\mu^{n-1} e^{-M\sqrt{n}} \leq c_{n-1} e^{-M\sqrt{n}} \leq \beta_{\max}^n.$$

Applying the n-th root and sending n to infinity yields

$$\mu \leq \beta_{\max}.$$

We briefly discuss the additional steps to generalise the proof to the case where Γ acts quasi-transitively on the graph G. The following additional definitions and results from [22] are necessary.

Let the action of Γ on G admit m orbits and $\{o_1, \dots, o_m\}$ be a system of representatives of the orbits. Let $r = r(h, \Gamma)$ be the smallest non-negative integer such that for any $0 \le i, j \le m$ there is some $v_j \in \Gamma o_j$ and a bridge $q(i, j)$ of length at most r starting at o_i and ending at v_j, such that v_j is the unique vertex of maximal height in the walk. The walk obtained by going along $q(j, i)$ in the reversed direction (from v_j to o_i) is a reversed bridge and will be denoted by $\bar{q}(i, j)$. It has been shown in [22] (Proposition 3.2 and Proposition 4.2) that $r(h, \Gamma)$ exists for any graph height function and moreover, for any $v \in V(G)$,

$$b_{n,v} \le \beta^{n+r}, \quad n \ge 0. \tag{3.11}$$

From now on we denote for $v \in V(G)$ and $a \ge 0$ by $B_{n,v}(a)$ the set of bridges of span a starting at v and by $b_{n,v}(a)$ the cardinality of this set.

Furthermore, for any $v \in V(G)$ and $a_1 > \cdots > a_k > 0$ let $H_{n,v}(a_1, \dots, a_k)$ be the set of HSWs in $H_{n,v}$ decomposable into an alternating sequence of bridges and reversed bridges of spans $a_1, \dots, a_k$ as described in the transitive case and let $h_{n,v}(a_1, \dots, a_k)$ be its cardinality. Lemma 3.2.3 can be replaced by the following:

Lemma 3.2.6. *Let $n, k \ge 1$, $a_1 > a_2 > \cdots > a_k > 0$ and $v \in V(G)$. Then*

$$\begin{aligned} h_{n,v}(a_1, \dots, a_k) \le (r+1)^{k-1} \sum_{m=0}^{n} &\left(\sum_{s=0}^{kr} \sum_{t \ge 0} b_{m+s,v}(a_1 + a_3 + \cdots + t) \right) \\ &\left(\sum_{s'=0}^{kr} \sum_{t' \ge 0} \bar{b}_{n-m+s',v}(a_2 + a_4 + \cdots + t') \right). \end{aligned} \tag{3.12}$$

Proof. Given a walk p in $H_{n,v}(a_1, \dots, a_k)$, we decompose it into the alternating sequence $p_1, p_2, \dots, p_k$ of bridges and reversed bridges as in the transitive case. The main difficulty is that it is not always possible to concatenate the bridges p_l and p_{l+2} directly, as p_l^+ and p_{l+2}^- may lie in different orbits. Let $(i(0), \dots, i(k-2))$ and $(j(0), \dots, j(k-2))$ be sequences defined such that $v \in \Gamma o_{i(0)}$, $p_l^+ \in \Gamma o_{i(l)}$ for $l \ge 1$ and $p_{l+2}^- \in \Gamma o_{j(l)}$ for $l \ge 0$. Let

$$q_l = \begin{cases} q(i(l), j(l)) & \text{if } l \text{ is odd,} \\ \bar{q}(i(l), j(l)) & \text{if } l \text{ is even.} \end{cases}$$

Define p_+ as the concatenation of the bridges $p_1, q_1, p_3, q_3, \dots$ (odd indices) and p_- as the concatenation of the reversed bridges $q_0, p_2, q_2, p_4, \dots$ (even indices). This construction can be seen in Figure 3.3.

Let m be the sum of the lengths of p_i having odd index i. Then p_+ is in $B_{m+s,v}(a_1 + a_3 + \cdots + t)$ for some $0 \le s \le kr$ and $t \ge 0$ and p_- is in $B_{n-m+s',v}(a_2 + a_4 + \cdots + t')$ for some $0 \le s' \le kr$ and $t' \ge 0$ as every q-walk has length at most r. The construction is not injective because for a given pair (p_+, p_-) we might not be able to directly identify the contained q-walks. However, the length of any q-walk is at most r, so there are at most $(r+1)$ possible positions per q-walk. Therefore any pair (p_+, p_-) can be constructed at most $(r+1)^{k-1}$ times and (3.12) follows. ☐

Lemma 3.2.4 is replaced by the following:

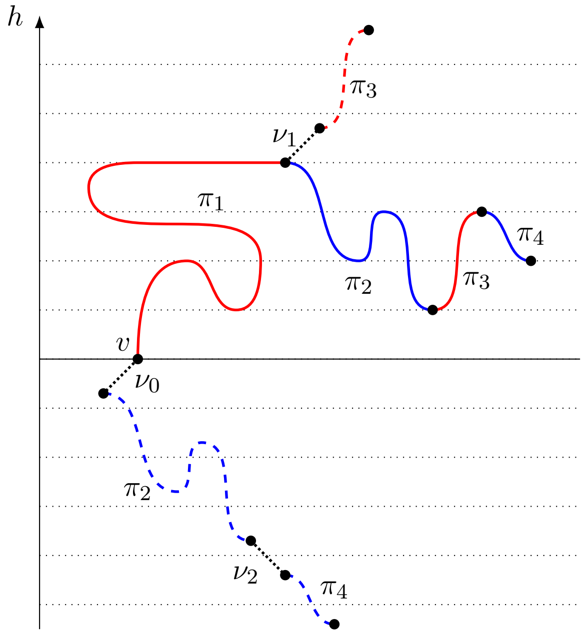


Figure 3.3: Decomposition of a HSW into bridges (red) and reversed bridges (blue) and construction of π_+ and π_- (dashed). Dotted lines are ν-walks.

Lemma 3.2.7. *There is a constant $L > 0$ such that for any $n \geq 0$ and $v \in V(G)$*

$$h_{n,v} \leq e^{L\sqrt{n}} \beta_{\max}^n. \tag{3.13}$$

Proof. We begin with the observation that for all integers $a, l \geq 0$

$$\sum_{t \geq 0} b_{l,v}(a+t) \leq b_{l,v}. \tag{3.14}$$

Starting with (3.4) and using Lemma 3.2.6 and (3.14) yields

$$h_{n,v} \leq \sum_{k>0} \sum_{\substack{a_1 > \cdots > a_k > 0 \\ a_1 + \cdots + a_k \leq dn}} (r+1)^k \sum_{m=0}^{n} \left(\sum_{s=0}^{kr} b_{m+s,v} \right) \left(\sum_{s'=0}^{kr} \bar{b}_{n-m+s',v} \right).$$

Let Δ denote the maximum over all degrees of vertices of G. Then it is easy to see that $\beta_{\max} \leq \Delta$. Furthermore, any partition of an integer A with k distinct parts satisfies $k(k+1) \leq 2A$ and therefore $k < \sqrt{2A}$. Using these observations, we end up with

$$\begin{aligned} h_{n,v} &\leq \sum_{k>0} \sum_{\substack{a_1 > \cdots > a_k > 0 \\ a_1 + \cdots + a_k \leq dn}} (r+1)^k (n+1)(kr+1)^2 \beta_{\max}^{n+2kr+2r} \\ &\leq dn\, p_D(dn)(r+1)^{\sqrt{2dn}}(n+1)(\sqrt{2dn}\, r+1)^2 \Delta^{2\sqrt{2dn}\, r+2r} \beta_{\max}^n. \end{aligned}$$

Finally, applying (3.6), for B large enough (3.13) follows. $\square$

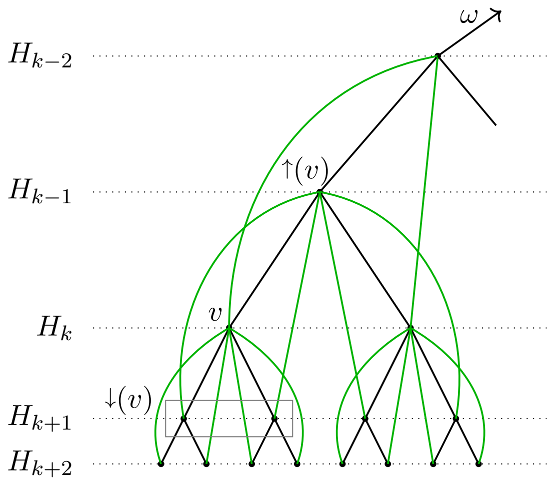


Figure 3.4: The grandparent graph G_{GP}. Edges connecting vertices to their grandparents are green, while the edges of the original 3-regular tree are black.

Finally we obtain the analogue to Lemma 3.2.5, which which can be proved in the same way.

Lemma 3.2.8. *There is a constant $M > 0$ such that for any $n \geq 0$ and $v \in V(G)$*

$$c_{n,v} \leq e^{M\sqrt{n}} \beta_{\max}^n. \tag{3.15}$$

From this statement it follows directly that $\mu \leq \beta_{\max}$, which finishes the proof of Theorem 3.2.2.

3.3 Bridges in the grandparent graph

In this section we discuss the grandparent graph G_{GP} introduced in Example 2.1.4 as an example of a graph not possessing any unimodular graph height function. We calculate the bridge constants and use the bridge theorem to obtain the connective constant.

Recall that the grandparent graph is constructed as follows. Fix some end ω of the infinite 3-regular tree T_3, let the graph "hang down" from the end ω and treat the vertex set of G_{GP} as the union of horizontal layers H_k, $k \in \mathbb{Z}$ as shown in Figure 3.4. Every vertex $v \in H_k$ is adjacent to one vertex in H_{k-1}, called parent of v and denoted by $\uparrow(v)$, and two vertices in H_{k+1}, called children of v. Write $\uparrow^k(v)$ for the k-th predecessor of v, that is the vertex obtained by applying $\uparrow$ k times to v. Furthermore denote by $\downarrow^k(v)$ the set of all vertices u such that $\uparrow^k(u) = v$. We add all undirected edges connecting a vertex v to its grandparent $\uparrow^2(v)$ to the graph and end up with the graph G_{GP} as shown in Figure 3.4.

Clearly the group $\mathrm{AUT}(G_{\mathrm{GP}})$ acts transitively on G_{GP}. Soardi and Woess showed in [54] that every graph automorphism fixes the end ω. Moreover they observed that for any vertex v of G_{GP}, any automorphism fixing v also fixes $\uparrow(v)$. On the other hand, there are automorphisms fixing $\uparrow(v)$ and mapping v to its sibling, so that

$$|\mathrm{AUT}(G_{\mathrm{GP}})_v \uparrow(v)| = 1 \neq 2 = \left|\mathrm{AUT}(G_{\mathrm{GP}})_{\uparrow(v)} v\right|,$$

thus the action of $\mathrm{AUT}(G_{\mathrm{GP}})$ is not unimodular. In particular G_{GP} is not a Cayley graph and it cannot admit a unimodular graph height function.

G_{GP} admits the obvious graph height function $(h, \mathrm{AUT}(G_{\mathrm{GP}}))$, where the map h associates to every vertex v the index k of the layer H_k containing v.

We fix a vertex o and use generating functions to count bridges starting at o and calculate the bridge constants $\beta(G_{\mathrm{GP}}, h)$ and $\overline{\beta}(G_{\mathrm{GP}}, h)$. From the structure of the graph it is intuitively clear that there are more bridges then reversed bridges starting at o, so we start by counting bridges. For $a \geq 0$ let $\mathcal{B}_{a,v}$ be the set of all bridges of span a starting at a vertex v and $\mathcal{B}_a(x)$ be the ordinary generating functions corresponding to this class, which is independent of v and given by

$$\mathcal{B}_a(x) = \sum_{n \geq 0} b_n(a) x^n.$$

Then $\mathcal{B}_a(x)$ is a polynomial for every $a \geq 0$ and it is not hard to obtain

$$\mathcal{B}_0(x) = 1, \quad \mathcal{B}_1(x) = 2x, \quad \mathcal{B}_2(x) = 4x + 4x^2 + 4x^3.$$

By (3.3), the generating function $F_{\mathrm{bridge}}(x)$ counting all bridges is

$$F_{\mathrm{bridge}}(x) = \sum_{n \geq 0} b_n x^n = \sum_{a \geq 0} \mathcal{B}_a(x).$$

For $a \geq 3$ recursively count all bridges $p = (v_0, v_1, \ldots, v_m) \in \mathcal{B}_{a,o}$. The different types of bridges discussed are shown in Figure 3.5. Every bridge p starts at $v_0 = o$, so either $v_1 \in \downarrow(o)$ or $v_1 \in \downarrow^2(o)$. If $v_1 \in \downarrow(o)$, then $(v_1, \ldots, v_m)$ of p must be a bridge in $\mathcal{B}_{a-1,v_1}$ (type 1).

Let now $v_1 \in \downarrow^2(o)$. We distinguish the following sub-cases: If p does not contain $\uparrow(v_1)$, the walk $(v_1, \ldots, v_m)$ must be a bridge in $\mathcal{B}_{a-2,v_1}$ (type 2).

Otherwise, there is some index $l > 1$ such that $v_l = \uparrow(v_1)$ and we can decompose the walk at v_l to obtain walks $p_1 = (v_0, \ldots, v_l)$ and $p_2 = (v_l, \ldots, v_m)$. Whenever a SAW reaches a descendent of a vertex v after containing both v and $\uparrow(v)$, it cannot reach a predecessor of v anymore. This means that p_1 can have one of two possible shapes, depending on the parity of l. Let $l = 2k$ in the case where l is even and $l = 2k+1$ otherwise. In both cases we have $v_i \in \downarrow^2(v_{i-1})$ for every $i \in [k-1]$. The k-th step satisfies $v_k = \uparrow(v_{k-1})$ if l is odd and $v_k \in \downarrow(v_{k-1})$ for even l. The walk concludes with the steps $v_i = \uparrow^2(v_{i-1})$ for every index $k+1 \leq i \leq l$. We call walks with this shape U-walks.

Since the span of p is a, the span of the U-walk p_1 can be at most a. Note that for any v there are 2 vertices in $\downarrow(v)$ and 4 vertices in $\downarrow^2(v)$. The generating function of U-walks of span at most a is thus given by

$$\mathcal{U}_a(x) = 4x^2 + 8x^3 + \cdots + (2x)^a.$$

There are three possible ways how the second part $p_2 = (v_l, \ldots, v_m)$ of p may look:

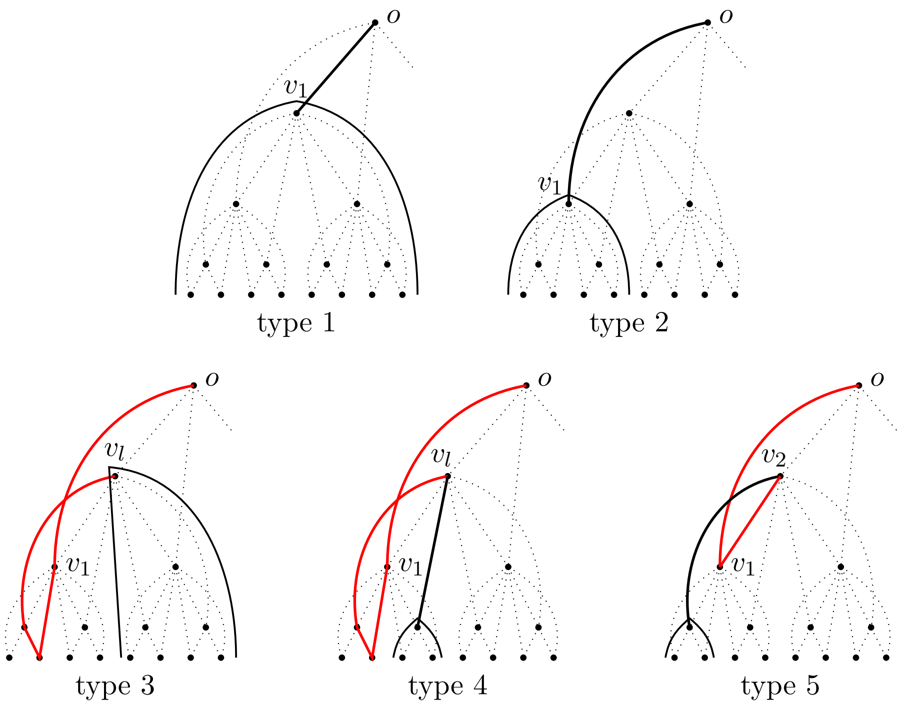


Figure 3.5: The five different types of walks in B_a^d. U-walks are drawn red.

– The walk p_2 is in $\mathcal{B}_{a-1,\uparrow(v_1)}$ and does not contain a vertex in $\downarrow(v_1)$. Precisely half of the bridges in $\mathcal{B}_{a-1,\uparrow(v_1)}$ satisfy this condition. (type 3)

– The walk p_2 contains a vertex of $\downarrow(v_1)$. If $l \geq 3$ then $v_{l-1} \in \downarrow(v_1)$, so v_{l+1} has to be the other vertex in $\downarrow(v_1)$ and $(v_{l+2}, \ldots, v_m) \in \mathcal{B}_{a-3,\uparrow(v_{l+2})}$. (type 4)

– Otherwise $l = 2$ and thus both vertices in $\downarrow(v_1)$ are available for v_{l+1}. In this case $(v_{l+2}, \ldots, v_m)$ must be in $\mathcal{B}_{a-3,\uparrow(v_{l+2})}$. (type 5)

Translating these combinatorial observations into generating functions yields the recursive formula

$$\mathcal{B}_a(x) = f_a(x)\mathcal{B}_{a-1}(x) + g(x)\mathcal{B}_{a-2}(x) + h_a(x)\mathcal{B}_{a-3}(x), \tag{3.16}$$

where for $x \neq 1/2$,

$$\begin{aligned} f_a(x) &= 2x + 2x^2 \frac{1-(2x)^{a-1}}{1-2x}, \\ g(x) &= 4x, \\ h_a(x) &= 8x^3 + 8x^4 \frac{1-(2x)^{a-2}}{1-2x}. \end{aligned}$$

Fix some x_0 in the real interval $[0, 1/2)$. Then $(\mathcal{B}_a(x_0))_{a \geq 0}$ is a sequence in $\mathbb{R}_{\geq 0}$. Equation (3.16) provides a recursive definition with non-negative coefficients. In particular, convergence of the series induced by the sequence $(\mathcal{B}_a(x_0))_{a\geq 0}$ is strongly related to the sum of coefficients of its recursive definition,

$$F_{\text{bridge}}(x_0) = \sum_{a\geq 0} \mathcal{B}_a(x_0) \begin{cases} < \infty & \text{if } \lim\limits_{a\to\infty} (f_a(x_0) + g(x_0) + h_a(x_0)) < 1, \\ = \infty & \text{if } \lim\limits_{a\to\infty} (f_a(x_0) + g(x_0) + h_a(x_0)) > 1. \end{cases}$$

From this observation it follows that the radius of convergence R_{bridge} of $F_{\text{bridge}}(x)$ is the threshold value for x_0, which can be found as the smallest positive root of the polynomial

$$1 - 8x + 10x^2 - 8x^3 + 8x^4.$$

So in particular the bridge constant is an algebraic number, which is approximately

$$\beta(G_{\text{GP}}, h) \approx 6.64993.$$

A similar construction can be used to calculate the reversed bridge constant. For this we define the generating functions $\overline{\mathcal{B}}_a(x)$ counting reversed bridges of span a as above. The main difference is that we obtain the following recursive formula by looking at the different cases for the final parts of bridges of span a:

$$\overline{\mathcal{B}}_a(x) = \overline{f}_a(x)\overline{\mathcal{B}}_{a-1}(x) + \overline{g}(x)\overline{\mathcal{B}}_{a-2}(x) + \overline{h}_a(x)\overline{\mathcal{B}}_{a-3}(x),$$

where for $x \neq 1/2$,

$$\begin{aligned} \overline{f}_a(x) &= x + x^2 \frac{1 - (2x)^{a-2}}{1 - 2x}, \\ \overline{g}(x) &= x, \\ \overline{h}_a(x) &= x^3 + x^4 \frac{1 - (2x)^{a-3}}{1 - 2x}. \end{aligned}$$

The radius of convergence of the generating function of reversed bridges is the smallest positive root of the polynomial

$$1 - 4x + 3x^2 - x^3 + x^4$$

and its reciprocal is the reversed bridge constant

$$\overline{\beta}(G_{\text{GP}}, h) \approx 3.10380.$$

As an application of the bridge theorem (Theorem 3.2.2) we obtain the connective constant of G_{GP},

$$\mu(G_{\text{GP}}) = \max\{\beta(G_{\text{GP}}, h), \overline{\beta}(G_{\text{GP}}, h)\} \approx 6.64993.$$

4 Context-free language of self-avoiding walks

In this chapter we introduce the apparently new approach of connecting self-avoiding walks with the theory of formal languages. The basic setting is as follows. Recall that an edge-labelled graph (G, ℓ), consists of a graph G and a labelling $\ell : E(G) \to \mathbf{\Sigma}$ assigning to every oriented edge e a label $\ell(e)$ of some given alphabet $\mathbf{\Sigma}$. The labelling is assumed to be deterministic, that is, different edges with the same initial vertex have distinct labels. Additionally, we always assume that the group $\mathrm{AUT}(G, \ell)$ of all ℓ-preserving graph automorphisms of G acts quasi-transitively on G; we say that G is a quasi-transitive edge-labelled graph.

The *language of self-avoiding walks* on a graph G starting at some given vertex o is

$$L_{\mathrm{SAW},o}(G) = L(\mathcal{P}_{\mathrm{SAW},o}),$$

where $\mathcal{P}_{\mathrm{SAW},o}$ denotes the set of all self-avoiding walks on G starting at o. Our main result completely characterises the set of all graphs having a regular or context-free language of self-avoiding walks:

Theorem 4.0.1. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph. Then for any choice of $o \in V(G)$, the following holds.*

(i) $L_{\mathrm{SAW},o}(G)$ *is regular if and only if all ends of G have size* 1*.*

(ii) $L_{\mathrm{SAW},o}(G)$ *is context-free if and only if all ends of G have size at most* 2*. In this case,* $L_{\mathrm{SAW},o}(G)$ *is unambiguous context-free.*

As already mentioned, the *word problem* of finitely generated groups was an important motivation for our studies. The term word problem describes the algorithmic problem of deciding whether two words over a given finite set S of generators of a group Γ represent the same element. Clearly this is equivalent to asking whether a given word represents the group identity of Γ. Looking at the Cayley graph $\mathrm{Cay}(\Gamma, S)$, where every edge is labelled with its corresponding generator, the word problem appears in a natural way. As discussed before, any given word corresponds to a labelled walk in $\mathrm{Cay}(\Gamma, S)$ starting at 1_Γ and the word corresponds to the identity if and only if its corresponding walk is closed, meaning that it also ends at 1_Γ. The term "word problem" often also stands for the language of all words over a given generating set S representing the group identity in Γ.

The word problem was first introduced in 1911 by Dehn [8] as one of three fundamental problems in the theory of infinite groups, the other two being the conjugacy problem and the group isomorphism problem. One year later he invented an algorithm [9] that solves both the word and conjugacy problem for the fundamental groups of closed orientable 2-dimensional manifolds of genus at least 2. Later, this algorithm became known as Dehn's algorithm and was extended and applied by various authors to a wide range of group theoretic decision problems.

Anisimov showed in [2] that the word problem is regular if and only if the group is finite, and this extends to quasi-transitive labelled graphs. In the ground-breaking work [46], Muller and Schupp showed that the word problem is context-free if and only if the group is virtually free. In particular, regularity, respectively context-freeness of the word problem are group invariants not depending on the specific generating set. In the subsequent work [47], context-free labelled graphs were defined via structural properties (not necessarily quasi-transitive), and these are precisely the (deterministically) labelled graphs for which the language of closed walks is context-free; see [6] by Ceccherini-Silberstein and Woess. For further work on language-theoretic issues related with groups, see e.g. Pélecq [48], as well as [5], [62], and for a new proof of the main result of [46] and related material, Dieckert and Weiss [11].

Applied to a Cayley graph of a group, Theorem 4.0.1 says that the group is virtually free if the language of self-avoiding walks is context-free, but the latter property is not a group invariant.

The inspiration for the present work came from the note [19] by Gilch and Müller, where they determined the SAW-generating function $F_{\mathrm{SAW},o}(z)$ for free products of finite graphs – an instance of the case where the language of self-avoiding walks is regular. Furthermore, in the computation of $F_{\mathrm{SAW},o}(z)$ for the bi-infinite ladder graph by Zeilberger in [63], a context-free grammar is inherent although not mentioned or used directly.

This chapter is organised as follows: In Section 4.1, we provide the necessary background on the end space of graphs and discuss strips in locally finite graphs, that is, two-ended quasi-transitive subgraphs. Their ends have the same finite size – the size of the strip. In the quasi-transitive case, if there is an end of finite size m, there must be a strip of the same size. Furthermore, if there is a thick end which is fixed by some non-elliptic automorphism (one not fixing a finite subset of $V(G)$), then there are strips of arbitrary size. In Section 4.2, the Pumping Lemmas for regular, respectively context-free languages are used to show the following. If G is quasi-transitive and contains a strip of size 2, then $L_{\mathrm{SAW},o}(G)$ cannot be regular, and if it contains a strip of size 3, then $L_{\mathrm{SAW},o}(G)$ cannot be context-free.

Thus, we are left with considering graphs whose ends have size at most 2. In Section 4.3 we first consider the case when all ends have size 1. Then the cut-vertex tree decomposition of G has finite blocks, and we derive that $L_{\mathrm{SAW},o}(G)$ is regular. If all ends have size 2, then we use the 3-block tree decomposition of G of Droms, Servatius and Servatius [12] to construct an unambiguous context-free grammar for $L_{\mathrm{SAW},o}(G)$. (Alternatively, one might use the vertex cuts of Dunwoody and Krön [15].) To conclude, if G contains ends of both sizes 2 and 1, one can combine 2-connectedness of the (possibly infinite) blocks of the cut-vertex tree decomposition with the method of the preceding case (ends of size 1) to get context-freeness.

In the final Section 4.4, we start with a discussion of implications and future work. We also recall some context-free examples, including one of Lindorfer [42], and provide two additional more detailed examples beneficial for a better understanding of the constructions used in the proofs in Section 4.3. In one of these examples the SAW-generating function is algebraic over $\mathbb{Q}$, but not rational.

4.1 Ends and strips in locally finite graphs

Recall that for our language-theoretic approach, it is convenient to consider the edge set $E(G)$ of our (locally finite, connected) graph G as being directed and with an involution $e \mapsto \bar{e}$ which inverts the orientation. When speaking about ends, it is sufficient to identify each pair of oppositely oriented edges with the same endpoints with one non-oriented edge. We shall frequently switch back and forward between these two viewpoints.

Recall from Chapter 2 that the ends of a graph G are defined as equivalence classes of rays contained in G, where two rays are equivalent if they cannot be separated by a finite set of vertices. The size of an end is defined as the maximum number of disjoint rays representing the end. A ray ρ is said to end up in a given (infinite) subset of $V(G)$, if all but finitely many vertices of ρ are contained in the subset. If ω is an end and $K \subseteq V(G)$ is finite, the unique component of $G - K$ in which all representing rays of ω end up is denoted $C(K, \omega)$. We say that the end ω belongs to the component $C(K, \omega)$. In a similar way we denote the component containing a given vertex $v \in V(G)$ by $C(K, v)$.

A *defining sequence* for an end ω consists of a sequence $(K_n)_{n\geq 0}$ of finite subsets of $V(G)$ such that $C(K_{n-1}, \omega) \supseteq K_n \cup C(K_n, \omega)$ for each n. As a consequence of Menger's theorem, any end ω of size m has a defining sequence K_n of size m, that is, $|K_n| = m$ for all n and there cannot be a defining sequence of smaller size.

Denote by ∂G the set of ends of the graph G and let $\widehat{G} := V(G) \cup \partial G$. A topology on $\widehat{G}$ is given as follows. In the above notation, for $\widehat{v} \in \widehat{G}$ let $\widehat{C}(K, \widehat{v})$ be the union of $C(K, \widehat{v})$ with the set of all ends whose rays end up in the set $C(K, \widehat{v})$. In other words, $\widehat{C}(K, \widehat{v})$ contains all vertices of $C(K, \widehat{v})$ and additionally all ends ω belonging to $C(K, \widehat{v})$, that is all ends such that $C(K, \omega) = C(K, \widehat{v})$. Then for any defining sequence $(K_n)_{n\geq 0}$ of an end ω, the family $(\widehat{C}(K_n, \omega))_{n\geq 0}$ is a neighbourhood base of ω. If $v \in V(G)$, we can take for K the finite set of neighbours of v in G to see that the topology is discrete on $V(G)$. With this topology, the space ∂G of ends is the boundary of a compactification of G.

Recall that a quasi-isometry between two metric graphs G and H with graph distances d_G and d_H is a mapping $\varphi : V(G) \to V(H)$ such that there are constants $A > 0$, $B, B' \geq 0$ such that for all $x_1, x_2 \in V(G)$ and $y \in V(H)$,

$$A^{-1} d_G(x_1, x_2) - B \leq d_H(\varphi x_1, \varphi x_2) \leq A\, d_G(x_1, x_2) + B \quad \text{and} \quad d_H(y, \varphi G) \leq B'. \tag{4.1}$$

The following lemma follows with some small additional effort from Lemma 21.3 and Lemma 21.4 in [61] and their proofs. The subscripts G and H refer to the respective graphs, their metrics, and so on. In particular, the maximum distance of two vertices in $K \subseteq V(G)$ in the graph G is denoted $\mathrm{diam}_G(K)$.

Lemma 4.1.1. *Let G and H be two simple, connected graphs with bounded vertex degrees. If $\varphi : V(G) \to V(H)$ is a quasi-isometry, then it extends to a continuous mapping $\widehat{G} \to \widehat{H}$ which restricts to a homeomorphism between the spaces of ends ∂G and ∂H.*

There is an increasing function $\theta : \mathbb{N} \to \mathbb{N}$ with the following property: if $\omega \in \partial G$ and $K \subseteq V(G)$ with $\operatorname{diam}_G(K) = k$ then there is $\overline{K} \subseteq V(H)$ with

$$\varphi K \subseteq \overline{K}, \quad \operatorname{diam}_H(\overline{K}) \leq \theta(k), \quad \text{and} \quad \varphi C_G(K, \omega) \supseteq C_H(\overline{K}, \varphi\omega) \cap \varphi G.$$

In particular, if ω has a defining sequence $(K_n)_{n\geq 0}$ with $\operatorname{diam}_G(K_n) \leq k < \infty$ for every n then $\varphi\omega$ has a defining sequence $(\overline{K}_n)_{n\geq 0}$ with $\operatorname{diam}_H(\overline{K}_n) \leq \theta(k)$, and if ω is a thick end, then so is $\varphi\omega$.

Ends having a defining sequence of finite bounded diameter are called *slim* ends. Clearly in any locally finite graph every thin end is slim. Thomassen and Woess showed in Theorem 4.4 of [56] that also the converse holds as long as the graph is transitive and their proof can be easily generalised to the quasi-transitive case. We briefly explain how this can be done. A *tight k-vertex-cut* of a connected graph G is a set K of k vertices such that $G - K$ has at least two components such that every vertex of K has a neighbour in each of the components. The following lemma is Proposition 4.2 in [56].

Lemma 4.1.2. *Let G be a connected, locally finite graph. Then for any $v \in V(G)$ and $k \in \mathbb{N}$, G has only finitely many tight k-vertex-cuts containing v.*

Having this lemma, we are ready to proof the mentioned generalisation of Theorem 4.4 of [56]. The main idea is that all but finitely many elements of a defining sequence of a thin end are tight vertex-cuts.

Lemma 4.1.3. *In every connected, locally finite, quasi-transitive graph G every thin end is slim. In particular, for any integer k there is some integer $m(k)$ such that any end of size k has a defining sequence of diameter at most $m(k)$.*

Proof. Let ω be a thin end of G of size k and let $(K_n)_{n\geq 0}$ be a defining sequence of ω consisting of sets of size k. Let $R_1, \ldots, R_k$ be pairwise disjoint rays belonging to ω. Then there is some $n_0 \geq 0$ such that each R_i, ($i \in [n]$) intersects every K_n for $n \geq n_0$. In particular K_n is a tight k-vertex-cut for $n > n_0$. Let R be a finite set of representatives of the vertex-orbits of the action of $\operatorname{AUT}(G)$ on G. Then by Lemma 4.1.2 there are only finitely many tight k-vertex-cuts containing a vertex of R, so their diameter is bounded from above by some integer $m(k)$. Now for any $n > n_0$, there are a graph automorphism $\gamma \in \operatorname{AUT}(G)$ and a tight tight vertex-cut K intersecting R such that $K_n = \gamma K$. We conclude that the diameter of the sets K_n is bounded by $m(k)$ for $n > n_0$ and thus ω is slim. □

Combining Lemma 4.1.1 and Lemma 4.1.3, in the case where both G and H are quasi-transitive, we obtain the following.

Corollary 4.1.4. *Let G and H be connected, quasi-transitive graphs and $\varphi : V(G) \to V(H)$ be a quasi-isometry. Then there is an increasing function $\eta : \mathbb{N} \to \mathbb{N}$ such that any end $\omega \in \partial G$ of size k maps to an end $\varphi\omega \in \partial H$ of size at most $\eta(k)$.*

Recall that any two Cayley graphs of the same finitely generated group Γ with respect to two different finite, symmetric generating sets are quasi-isometric. In particular, Corollary 4.1.4 tells us that the extension of the quasi-isometry ϕ serves as a bijection between ends of different Cayley graphs preserving thickness and thinness of ends. Making use of this, we can talk about the *ends* of a group, which are the ends of any (all) of its Cayley graphs.

The action of the automorphism group of a locally finite, connected graph extends in an obvious way to the space of ends. Recall that the automorphisms can be classified into 3 types. An automorphism $\gamma \in \mathrm{AUT}(G)$ is called

– *elliptic*, if its fixes a finite subset of $V(G)$,

– *parabolic*, if it is not elliptic and fixes a unique end of G, and

– *hyperbolic*, if it is not elliptic and fixes each of a unique pair of ends of G.

While this terminology was not used by Halin in [26], he showed that for a non-elliptic automorphism γ and every $v \in V(G)$ the sequence $(v, \gamma v, \gamma^2 v, \dots)$ uniquely defines an end of G called the *direction* of γ, denoted by $D(\gamma)$. The sequence $(v, \gamma v, \gamma^2 v, \dots)$ converges to the end $D(\gamma)$ with respect to the topology on $\widehat{G}$ and the end is fixed by γ. The following theorem serves as one of the main pillars for our results.

Theorem 4.1.5 (Halin [26, Theorem 9])**.** *Let γ be a non-elliptic automorphism acting on a simple, locally finite, connected graph G. Then the following holds:*

(i) $D(\gamma)$ and $D(\gamma^{-1})$ have the same size $m \in \mathbb{N} \cup \infty$.

(ii) $D(\gamma) \neq D(\gamma^{-1})$ (γ is hyperbolic) if and only if $m < \infty$.

(iii) There exist m disjoint γ-invariant double rays in G.

(iv) If γ is hyperbolic there are disjoint γ-invariant double rays $R_1, \dots, R_m$, a set $K \subseteq V(G)$ with $|K| = m$ and an integer k such that $(\gamma^{kn} K)_{n \geq 0}$ and $(\gamma^{-kn} K)_{n \geq 0}$ are defining sequences for $D(\gamma)$ and $D(\gamma^{-1})$ respectively. Each R_i ($i \leq m$) meets every $\gamma^{kn} K$ ($n \in \mathbb{Z}$) in precisely one vertex.

Definition 4.1.6. A locally finite, connected graph S is called a *strip* if it is quasi-transitive and has precisely two ends.

The structure of strips is well understood. We collect without proof those basic facts needed below. More about strips (including the original slightly different definition) can be found in [36] by Jung and Watkins and in [34] and [35] by Imrich and Seifter.

For a strip S there is some hyperbolic automorphism γ fixing each of the two ends ω^+ and ω^- of S. Thus by the previous theorem ω^+ and ω^- have the same finite size m. Moreover it provides

(S1) a finite set $K \subseteq V(S)$ with $|K| = m$, together with

(S2) an automorphism $\tau \in \mathrm{AUT}(S)$, such that $(\tau^n K)_{n\geq 0}$ and $(\tau^{-n}K)_{n\geq 0}$ are defining sequences for ω^+ and ω^-, respectively and

(S3) m disjoint, τ-invariant double rays, each of them intersecting every $\tau^n K$ in precisely one vertex.

In this situation we call S a τ-strip of size m. We use the same terminology if S is a subgraph of a bigger graph G, and $\tau \in \mathrm{AUT}(G)$ is an automorphism whose restriction to S has the above properties. By the following lemma, we may assume that the subgraph H of S spanned by $C(K,\omega^+) \setminus C(\tau K, \omega^+)$ is finite and connected, otherwise replace τ by τ^2.

Lemma 4.1.7. *Let S be a strip and let K and τ be as given in (S1) and (S2). Then the induced subgraph H of S spanned by $C(K,\omega^+) \setminus C(\tau^2 K, \omega^+)$ is connected.*

Proof. Let $R_1, \dots, R_m$ be the m disjoint, τ-invariant double rays from (S3). We first show that every vertex of H is connected to some R_i in H. Let P be a shortest path in $S[C(K,\omega^+)]$ connecting $v \in V(H)$ to some R_i. Then v is the only vertex of P contained in a ray R_j, so in particular it cannot contain a vertex of $\tau^2 K$. We conclude that P is contained in H.

It remains to show that any two rays R_i and R_j are connected by a path in H. Suppose for a contradiction that this is not the case. Let P be a shortest path in $S[C(K,\omega^+)]$ connecting any two double rays R_i and R_j, which are not connected in H. Let v_1 and v_2 be the endpoints of P and let $i_1, i_2 \geq 0$ such that $v_1 \in C(\tau^{i_1}K,\omega^+) \setminus C(\tau^{i_1+1}K,\omega^+)$ and $v_2 \in C(\tau^{i_2}K,\omega^+) \setminus C(\tau^{i_2+1}K,\omega^+)$. By exchanging v_1 and v_2 if necessary, we may assume that $i_2 \geq i_1$. Furthermore we may relabel the R_i such that v_1 and v_2 lie on the rays R_1 and R_2, respectively. Note that the interior vertices of P cannot lie on the double rays $R_1, \dots, R_m$: If an interior vertex v of P lies on some R_i, then by minimality of P the double ray R_i must be connected to both R_1 and R_2 in H, contradicting the assumption that R_1 and R_2 are not connected in H. It follows directly that $|i_2 - i_1| \leq 1$, otherwise P must intersect $\tau^{i_1+1}K$ and thus some R_i. But then the path P is contained in $\tau^{i_1}H$: if this is not the case, P contains either a vertex of $\tau^{i_1}K$ or a vertex of $C(\tau^{i_1+2}K,\omega^+)$. As before, both cases imply that P contains a vertex of some R_i. We conclude that $\tau^{-i_1}P$ is contained in H, contradicting our assumption. □

The following lemma refines the well-known argument that in a quasi-transitive graph with more than one end, the directions of hyperbolic automorphisms are dense in the space of ends.

Lemma 4.1.8. *Let G be simple, locally finite and connected graph and let $\Gamma \leq \mathrm{AUT}(G)$ act quasi-transitively on G. If G has a thin end of size m then G contains a τ-strip of size m for some $\tau \in \Gamma$.*

Proof. Let the end ω of G have size m, take m disjoint rays in ω and let $(K_n)_{n\geq 0}$ be a defining sequence for ω such that $|K_n| = m$ for all n and every K_n meets each of the m rays. Fix a finite set R of representatives of the orbits given by the action of Γ on

G. For $n \geq 0$ write $C_n = C(K_n, \omega)$ and let $\gamma_n \in \Gamma$ such that $\gamma_n K_n$ contains an element of R. Then every $\gamma_n K_n$ is a tight m-vertex cut, that is, a set of cardinality m such that $G - \gamma_n K_n$ has at least two components and every vertex of $\gamma_n K_n$ has at least one neighbour in each of them. Also $\gamma_n C_n$ is a component of $G - \gamma_n K_n$.

By Lemma 4.1.2 there are only finitely many tight m-vertex cuts containing some given vertex $v \in R$ and clearly every cut splits the graph in finitely many components, hence $\{(\gamma_n K_n, \gamma_n C_n) \mid n \geq 0\}$ is a finite set. Pick some $j > i \geq 0$ such that $(\gamma_i K_i, \gamma_i C_i) = (\gamma_j K_j, \gamma_j C_j)$ and let $\tau = \gamma_j^{-1}\gamma_i$. Then

$$C_i \supseteq \tau(K_i \cup C_i).$$

We show that τ is hyperbolic and its direction $D(\tau)$ has size m. Indeed, $(\tau^n K_i)_{n \geq 0}$ is a defining sequence for the unique end $D(\tau)$ belonging to each of the components $\tau^n C_i$, where $n \in \mathbb{Z}$. In particular $D(\tau)$ has size at most m. By Theorem 4.1.5, τ is hyperbolic. On the other hand there are m disjoint paths from K_i to τK_i. Their images under τ^n ($n \in \mathbb{Z}$) build m disjoint τ-invariant double rays $R_1, \dots, R_m$, implying that $D(\tau)$ has size m, as required. Add to those double rays a finite collection of finite paths connecting those double rays with each other and all their images under τ^n, $n \in \mathbb{Z}$. After possibly replacing τ by a suitable power τ^k, ($k \geq 1$), we obtain a subgraph of G which is a τ-strip of size m. □

Lemma 4.1.9. *Let G be a simple, locally finite, connected graph and $\Gamma \leq \mathrm{AUT}(G)$ act quasi-transitively on G. If Γ contains a parabolic element then for every $m \geq 1$, G contains a τ-strip of size at least m for some suitable $\tau \in \Gamma$.*

Proof. Suppose that $\gamma \in \Gamma$ is parabolic and ω is the unique end fixed by γ. By Theorem 4.1.5, there are countably many disjoint double rays R_n, $n \in \mathbb{N}$, which are invariant under γ and represent ω. For $m \in \mathbb{N}$, we can find some connected, finite subgraph K of G meeting each of $R_1, \dots, R_m$. Let v be a vertex of K contained in R_1. By local finiteness of G every ball of radius r around v contains only finitely many vertices, so we can find some $k \in \mathbb{N}$ such that $d_G(v, \gamma^l v) > 2\,\mathrm{diam}_G(K)$ for every $l \geq k$. We set $\tau = \gamma^k$ and conclude that the subgraph spanned by $R_1, \dots, R_m$ together with all the $\tau^n K$, $n \in \mathbb{Z}$, is a τ-strip and its two ends have size at least m. □

Remark 4.1.10. Whenever a graph G contains a τ-strip of size m, it also contains a τ^2-strip of any size in $l \in \{1, \dots, m-1\}$. This can be seen by deleting $m - l$ vertices from the set K in the definition of τ-strips and their images under τ^{2n} for $n \in \mathbb{Z}$.

4.2 Context-freeness and ends

In this section we prove the first half of Theorem 4.0.1:

Theorem 4.2.1. *Let G be a simple, connected, locally finite, deterministically edge-labelled graph on which $\mathrm{AUT}(G, \ell)$ acts quasi-transitively and let o be a vertex of G.*

(i) If $L_{\mathrm{SAW},o}(G)$ is context-free, every end of G has size at most 2.

(ii) If $L_{\mathrm{SAW},o}(G)$ is regular, every end of G has size 1.

The proof is based on the following two lemmas and two propositions.

Lemma 4.2.2. *Let G and o be as in Theorem 4.2.1 and let G' be a subgraph of G which is invariant under a subgroup Γ of $\mathrm{AUT}(G,\ell)$ acting quasi-transitively on G'. Suppose $L_{\mathrm{SAW},o}(G)$ is regular, respectively context-free. Then there is a vertex o' of G' such that $L_{\mathrm{SAW},o'}(G')$ is also regular, respectively context-free.*

Proof. G' is also a deterministically edge-labelled graph. For any $o' \in V(G')$, the language $L(\mathcal{P}')$ induced by the set $\mathcal{P}'$ of *all* walks of length at least one in G' starting at o', is regular. To see this, we construct a regular grammar $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ over the label alphabet $\mathbf{\Sigma}$ of G such that $L(\mathbf{G}) = L(\mathcal{P}')$. Denote by $\Gamma V(G')$ and $\Gamma E(G')$ the finite set of orbits of the action of Γ on $V(G')$ and $E(G')$, respectively. Every edge orbit Γe naturally connects the two not necessarily different vertex orbits Γe^- and Γe^+. The set of non-terminals

$$\mathbf{N} = \{S\} \cup \{V_{\Gamma v} \mid \Gamma v \in \Gamma V(G)\}$$

together with the production rules

$$\mathbf{P} = \{S \vdash V_{\Gamma o'}\} \cup \{V_{\Gamma e^-} \vdash \ell(e) V_{\Gamma e^+} \mid \Gamma e \in \Gamma E(G')\} \cup \{V_{\Gamma e^-} \vdash \ell(e) \mid \Gamma e \in \Gamma E(G')\}$$

generates the desired language $L(\mathcal{P}')$.

Now there is some $o' \in V(G')$ with $d(o, V(G')) = d(o, o')$, and there is a walk p_0 in G of that length from o to o'. Let $w_0 = \ell(p_0)$, and let $p_0\mathcal{P}'$ be the set of all concatenated walks $p_0 p'$, where $p' \in \mathcal{P}'$. Thus, $L(p_0\mathcal{P}') = w_0 L(\mathcal{P}') = \{w_0 w \mid w \in L(\mathcal{P}')\}$ is again regular.

If $L_{\mathrm{SAW},o}(G)$ is regular (respectively context-free), then by [32] also $L_{\mathrm{SAW},o}(G) \cap L(p_0\mathcal{P}')$ is regular (respectively context-free). Since o' is the only vertex of p_0 contained in $V(G')$,

$$L_{\mathrm{SAW},o}(G) \cap L(p_0\mathcal{P}') = w_0 L_{\mathrm{SAW},o'}(G').$$

If we delete from the latter language the common prefix w_0, we also get a regular (respectively context-free) language. □

Lemma 4.2.3. *Let G be a simple, connected, infinite, locally finite, deterministically edge-labelled graph and let $\Gamma \leq \mathrm{AUT}(G,\ell)$ act quasi-transitively on G. Assume that $L_{\mathrm{SAW},o}(G)$ is context-free for some $o \in V(G)$. Then Γ contains a non-elliptic element.*

Proof. Let l_p be the pumping length of $L_{\mathrm{SAW},o}(G)$ given by Lemma 2.2.2 and $w \in L_{\mathrm{SAW},o}(G)$ with $|w| \geq l_p$. Then w can be written as $w = xyz\tilde{y}\tilde{x}$ for some $x, \tilde{x}, y, \tilde{y}, z \in \mathbf{\Sigma}^*$ such that $|yz\tilde{y}| \leq l_p$, $|y\tilde{y}| \geq 1$ and $xy^n z\tilde{y}^n\tilde{x} \in L_{\mathrm{SAW},o}(G)$ for all $n \geq 0$. Now either $|y| > 0$, or $|y| = 0$ and $|\tilde{y}| > 0$. Set $a = x$, $b = y$ in the first case and $a = xz$, $b = \tilde{y}$ in the second case so that $a, b \in \mathbf{\Sigma}^*$ and $|b| > 0$.

Let v_0 be the end-vertex of the walk starting at o and labelled by the word a. Then, for every $n \geq 0$, let p_n be the unique self-avoiding walk of length $n\,|b|$ starting at v_0 and having label b^n. Thus, p_{n+1} is a self-avoiding extension of p_n, and in the limit we obtain

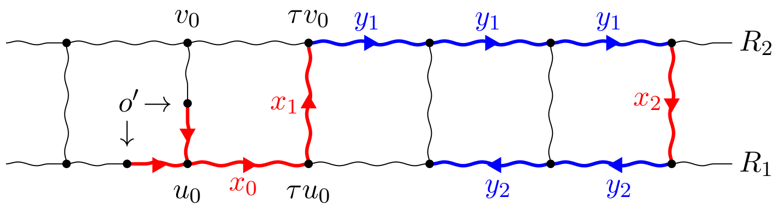


Figure 4.1: Labelled subdivision of the ladder graph.

a ray R with vertex set $\{v_0, v_1, \dots\}$. Since Γ acts quasi-transitively on G there must be some $\tau \in \Gamma$ and some $j > i \geq 0$ such that $\tau v_{i|b|} = v_{j|b|}$. Without loss of generality (up to truncation of an initial piece of R) assume $i = 0$. Then $\tau^n v_0 = v_{nj|b|} \neq v_0$ for every $n \geq 1$ and Proposition 12 in [26] yields that τ is non-elliptic. □

Remark 4.2.4. In group theoretical terms the previous lemma says that if the finitely generated group $\Gamma \leq \mathrm{AUT}(G, \ell)$ acts quasi-transitively on G and is an infinite torsion group, that is an infinite group where all elements have finite order, then $L_{\mathrm{SAW},o}(G)$ is *not* context-free.

Proposition 4.2.5. *Let G be a simple, connected, infinite, quasi-transitive deterministically edge-labelled graph. If G contains a τ-strip S of size 2, where $\tau \in \Gamma$, then $L_{\mathrm{SAW},o}(G)$ is not regular.*

Proof. As a τ-strip, S contains two τ-invariant double rays R_1 and R_2 given by their sequences of vertices $(u_n)_{n\in\mathbb{Z}}$ and $(v_n)_{n\in\mathbb{Z}}$. The graph H mentioned in the definition of τ-strips is connected and therefore contains a path P connecting R_1 and R_2. Without loss of generality, we assume that P connects the vertices u_0 and v_0 and does not contain further vertices of R_1 or R_2. The vertex sets of the induced subgraphs $\tau^n H$ of S are pairwise disjoint, so the paths $\tau^n P$ are also pairwise disjoint. As a consequence, the subgraph G' of G spanned by R_1, R_2 and all $\tau^n P$ ($n \in \mathbb{Z}$) is a τ-invariant subdivision of the bi-infinite ladder, see Figure 4.1.

We suppose that $L_{\mathrm{SAW},o}(G)$ is regular for $o \in V(G)$ and will reach a contradiction. Lemma 4.2.2 applied to G' yields an $o' \in V(G')$ such that $L_{\mathrm{SAW},o'}(G')$ is regular. Without loss of generality, we assume that either o' lies on R_1 between u_0 and $\tau^{-1}u_0$ and is distinct from the latter, or that o' lies on P and is distinct from v_0. (Otherwise we can exchange the two rays.) In Figure 4.1, we indicate the possible positions of o'.

Let x_0 be the label of the walk from o' to τv_0 via u_0 and τu_0. Write x_1 for the common label of the walks from $\tau^n u_0$ to $\tau^n v_0$ on $\tau^n P$ ($n \in \mathbb{Z}$) and x_2 for the label of the reversed walks. Next, let y_1 denote the common label of each of the walks from any $\tau^n v_0$ to $\tau^{n+1} v_0$ within R_2. And finally, let y_2 be the common label of the walks from any $\tau^{n+1} u_0$ to $\tau^n u_0$ within R_1. Each of the words $x_0, x_1, x_2, y_1, y_2 \in \mathbf{\Sigma}^*$ is non-empty, but they are in general not just elements of $\mathbf{\Sigma}$.

The language consisting of all words

$$w = x_0 x_1 y_1^k x_2 y_2^l, \quad k, l \in \mathbb{N}, \tag{4.2}$$

is regular, so that by the closure properties of regular languages, also its intersection with $L_{\mathrm{SAW},o'}(G')$, which we denote by $\bar{L}$, is regular. Now, $\bar{L}$ consists of all words of the form (4.2) corresponding to self-avoiding walks starting at o'. Looking at Figure 4.1, such a walk goes from o' to τu_0, then upwards to τv_0 and to the right to $\tau^{k+1} v_0$ along R_2, then downwards to $\tau^{k+1} u_0$, and finally to the left to $\tau^{k+1-l} u_0$. Thus, in order to be self-avoiding, one must have $k > l$.

Let l_p be the pumping length of Lemma 2.2.1 for $\bar{L}$, and let $w = x_0 x_1 y_1^{l_p+1} x_2 y_2^{l_p} \in \bar{L}$. In the decomposition $w = xy\tilde{x}$ of the lemma, $|y\tilde{x}| \leq l_p$ implies that $y\tilde{x}$ is a postfix of $y_2^{l_p}$. That is, $y_2^{l_p} = \tilde{y} y \tilde{x}$ for some word $\tilde{y} \in \mathbf{\Sigma}^*$. Now also $w' = xy^2\tilde{x}$ must be in $\bar{L}$, so there must be k, l such that

$$x_0 x_1 y_1^k x_2 y_2^l = w' = x_0 x_1 y_1^{l_p+1} x_2 \tilde{y} y^2 \tilde{x}.$$

Since the labelling is deterministic and the first symbol of y_1 and the first symbol of x_2 are both labels of different edges starting at v_0, these symbols must be different. We conclude that $k = l_p + 1$ and $y_2^l = \tilde{y} y^2 \tilde{x}$. This is longer than $y_2^{l_p}$, so $l \geq l_p + 1 = k$. But then the walk with label w starting at o' is not self-avoiding, a contradiction. □

Proposition 4.2.6. *Let G be as in Proposition 4.2.5. If G contains a τ-strip S of size 3, where $\tau \in \Gamma$, then $L_{\mathrm{SAW},o}(G)$ is not context-free.*

Proof. We suppose that $L_{\mathrm{SAW},o}(G)$ is context-free and will again arrive at a contradiction.

As a τ-strip, S contains three τ-invariant double rays R_1, R_2 and R_3. For $i \in \{1, 2, 3\}$ let $(v_n^{(i)})_{n \in \mathbb{Z}}$ be the sequence of vertices of the ray R_i. As in Proposition 4.2.5, find a path P_1 in the connected induced subgraph H of S connecting two of these rays. Without loss of generality assume that P_1 is a path from $v_0^{(1)}$ to $v_0^{(2)}$ not containing further vertices of R_1, R_2 or R_3. In a similar way find a path P_2 in the connected induced subgraph τH of S connecting R_3 and one of the other two rays. We may assume that P_2 is a path from $v_r^{(2)}$ to $v_r^{(3)}$ ($r \geq 1$) not containing further vertices of the three rays. The vertex sets of the graphs $\tau^n H$ ($n \in \mathbb{Z}$) are pairwise disjoint, so the paths in $\{\tau^{2n} P_1 \mid n \in \mathbb{Z}\} \cup \{\tau^{2m} P_2 \mid m \in \mathbb{Z}\}$ are pairwise disjoint. The subgraph G' of G spanned by the three rays and all images $\tau^{2n} P_1$ and $\tau^{2n} P_2$ ($n \in \mathbb{Z}$) is a τ^2-periodic subdivision of the bi-infinite 3-ladder, see Figure 4.2. For convenience we replace τ with τ^2, so that G' is τ-periodic.

Again, Lemma 4.2.2 applies to G', and there is $o' \in V(G')$ such that $L_{\mathrm{SAW},o'}(G')$ is context-free.

Up to possibly renumbering the rays, inverting their direction or exchanging R_1 with R_3, we can assume without loss of generality that o' lies on the "rectangle" with corners $v_0^{(1)}, \tau^{-1} v_0^{(1)}, \tau^{-1} v_0^{(2)}$ and $v_0^{(2)}$, but not on the path P_1. In Figure 4.2, we indicate the possible positions of o'.

Let $x_1 \in \mathbf{\Sigma}^*$ be the label of the self-avoiding walk constructed as follows. Start at o', run around that rectangle in clockwise order up to $v_0^{(1)}$, then move along P_1 to $v_0^{(2)}$, follow R_2 until $v_r^{(2)}$ and finally reach $v_r^{(3)}$ via P_2. Furthermore let x_2 be the label of the self-avoiding walk starting in $v_r^{(3)}$, following P_2 to $v_r^{(2)}$, then moving along R_2 until

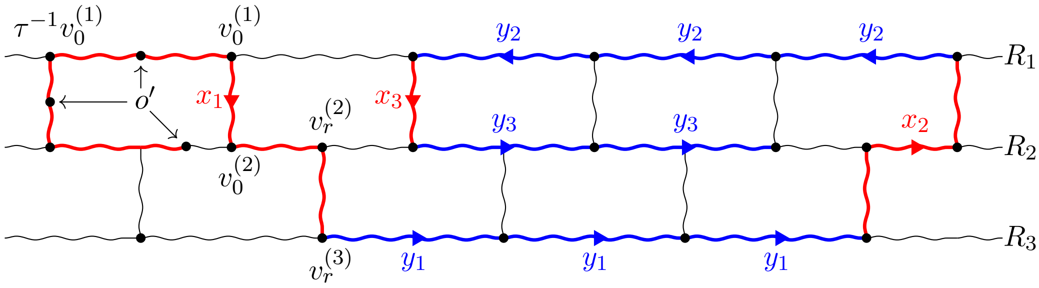


Figure 4.2: Labelled subdivision of the 3-ladder graph.

reaching $\tau v_0^{(2)}$ and finally reaching $\tau v_0^{(2)}$ via τP_1. Thereafter, let x_3 be the label of the walk from $v_0^{(1)}$ to $v_0^{(2)}$ on P_1.

Let y_1 denote the label of the walk from $v_r^{(3)}$ to $\tau v_r^{(3)}$ on R_3, let y_2 be the label of the walk from $\tau v_0^{(1)}$ to $v_0^{(1)}$ on R_1 and let y_3 be the label of the walk from $v_0^{(2)}$ to $\tau v_0^{(2)}$ on R_2. The automorphism τ is label preserving, any translates of the previous walks are also labelled with the same words. Each of the words x_i, y_i is non-empty, but again, they are in general not just elements of $\mathbf{\Sigma}$.

Similarly to the previous proposition, the language L_1 consisting of all words

$$w = x_1 y_1^k x_2 y_2^l x_3 y_3^m, \quad k, l, m \in \mathbb{N}, \tag{4.3}$$

is regular. Recall that the class of context-free languages is closed under homomorphisms, inverse homomorphisms and intersection with regular languages. Hence the language $L_2 = L_1 \cap L_{\mathrm{SAW},o'}(G')$ must be context-free. Following the arrows in Figure 4.2, one can see a self-avoiding walk with such a label, with $k = l = m+1 = 3$. In general, for a walk with label w as in (4.3) to be self-avoiding, one must have

$$k \geq l > m. \tag{4.4}$$

We define a language homomorphism

$$\phi : \{a_1, b_1, a_2, b_2, a_3, b_3\}^* \to \mathbf{\Sigma}^*$$

by setting $\phi(a_i) = x_i$ and $\phi(b_i) = y_i$ for $i \in [3]$. Note that a_i and b_i are single letters while x_i and y_i are words over the alphabet $\mathbf{\Sigma}$ and may consist of multiple letters. Then the language

$$L_3 = \{a_1 b_1^k a_2 b_2^l a_3 b_3^m \mid k \geq l > m \geq 1\} \tag{4.5}$$

is context-free because

$$L_3 = \phi^{-1}(L_2) \cap \{a_1 b_1^k a_2 b_2^l a_3 b_3^m \mid k, l, m \in \mathbb{N}\}.$$

Note that this statement strongly relies on the fact that the edge-labelling ℓ is deterministic: the image $\phi(w)$ of any word $w \in \{a_1 b_1^k a_2 b_2^l a_3 b_3^m \mid k, l, m \in \mathbb{N}\}$ is the label of a unique walk in $\mathcal{P}$ and thus has a unique representation of the form (4.3), which lies in L_2 if and only if $k \geq l > m \geq 1$.

However, using the pumping lemma for context-free languages it is an easy exercise to show that L_3 cannot be context-free, leading to a contradiction. Let l_p be the pumping length from Lemma 2.2.2 and let $w = a_1 b_1^{l_p+1} a_2 b_2^{l_p+1} a_3 b_3^{l_p} \in L_3$ and $w = xyz\tilde{y}\tilde{x}$ be a decomposition of w such that $|yz\tilde{y}| \leq l_p$ and $|y\tilde{y}| \geq 1$. If $y\tilde{y}$ contains some a_i, then clearly $xy^2z\tilde{y}^2\tilde{x}$ is not in L_3. Otherwise $y\tilde{y}$ cannot contain both of b_1 and b_3. If it does not contain b_3, then $xz\tilde{x}$ ist not in L_3, otherwise $xy^2z\tilde{y}^2\tilde{x}$ is not in L_3. This can be seen by comparing the number of occurrences of the letters b_1, b_2 and b_3 with the conditions on k, l, m specified in (4.5). We conclude that w is not pumpable, thus L_3 cannot be context-free. □

For the proof of Theorem 4.2.1 we will need Bass-Serre theory. As the topic cannot be briefly introduced, we do not give all definitions here. The reader is referred to Serre [53] and Dicks and Dunwoody [10].

The *ends* of a finitely generated group are the ends of any of its Cayley graphs with respect to a finite, symmetric set of generators. In fact, they do not depend on the choice of the generating set; see Remark 2.1.3.

In terms of Bass-Serre theory, a group Γ is accessible if it is the fundamental group of a finite graph of groups having finite edge groups and vertex groups which are finite or have exactly one end. The following lemma is Corollary IV.1.9 in [10].

Lemma 4.2.7. *A group Γ is the fundamental group of a finite graph of finite groups if and only if Γ is finitely generated and virtually free.*

Recall that a locally finite graph G is accessible if there is an integer k such that any two ends of G can be separated by a set containing k or fewer vertices. Thomassen and Woess [56] showed that a simple, connected, locally finite, transitive graph G is accessible if and only if there is an integer M such that each thin end of G has size at most M. Moreover, they also proved that a finitely generated group is accessible if and only if some (and therefore all) of its Cayley-graphs are accessible.

Proof of Theorem 4.2.1. First, G cannot be one-ended. Indeed, in that case, that end has to be thick. If $\Gamma = \mathrm{AUT}(G, \ell)$ has only elliptic elements, then $L_{\mathrm{SAW},o}(G)$ is not context-free by Lemma 4.2.3. Otherwise, Γ has parabolic elements, and Lemma 4.1.9 combined with Proposition 4.2.6 implies as well that $L_{\mathrm{SAW},o}(G)$ is not context-free.

Thus, G has more than one end, whence there are thin ends. If $L_{\mathrm{SAW},o}(G)$ is context-free then by Lemma 4.1.8 and Proposition 4.2.6 all thin ends have size at most 2. We need to prove that there are no thick ends.

Recall from Lemma 2.1.2 and its proof that in the graph G^{2D+1}, the orbit Γo induces a Cayley graph $\mathrm{Cay}(\Gamma, S)$ of the finitely generated group Γ. The identity mapping $\iota : \Gamma o \hookrightarrow V(G)$ induces a quasi-isometric embedding of $\mathrm{Cay}(\Gamma, S)$ into G. Indeed, it is bi-Lipschitz and quasi-surjective, that is, $B = 0$ in (4.1). Consequently, by Corollary 4.1.4, the size of all thin ends of $\mathrm{Cay}(\Gamma, S)$ is bounded from above by $\eta(2)$. By [56], the group Γ is accessible. Thus, it is the fundamental group of a finite graph of finitely generated (sub)groups which are finite or one-ended. If all of them are finite, then by Lemma 4.2.7, Γ is virtually free, so that the group Γ has only thin ends since any of its Cayley graphs is quasi-isometric with a tree, see for example [60].

Thus, if Γ has a thick end, then it must have a finitely generated subgroup Γ_1 having one (thick) end. Above, we have identified Γ with the vertex set Γo in G, and $\Gamma_1 o$ is contained in that orbit. Under this identification, the group unit corresponds to the "root" vertex o. Let S_1 be a finite, symmetric set of generators of Γ_1. Then for each $s \in S$ there is a (shortest) path P_s in G connecting o to the image so. Choose these paths such that $P_{s^{-1}} = s^{-1}P_s$ and consider the subgraph G_1 of G with

$$V(G_1) = \bigcup_{\gamma\in\Gamma_1, s\in S} \gamma(V(P_s)) \quad \text{and} \quad E(G_1) = \bigcup_{\gamma\in\Gamma_1, s\in S} \gamma(E(P_s)).$$

Clearly, G_1 is a connected subgraph of G and inherits the labels from the edges of G. Also, G_1 is quasi-isometric with the Cayley graph $\mathrm{Cay}(\Gamma_1, S_1)$. Indeed, the embedding $\Gamma_1 \hookrightarrow V(G_1)$, $\gamma \mapsto \gamma o$ is bi-Lipschitz and quasi-surjective, that is, $B = 0$ in (4.1).

Therefore, G_1 has one end, which has to be thick, and it is quasi-transitive under Γ_1. But then we are back to the situation of the beginning of this proof, that is, $L_{\mathrm{SAW},o}(G_1)$ cannot be context-free. But this contradicts Lemma 4.2.2.

We conclude that Γ and thus also G has no thick ends. □

4.3 Graphs with context-free language of self-avoiding walks

The goal of this section is to prove the second half of Theorem 4.0.1:

Theorem 4.3.1. *Let G be a simple, connected, locally finite, deterministically edge-labelled graph on which $\mathrm{AUT}(G, \ell)$ acts quasi-transitively. Then for every vertex o of G the following holds:*

(i) If all ends of G have size 1, then $L_{\mathrm{SAW},o}(G)$ is regular.

(ii) If all ends of G have size at most 2, then $L_{\mathrm{SAW},o}(G)$ is unambiguous context-free.

For an integer $k > 0$ a graph G is called *k-connected* if it has more than k vertices and no set of less than k vertices is a separating set in G. A *(2-)block* in a graph G is a maximal connected subgraph of G containing no cut-vertex. If G is connected and has at least 2 vertices, every block of G is either a pair of vertices connected by an edge or a 2-connected graph. The intersection of 2 different blocks of G is either empty or a cut-vertex in G. The *block-cut-vertex tree* $T_2(G)$ corresponding to G is the graph having as vertices the blocks and the cut-vertices of G, where a block is adjacent to every cut-vertex it contains. Denote for an edge e of $T_2(G)$ by $B(e)$ the block and by $c(e)$ the cut-vertex incident to e. More about blocks and the block-cut-vertex tree can be found in [58].

If G is locally finite, every cut-vertex of G belongs to a finite number of blocks and therefore has finite degree in $T_2(G)$. On the other hand, an infinite block of G can contain infinitely many cut-vertices, so $T_2(G)$ need not be locally finite.

For any automorphism $\gamma \in \text{AUT}(G)$, the image γB of a block B of G is again a block of G and the same holds for cut-vertices. Whenever a cut-vertex c is contained in a block B, its image $\gamma(c)$ is contained in $\gamma(B)$, so $\text{AUT}(G)$ acts on the graph $T_2(G)$ by automorphisms. Additionally, if $\Gamma \leq \text{AUT}(G)$ acts quasi-transitively on G, it also acts quasi-transitively on $T_2(G)$.

The following lemma shows that blocks of quasi-transitive graphs are always quasi-transitive graphs.

Lemma 4.3.2. *Let G be a simple, connected, locally finite graph and suppose $\Gamma \leq \text{AUT}(G)$ acts quasi-transitively on G. Then for any block B of G, the set-wise stabiliser Γ_B of B in Γ acts quasi-transitively on B.*

Proof. Γ acts quasi-transitively on G, so it acts with finitely many orbits on $E(B)$, in other words, the set $\{\Gamma e \mid e \in E(B)\}$ is finite. But every $\gamma \in \Gamma$ mapping some edge of B onto another edge of B clearly fixes the block B and is therefore also contained in Γ_B. This implies that also Γ_B acts with finitely many orbits on $E(B)$ and thus quasi-transitively on B. □

The following lemma is a simple consequence of the fact that a block cannot contain ends of size 1. This is true because any defining sequence of such an end consists of cut-vertices.

Lemma 4.3.3. *Let G be a locally finite graph such that all ends of G are of size 1. Then every block of G is finite.*

In the case where the graph G has infinite blocks, we want to further decompose them. There are different ways to do this. One natural way is to use *3-block decompositions*, first introduced by Tutte (see [58]) for finite graphs and then generalised to infinite graphs by Droms, Servatius and Servatius in [12]. In this theory, sometimes graphs may have multiple edges between a single pair of vertices.

Let G and H be two not necessarily simple graphs and let $e \in E(G)$ and $f \in E(H)$ be (directed) edges. The *edge amalgam* of G and H along the edges e and f is the graph obtained from the disjoint union of G and H by identifying the vertices e^- with f^-, e^+ with f^+ and erasing the edges e and f and their inverse edges $\bar{e}$ and $\bar{f}$. A convenient way to represent a sequence of edge amalgamations of a (not necessarily finite) set of graphs is the *edge-amalgam tree* T. Vertices of T are the graphs used in the amalgamation. For clarity we denote by $\mathcal{G}(s)$ the graph corresponding to the vertex s of T. Two vertices s and t are connected by a pair of directed edges $(a, \bar{a})$ in T if $\mathcal{G}(s)$ and $\mathcal{G}(t)$ are amalgamated along some edges $e \in E(\mathcal{G}(s))$ and $f \in E(\mathcal{G}(t))$. We additionally introduce a label function λ assigning to every directed edge $a \in E(T)$ the edge e of $\mathcal{G}(a^-)$ used in the amalgamation of $\mathcal{G}(a^-)$ and $\mathcal{G}(a^+)$. For $s \in V(T)$ an edge e of $\mathcal{G}(s)$ is called *virtual* if e or $\bar{e}$ is the label of some edge of T, otherwise it is called *non-virtual.* Denote the resulting graph obtained from a sequence of edge amalgamations given by an amalgamation tree T by $\mathcal{G}(T)$. Then virtual edges disappear during the progress, while non-virtual edges are still present in $\mathcal{G}(T)$.

A *multilink* is a graph consisting of 2 vertices and a (finite) positive number of undirected edges between these vertices. A graph is said to be a *3-block* if it contains at least 3 pairs of directed edges and is either a cycle (closed path), a multilink or a locally finite 3-connected graph. An edge-amalgam tree T is called a *3-block tree* if for every $s \in V(T)$, the graph $\mathcal{G}(s)$ is a 3-block and additionally for every edge $a \in E(T)$ the corresponding graphs $\mathcal{G}(a^-)$, $\mathcal{G}(a^+)$ are neither both multilinks nor both cycles.

Theorem 4.3.4 ([12, Theorem 1]). *For any locally finite, 2-connected graph G there is a unique 3-block tree T such that $G = \mathcal{G}(T)$.*

For a given locally finite 2-connected graph G the unique 3-block tree given by the above theorem will henceforth be denoted by $T_3(G)$. The proof of the theorem is constructive and allows us to "decompose" G in a unique way into (possibly infinitely many) 3-blocks, such that G is obtained from amalgamating these 3-blocks as given by $T_3(G)$. The set of vertices and the set of non-virtual edges of each 3-block will be considered as a subsets of $V(G)$ and $E(G)$, respectively. A single vertex of G may appear in more than one block.

We state without proof the following two properties of $T_3(G)$:

(T1) For every virtual edge $e = \lambda(a)$, $a \in E(T_3(G))$, there is a finite sub-tree T' of $T_3(G)$ containing a^- but not a^+ and a path in $\mathcal{G}(T')$ connecting the endpoints of e and consisting of edges of G.

(T2) Let $s \in V(T_3(G))$ and $v \in V(\mathcal{G}(s))$. Then s is contained in a finite sub-tree T' of $T_3(G)$ such that all edges of $\mathcal{G}(T')$ incident to v are edges of G.

Due to the uniqueness of the decomposition, symmetries on the graph G carry over to T in a canonical way by mapping 3-blocks onto 3-blocks. Moreover, as in the case of 2-blocks, any $\Gamma \leq \mathrm{AUT}(G)$ acting quasi-transitively on G also acts quasi-transitively on $T_3(G)$.

Lemma 4.3.5. *Let G be a simple, locally finite, 2-connected graph such that all ends of G are of size at most 2. Then every 3-block of G is finite.*

Proof. Clearly, multilinks and cycles are finite. Every end of a 3-connected graph must be of size at least 3 because every defining sequence of an end consists of separating sets.

Assume that there is an infinite 3-block B of G. Then it contains an end ω of size at least 3 and this end contains 3 disjoint rays. By property (T1) of $T_3(G)$, we can replace all virtual edges contained in the rays by finite paths consisting of non-virtual edges of G which are not in B. We obtain 3 disjoint rays in G, which belong to the same end of G. This is a contradiction to the assumption that all ends of G have size at most 2. □

As an important consequence of this lemma, decomposition trees $T_3(G)$ of graphs G as in Lemma 4.3.5 are locally finite because any 3-block contains only finitely many virtual edges.

The proof of Theorem 4.3.1 will be done in several steps. First we decompose the given graph G into its 2-blocks and introduce a regular language L encoding the behaviour of walks on the block-cut-vertex tree $T_2(G)$. In the second step we further decompose each 2-block B into its (finite) 3-blocks and use the 3-block tree $T_3(B)$ to construct a context-free grammar for the language of self-avoiding walks on B. This second step will be discussed first and is comprised in the upcoming Theorem 4.3.6. Finally we use language substitution to obtain the language of self-avoiding walks on G by replacing the variables of L by the languages on the corresponding 2-blocks. Examples 4.4.2 and 4.4.3 may be beneficial for a better understanding of the upcoming proofs.

Theorem 4.3.6. *Let G be a 2-connected, locally finite, deterministically edge-labelled graph and let $\Gamma \leq \mathrm{AUT}(G, \ell)$ act quasi-transitively on G. If every end of G is of size at most 2, then for every $o \in V(G)$, the language $L_{\mathrm{SAW},o}(G)$ is unambiguous context-free.*

Proof. Let $T_3 := T_3(G)$ be the 3-block tree of G and let $R \subseteq E(T_3)$ be a set of representatives of the finite set of orbits $\{\Gamma e \mid e \in E(T_3)\}$ of directed edges of T_3. Recall that any $a \in E(T_3)$ is labelled with some virtual edge $\lambda(a)$ of the part $\mathcal{G}(a^-)$. Denote by ρa the representative of a in R and for a vertex $u \in \lambda(a)$ by ρu a representative of u contained in the edge $\lambda(\rho a)$. Define for $a \in R$ and $u \in \lambda(a)$ the set $\mathcal{P}^0_{a,u}$ of self-avoiding walks in $\mathcal{G}(a^+)$ starting at u and not containing the virtual edge $\lambda(\bar{a})$ or its inverse edge. Let $\mathcal{P}^1_{a,u}$ be the subset of $\mathcal{P}^0_{a,u}$ consisting of all walks not containing the second vertex of $\lambda(a)$. Note that both sets are finite because the 3-block $\mathcal{G}(a^+)$ is finite by Lemma 4.3.5. Fix some root vertex r of T_3 such that the 3-block $\mathcal{G}(r)$ contains the vertex o of $V(G)$. We denote by $\mathcal{P}_o$ the set of self-avoiding walks in $\mathcal{G}(r)$ starting at o.

We extend the label function ℓ on G to 3-blocks of G. Labels of non-virtual edges are inherited from G and for any $a \in E(T_3)$ label the virtual directed edge $e = \lambda(a)$ by $U_{\rho a,\rho u}$ and its inverse edge $\bar{e}$ by $U_{\rho a,\rho v}$, where $u = e^-$ and $v = e^+$ are initial and terminal vertex of e. The extended label function is again denoted by ℓ and maps into $\mathbf{\Sigma} \cup \mathbf{\Sigma}'$, where $\mathbf{\Sigma}$ is the label alphabet of G and $\mathbf{\Sigma}' = \{U_{a,u} \mid a \in R,\ u \in \lambda(a)\}$.

We present a grammar $\mathcal{C} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ generating the language of self-avoiding walks in G starting at o. The finite set of variables is given by

$$\mathbf{N} = \{S\} \cup \mathbf{\Sigma}' \cup \{I^i_{a,u} \mid a \in R, u \in \lambda(a), i \in \{0,1\}\}.$$

Let $a \in R$ and $u \in \lambda(a)$. Then for any directed edge $b \neq \bar{a}$ in $E(T_3)$ starting at a^+, the productions given below are contained in $\mathbf{P}$.

$I^0_{a,u} \vdash \ell(p)$	for every $p \in \mathcal{P}^0_{a,u}$ ending with a non-virtual edge,
$I^0_{a,u} \vdash \ell(p) I^0_{\rho b,\rho v}$	for every $p \in \mathcal{P}^0_{a,u}$ ending at a vertex $v \in \lambda(b)$ and not containing the second vertex of $\lambda(b)$,
$I^0_{a,u} \vdash \ell(p) I^1_{\rho b,\rho v}$	for every $p \in \mathcal{P}^0_{a,u}$ ending at a vertex $v \in \lambda(b)$ and containing the second vertex of $\lambda(b)$, but not $\lambda(b)$,
$I^1_{a,u} \vdash \ell(p)$	for every $p \in \mathcal{P}^1_{a,u}$ ending with a non-virtual edge,
$I^1_{a,u} \vdash \ell(p) I^0_{\rho b,\rho v}$	for every $p \in \mathcal{P}^1_{a,u}$ ending at a vertex $v \in \lambda(b)$ if the second vertex of $\lambda(b)$ is neither contained in p nor in $\lambda(a)$,

$I^1_{a,u} \vdash \ell(p) I^1_{\rho b, \rho v}$ for every $p \in \mathcal{P}^1_{a,u}$ ending at a vertex $v \in \lambda(b)$ if the second vertex of $\lambda(b)$ is either contained in p or in $\lambda(a)$,

$U_{a,u} \vdash \ell(p)$ for every $p \in \mathcal{P}^0_{a,u}$ ending at the second vertex of $\lambda(a)$.

The set of productions $\mathbf{P}$ is completed by adding for every edge $b \in E(T_3)$ starting at the root r the following rules.

$S \vdash \ell(p)$ for every $p \in \mathcal{P}_o$ ending with a non-virtual edge,

$S \vdash \ell(p) I^0_{\rho b, \rho v}$ for every $p \in \mathcal{P}_o$ ending at a vertex $v \in \lambda(b)$ and not containing the second vertex of $\lambda(b)$,

$S \vdash \ell(p) I^1_{\rho b, \rho v}$ for every $p \in \mathcal{P}_o$ ending at a vertex $v \in \lambda(b)$ and containing the second vertex of $\lambda(b)$ but not $\lambda(b)$.

The set $\mathbf{P}$ is finite because the tree $E(T_3)$ is locally finite and all sets $\mathcal{P}^i_{a,u}$ and $\mathcal{P}_o$ are finite. Now we briefly discuss why the given grammar $\mathcal{C}$ unambiguously generates the language of self-avoiding walks in G starting at o.

Let $a = st$ be an edge of T_3 and T' be the component of $T_3 - \{s\}$ containing t. A self-avoiding walk p of length at least 1 in G is called an *I-walk with direction a* if it starts at a vertex $u \in \lambda(a)$ and all edges of the walk are contained in $\mathcal{G}(T')$. A U-walk with direction a is an I-walk with direction a ending at a vertex of $\lambda(a)$.

Then the following statements hold.

(a) For $a \in E(T)$ and $u \in \lambda(a)$, the variable $U_{\rho a, \rho u}$ unambiguously generates the language of all U-walks with direction a starting at u.

(b) For $a \in E(T)$ and $u \in \lambda(a)$, the variable $I^0_{\rho a, \rho u}$ unambiguously generates the language of all I-walks with direction a starting at u, and the variable $I^1_{\rho a, \rho u}$ unambiguously generates the language of all I-walks with direction a starting at u and not containing the second vertex of $\lambda(a)$.

(c) The start symbol S unambiguously generates the language $L_{\mathrm{SAW},o}(G)$.

Rigorous proofs for these statements are long and technical, so we only sketch them here.

Let s be a vertex of T and $\mathcal{G}(s)$ be its corresponding 3-block. Define the projection $p(s)$ of a self-avoiding walk p onto $\mathcal{G}(s)$ in the following way: Let $v_1, \ldots, v_k$ be the sequence of vertices of p which are contained in $\mathcal{G}(s)$ ordered by their occurrence in p and p_i be the sub-walk of p connecting v_i and v_{i+1}. For every i there are 2 cases: If p_i is a single edge of $\mathcal{G}(s)$, add this edge to $p(s)$. Otherwise there is an edge a of T such that p_i is a U-walk with direction a and we add the virtual edge $\lambda(a)$ (or its inverse), which connects v_i and v_{i+1}. This edge can be seen as a shortcut for the U-walk p_i. For reasons of ambiguity, if p_{k-1} is a U-walk and p ends at v_k, we do not add the corresponding virtual edge and consider our projection to end at v_{k-1}. The resulting $p(s)$ is a self-avoiding walk in $\mathcal{G}(s)$.

For every U-walk p with direction a starting at u we obtain the word $\ell(p(a^+))$ corresponding to the projection of p onto the 3-block $\mathcal{G}(a^+)$ from the variable $U_{a,u}$ in a single derivation step. A simple induction shows that p is generated by $U_{a,u}$. Moreover for any word w generated by $U_{a,u}$, the walk p starting at u and having label w is a U-walk with direction a starting at u. The word w can only be obtained from a unique sequence of rightmost derivations because in every step we have to generate the string corresponding to the projection of p onto some 3-block. Statement (a) follows.

For any I-walk p with direction a starting at u we derive $\ell(p(a^+))$ in a single step of derivation from $I^0_{\rho a,\rho u}$ and if p does not contain the second vertex of $\lambda(a)$, also from $I^1_{\rho a,\rho u}$. Using (a) and induction, this implies that $\ell(p)$ is generated by the corresponding variables. Furthermore note that $I^0_{\rho a,\rho u}$ only appears in the right hand side of productions if the second vertex of $\lambda(a)$ is not contained in the projection of p on any block previously visited by p. Using this observation it is straight forward to show that walks corresponding to words generated by $I^0_{\rho a,\rho u}$ and $I^1_{\rho a,\rho u}$ are indeed I-walks with direction a starting at u and that every such walk is generated unambiguously. This finishes the proof of (b).

Finally, for every SAW p starting at o we derive $\ell(p(r))$ in a single step from S and (c) follows from (a) and (b) as before. □

Proof of Theorem 4.3.1. The group $\Gamma = \mathrm{AUT}(G, \ell)$ acts quasi-transitively on the block-cut-vertex tree $T_2 := T_2(G)$. Edges of T_2 are treated as being undirected, we do not distinguish between e and $\bar{e}$. We use a similar approach as in 4.2.2 to create a regular grammar $\mathbf{G}$ encoding the movement of a walk on the tree T_2. Denote by $\Gamma V(T_2)$ the set of vertex orbits and by $\Gamma E(T_2)$ the set of undirected edge orbits of the action of Γ on T_2. Every element of $\Gamma V(T_2)$ corresponds to either a class of blocks or a class of cut-vertices of G, so for $\tilde{e} \in \Gamma E(T_2)$ we write again $B(\tilde{e})$ (block) and $c(\tilde{e})$ (cut-vertex) for the vertex orbits of the endpoints of any edge e representing $\tilde{e}$. Note that for any cut-vertex c of G, all edges of T_2 incident to c lie in different orbits with respect to Γ because by Lemma 2.1.2, Γ acts fixed-point-freely on G. For $\tilde{e} \in \Gamma E(T_2)$ let $N(\tilde{e}) = \{\tilde{f} \in \Gamma E(T_2) \setminus \{\tilde{e}\} \mid c(\tilde{e}) = c(\tilde{f})\}$. Fix some block B_o containing the vertex o.

Let L be the regular language generated by the grammar $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where

$$\mathbf{N} = \{S\} \cup \{I_{\tilde{e}} \mid \tilde{e} \in \Gamma E(T_2)\},$$

$$\mathbf{\Sigma} = \{x_o\} \cup \{x_{o,\tilde{f}} \mid \tilde{f} \in \Gamma E(T_2)\} \cup \{x_{\tilde{e}} \mid \tilde{e} \in \Gamma E(T_2)\} \cup \{x_{\tilde{e},\tilde{f}} \mid \tilde{e}, \tilde{f} \in \Gamma E(T_2)\},$$

and the set of productions $\mathbf{P}$ consists of

$$\begin{aligned}
&S \vdash x_o,\\
&S \vdash x_{o,\tilde{f}} I_{\tilde{g}} && \text{for } \tilde{f} \in \Gamma E(T_2),\ \tilde{g} \in N(\tilde{f}),\\
&I_{\tilde{e}} \vdash x_{\tilde{e}} && \text{for } \tilde{e} \in \Gamma E(T_2),\\
&I_{\tilde{e}} \vdash x_{\tilde{e},\tilde{f}} I_{\tilde{g}} && \text{for } \tilde{e}, \tilde{f} \in \Gamma E(T_2),\ \tilde{g} \in N(\tilde{f}).
\end{aligned}$$

For $\tilde{e} \in \Gamma E(T_2)$ fix some representative $e \in E(T_2)$ and let $L(\tilde{e})$ be the language of self-avoiding walks of length at least 1 in the block $B(e)$ starting at the vertex $c(e)$. Clearly,

$L(\tilde{e})$ does not depend on the choice of e. By Lemma 4.3.2 the stabiliser $\Gamma_{B(e)}$ acts quasi-transitively on the graph $B(e)$ and by assumption on G all ends of $B(e)$ have size at most 2. Hence Theorem 4.3.6 applies and $L(\tilde{e})$ is context-free. Denote for $\tilde{f} \in \Gamma E(T_2)$ by $L(\tilde{e}, \tilde{f})$ the subset of $L(\tilde{e})$ corresponding to walks ending at vertices of $B(e)$, which lie in the orbit $c(\tilde{f})$. Note that $L(\tilde{e}, \tilde{f})$ may be an empty language if $B(\tilde{e}) \neq B(\tilde{f})$ or if $\tilde{e} = \tilde{f}$ and $c(e)$ is the only representative of $c(\tilde{e})$ in $B(e)$. As the intersection of the unambiguous context-free language $L(\tilde{e})$ and the regular language of all walks starting at c and ending at a representative of $c(\tilde{f})$, $L(\tilde{e}, \tilde{f})$ is unambiguous context-free. In a similar way let $L(o)$ be the language of all walks in B_o starting at o and $L(o, \tilde{f})$ be the subset of $L(o)$ corresponding to walks ending at a representative of $c(f)$.

Let φ be the substitution of languages given for $\tilde{e}, \tilde{f} \in \Gamma E(T_2)$ by

$$\varphi(x_o) = L(o), \quad \varphi(x_{o,\tilde{f}}) = L(o, \tilde{f}), \quad \varphi(x_{\tilde{e}}) = L(\tilde{e}), \quad \varphi(x_{\tilde{e},\tilde{f}}) = L(\tilde{e}, \tilde{f}).$$

Then by Theorem 3.4.1 in [32] the result $\varphi(L)$ of the substitution is context-free.

If every end of G has size at most 1, by Lemma 4.3.3 every block of G is finite and thus also the language of self-avoiding walks in the block is finite. We conclude that in this case $\varphi(L)$ is regular.

For $e \in E(T_2)$, a self-avoiding walk p of length at least 1 in G is called an *I-walk with direction* e if it starts at $c(e)$ and its first edge lies in the block $B(e)$. Denote by $\tilde{e}$ the orbit of e under Γ. Then the following statements hold:

(a) The variable $I_{\tilde{e}}$ generates a regular language $L_{\tilde{e}}$ such that $\varphi(L_{\tilde{e}})$ is the language of I-walks with direction e.

(b) $\varphi(L)$ is the unambiguous context-free language $L_{\mathrm{SAW},o}(G)$ of self-avoiding walks in G starting at o.

As before we only sketch the proofs for these statements.

Let p be an I-walk with direction $e \in E(T_2)$. Let p_1 be the part of p contained in the block $B(e)$. If $p_1 = p$, then $\ell(p_1)$ is contained in $L(\tilde{e})$ and therefore obtained in a single step of derivation. Otherwise there is some $f \in E(T_2)$ such that p leaves $B(e)$ via $c(f)$ and $\ell(p_1)$ is contained in $L(\tilde{e}, \tilde{f})$. In this case p enters one of the other blocks containing $c(f)$, which are blocks $B(g)$, $g \in E(T_2)$, such that $\tilde{g} \in N(\tilde{f})$. The rest of p is an I-walk with direction g. A simple induction shows that $\ell(p)$ is contained in $\varphi(L_{\tilde{e}})$.

On the other hand, every word $w \in \varphi(L_e)$ corresponds to a unique walk p starting at $c(e)$ labelled by w. This walk p is self-avoiding, because the parts of p contained in blocks are self-avoiding, and whenever leaving a block $B(f)$, $f \in E(T_2)$, p can never enter $B(f)$ again because T_2 is a tree and $N(\tilde{f})$ does not contain $\tilde{f}$. This implies that p is an I-walk with direction e and proves (a).

In the same way it can be seen that for any $p \in \mathcal{P}_{\mathrm{SAW},o}$, the label of the part p_0 contained in B_o is contained in $L(o)$ if $p_0 = p$, and in $L(o, f)$ if p leaves B_o via a vertex in the class $c(f)$. Therefore $\ell(p)$ is contained in $\varphi(L)$. In the same way as in (a) we obtain that every walk starting in o and corresponding to a word in $\varphi(L)$ is self-avoiding.

Let $w \in \varphi(L)$ and p be the SAW starting at o and having label w. There is a unique word $w' \in L$ such that $w \in \varphi(w')$. This word w' is given by the sequence of blocks visited by p. Furthermore, for every $a \in \mathbf{\Sigma}$ the language $\varphi(a)$ is unambiguous context-free, hence $\varphi(L)$ is also unambiguous context-free. This shows statement (b) and finishes the proof. $\square$

4.4 Discussion and examples

The proof of Theorem 4.3.1 is constructive and the obtained grammar can be used to calculate the generating function of self-avoiding walks $F_{\mathrm{SAW},o}(z)$ and the connective constant of graphs satisfying the conditions of the theorem.

Given some language L, the ordinary generating function $F_L(z)$ is the power series

$$F_L(z) := \sum_{w \in L} z^{|w|}.$$

Using the algebraic theory of context-free languages Chomsky and Schützenberger developed in [7], the productions of an unambiguous context-free grammar $\mathcal{C}$ generating the language L can be translated into a system of polynomial equations having as one of its solutions the language generating function $F_L(z)$.

A power series $F(z)$ is called *algebraic* over a field K if it satisfies a polynomial equation of the form $P\big(z, F(z)\big) = 0$, where $P(x, y)$ is a bivariate polynomial in $K[x, y]$. From classical elimination theory (see for example [39]) it follows that any component of a solution of a system of polynomial equations having coefficients in K is algebraic over K, in particular $F_L(z)$ is algebraic over $\mathbb{Q}$.

Let G be a connected, locally finite, deterministically edge-labelled graph and o be a vertex of G such that the language $L_{\mathrm{SAW},o}(G)$ is unambiguous context-free. Then the label function ℓ acts as a bijection between the set $\mathcal{P}_{\mathrm{SAW},o}$ of self-avoiding walks starting at o and its language $L_{\mathrm{SAW},o}(G)$, whence the SAW-generating function satisfies $F_{\mathrm{SAW},o}(z) = F_{L_{\mathrm{SAW},o}(G)}(z)$. All singularities of algebraic functions are algebraic numbers, so in particular the radius of convergence of $F_{\mathrm{SAW},o}(z)$ and thus also the connective constant of the graph G are algebraic numbers.

Example 2.1.4 shows that there are transitive graphs G not admitting any deterministic labelling ℓ such that $\mathrm{AUT}(G, \ell)$ acts quasi-transitively on G. Nevertheless, the following statement can be shown using the previous discussions and the ideas and proofs of Section 4.3, but generating functions counting walks, instead of grammars and language theory.

Corollary 4.4.1. *Whenever all ends of a connected, locally finite, quasi-transitive graph G are of size at most 2, the SAW-generating function $F_{\mathrm{SAW},o}(z)$ is algebraic over $\mathbb{Q}$. In particular the connective constant $\mu(G)$ is an algebraic number.*

Alm and Janson showed in [1] that the generating functions of self-avoiding walks on one-dimensional lattices are algebraic over $\mathbb{Q}$, independently of the size of the ends. In Chapter 6, we shall examine how our results can be extended to quasi-transitive graphs having only thin ends. In future work we plan to investigate under which structural conditions on the graph, the language of self-avoiding walks is accepted by a multipass automaton as in [4].

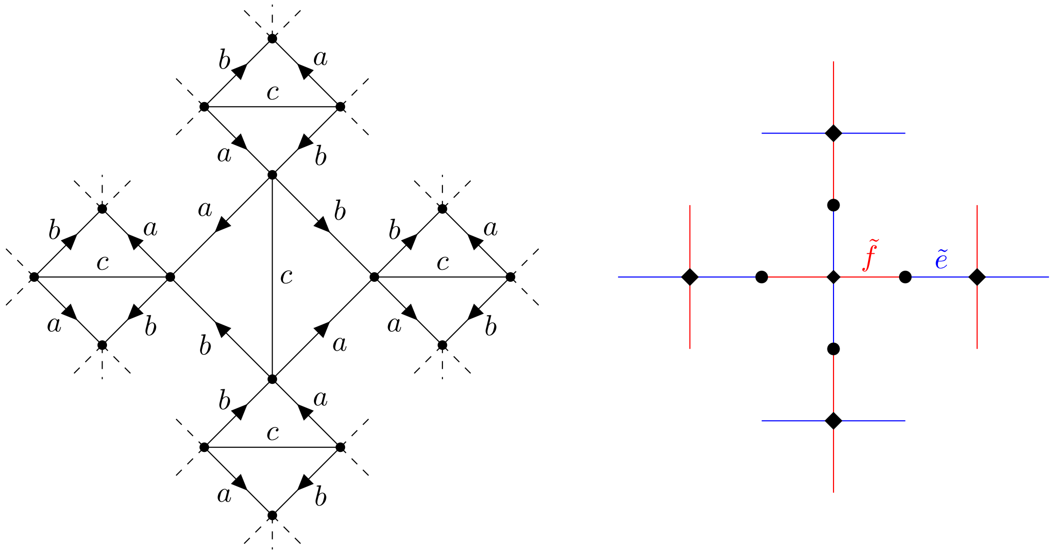


Figure 4.3: The labelled graph X (left) and its block-cut-vertex tree (right). Block-vertices are drawn as diamonds, the edge orbits $\tilde{e}$ and $\tilde{f}$ have different colours.

Some concrete examples, where we used Theorem 4.3.1 and its constructive proof to obtain the SAW-generating functions and the connective constants, include the "sandwich" of two k-regular trees, already treated in a slightly different way in [42] and the above mentioned grandparent graph. We conclude the chapter with a detailed discussion of two additional examples. The Cayley graph in the upcoming example has only ends of size one, so it demonstrates the construction in the proof of Theorem 4.3.1. While the graph is transitive and has finite blocks, these blocks are not transitive graphs.

Example 4.4.2. Consider the group $\Gamma = \langle a, c \mid c^2 = 1 \rangle$, which is the free product of an infinite cyclic group and a group with two elements. Let $b = ca$ and let G be the Cayley graph $\mathrm{Cay}(\Gamma, S)$, where S is the symmetric generating set $S = \{a, a^{-1}, b, b^{-1}, c\}$. Edges are labelled by the generators in the usual way. The resulting graph can be seen in Figure 4.3. It is clear that all ends of G have size 1, hence the language of self-avoiding walks on G is regular.

The blocks of G consist of 4 vertices and 5 edges. The block-cut-vertex tree T_2 of G is a biregular tree, where every block has degree 4 and every cut-vertex has degree 2. Obviously the blocks are not transitive graphs, therefore the action of $\mathrm{AUT}(G)$ on $E(T_2)$ admits more than one orbit. The two orbits can be seen in Figure 4.3; we will denote them by $\tilde{e}$ and $\tilde{f}$.

Every block is finite, so a grammar can be simply constructed as in the proof of Theorem 4.3.1 without further decomposition of the blocks into 3-blocks. The variables $L_{\tilde{e}}$ and $L_{\tilde{e},\tilde{f}}$ generate the languages $L(\tilde{e})$ and $L(\tilde{e},\tilde{f})$ defined in the proof. The set of productions of a grammar G generating $L_{\mathrm{SAW},o}(G)$ (for an arbitrary choice of o) looks as follows:

$$
\begin{aligned}
S &\vdash \epsilon \mid I_{\tilde{e}} \mid I_{\tilde{f}}, \\
I_{\tilde{e}} &\vdash L_{\tilde{e}} \mid L_{\tilde{e},\tilde{e}} I_{\tilde{f}} \mid L_{\tilde{e},\tilde{f}} I_{\tilde{e}}, \\
I_{\tilde{f}} &\vdash L_{\tilde{f}} \mid L_{\tilde{f},\tilde{e}} I_{\tilde{f}} \mid L_{\tilde{f},\tilde{f}} I_{\tilde{e}}, \\
L_{\tilde{e}} &\vdash a \mid ab^{-1} \mid ab^{-1}a \mid b \mid ba^{-1} \mid ba^{-1}b \mid c \mid ca \mid cb, \\
L_{\tilde{f}} &\vdash a^{-1} \mid a^{-1}b \mid a^{-1}ba^{-1} \mid a^{-1}c \mid a^{-1}ca \mid b^{-1} \mid b^{-1}a \mid b^{-1}ab^{-1} \mid b^{-1}c \mid b^{-1}cb, \\
L_{\tilde{e},\tilde{e}} &\vdash ab^{-1} \mid ba^{-1} \mid c, \\
L_{\tilde{e},\tilde{f}} &\vdash a \mid ab^{-1}a \mid b \mid ba^{-1}b \mid ca \mid cb, \\
L_{\tilde{f},\tilde{e}} &\vdash a^{-1} \mid a^{-1}ba^{-1} \mid a^{-1}c \mid b^{-1} \mid b^{-1}ab^{-1} \mid b^{-1}c, \\
L_{\tilde{f},\tilde{f}} &\vdash a^{-1}b \mid b^{-1}a.
\end{aligned}
$$

Note that although this grammar is not regular, it is easily possible to get a regular grammar (with only 3 variables) by substituting all L-variables in the first 3 lines by their produced languages given by lines 4 to 9. Translating the productions into a system of equations and solving it yields

$$F_{\mathrm{SAW},o}(z) = -\frac{(2z-1)\left(2z^2+z+1\right)\left(2z^3+2z^2+2z+1\right)}{4z^6+8z^5+8z^4+2z^3-4z+1}.$$

The connective constant of G is the reciprocal of the smallest positive root of the denominator of this generating function,

$$\mu(G) \approx 3.6279766.$$

Finally we provide an example where we end up with an SAW-generating function, which is algebraic over $\mathbb{Q}$, but not rational.

Example 4.4.3. Consider the group $\Gamma = \langle a, b, c \mid a^2 = b^2 = c^2 = (ab)^3 = (bc)^2 = 1 \rangle$, which is a free product with amalgamation of dihedral groups of order 4 and 6. Let G be the Cayley graph $\mathrm{Cay}\,(\Gamma, \{a, b, c\})$. As before label the edges by the corresponding generators. The resulting graph can be seen in Figure 4.4. All ends of G have size 2, so the language of self-avoiding walks in G is context-free. The 3-block decomposition of G can be seen in Figure 4.4. It yields 3 types of 3-blocks, denoted by A, B and C in correspondence to the labels they contain. $E(T_3(G))$ contains 4 types of edges, they will be denoted by AB, BA, BC, CB, depending on the pair of 3-blocks they connect (e.g. AB starts at A and ends at B).

A grammar generating the languages of self-avoiding walks constructed as in the proof

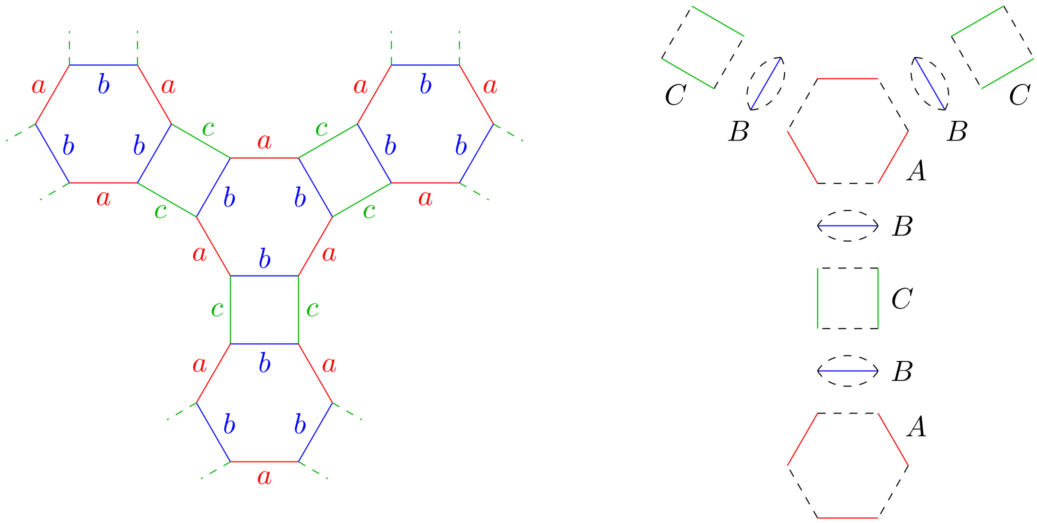


Figure 4.4: The labelled graph X (left) and its 3-block decomposition (right). Any non-virtual edge has the same colour as its generator; virtual edges are dashed.

of Lemma 4.3.6 is given by the following productions:

$$
\begin{aligned}
S &\vdash \epsilon \mid I^0_{BA} \mid I^0_{BC} \mid b \mid bI^1_{BA} \mid bI^1_{BC} \mid U_{BA}I^1_{BC} \mid U_{BC}I^1_{BA},\\
I^0_{BA} &\vdash a \mid aI^0_{AB} \mid aU_{AB}a \mid aU_{AB}aI^0_{AB} \mid aU_{AB}aU_{AB}a,\\
I^1_{BA} &\vdash a \mid aI^0_{AB} \mid aU_{AB}a \mid aU_{AB}aI^0_{AB},\\
I^0_{AB} &\vdash I^0_{BC} \mid b \mid bI^1_{BC}, & I^1_{AB} &\vdash I^1_{BC},\\
I^0_{BC} &\vdash c \mid cU_{CB}c \mid cI^0_{CB}, & I^1_{BC} &\vdash c \mid cI^0_{CB},\\
I^0_{CB} &\vdash I^0_{BA} \mid b \mid bI^1_{BA}, & I^1_{CB} &\vdash I^1_{BA},\\
U_{AB} &\vdash b \mid U_{BC}, & U_{BA} &\vdash aU_{AB}aU_{AB}a,\\
U_{CB} &\vdash b \mid U_{BA}, & U_{BC} &\vdash cU_{CB}c.
\end{aligned}
$$

Translating this set of productions into the corresponding system of equations and solving this system yields the SAW-generating function

$$F_{\mathrm{SAW},o}(z) = \frac{P(z) + Q(z)\sqrt{-4z^8 - 4z^6 + 1}}{z^{12}\left(2z^{10} + 8z^9 + 13z^8 + 12z^7 + 7z^6 + 4z^5 + 5z^4 + 4z^3 + z^2 - 2z - 1\right)},$$

where $P(z)$ and $Q(z)$ are two polynomials of degree 23 and 17, respectively. As before, the connective constant of G can be found as the reciprocal of the smallest positive real root of the polynomial of degree 10 in the denominator,

$$\mu(G) \approx 1.8306977.$$

5 Comparing consecutive letter counts in multiple context-free languages

Formal language theory makes use of mathematical tools to study the syntactical aspects of natural and artificial languages. Two of the best known and most studied classes of formal languages are context free and context sensitive languages, generated by context free grammars and context sensitive grammars, respectively. Context-free grammars have convenient generative properties, but they are not able to model cross-serial dependencies, occurring in Swiss German and a few other natural languages. The expressive power of context-sensitive grammars on the other hand often exceeds our requirements, and the decision problem whether a given string belongs to the language generated by such a grammar is PSPACE-complete.

To overcome these issues, intermediate classes of “mildly context sensitive languages” were independently introduced by Vijay-Shanker et al. [59] and Seki et al. [52] in the form of context-free rewriting systems and multiple context-free grammars (MCFGs). These concepts turn out to be equivalent in the sense that they both lead to the same class of languages, called multiple context-free languages (MCFLs). While MCFGs are able to model cross-serial dependencies by dealing with tuples of strings, the languages generated by them share several important properties with context free languages, such as polynomial time parsability and semi-linearity.

MCFLs can be distinguished depending on the largest dimension m of tuples involved. The m-MCFLs obtained in this way form an infinite strictly increasing hierarchy

$$\text{CFL} = 1\text{-MCFL} \subsetneq 2\text{-MCFL} \subsetneq \ldots \subsetneq m\text{-MCFL} \subsetneq (m+1)\text{-MCFL} \subsetneq \ldots \subsetneq \text{CSL},$$

where CFL and CSL denote the classes of context free and context sensitive languages, respectively.

A highlight in the theory of MCFGs is a result by Salvati [51], stating that the language $O_2 = \{w \in \{a, \bar{a}, b, \bar{b}\}^* \mid |w|_a = |w|_{\bar{a}} \wedge |w|_b = |w|_{\bar{b}}\}$ occurring as the word problem of the group $\mathbb{Z}^2$ is a 2-MCFL. Ho [33] generalised this result by showing that for any positive integer d the word problem of $\mathbb{Z}^d$ is multiple context-free.

In this chapter we study languages defined by comparing lengths of runs of consecutive identical letters and show that they are able to separate the layers of the hierarchy mentioned above. In particular we consider languages of the form

$$L_k = \{a_1^{n_1} a_2^{n_2} \cdots a_k^{n_k} \mid n_1 \geq n_2 \geq \cdots \geq n_k \geq 0\}$$

and generalisations thereof. The languages L_1 and L_2 are easily seen to be context-free, and it is a standard exercise to show that L_3 is not context-free by using the pumping lemma for context free languages. Our main result generalises these observations.

Theorem 5.0.1. *The language* $L_k = \{a_1^{n_1} a_2^{n_2} \cdots a_k^{n_k} \mid n_1 \geq n_2 \geq \cdots \geq n_k \geq 0\}$ *is a* $\lceil k/2 \rceil$*-MCFL but not a* $(\lceil k/2 \rceil - 1)$*-MCFL.*

The first part of Theorem 5.0.1 is verified by constructing an appropriate grammar. For the second part, one might hope that it is implied by a suitable generalisation of the pumping lemma to m-MCFLs, but unfortunately such a generalisation does not exist.

In [52] Seki et al. generalise the concept of pumpability to m-pumpability. However, their weak pumping lemma for m-MCFLs only confirms the existence of m-pumpable strings in infinite m-MCFLs and not that all but finitely many words in the language are m-pumpable. In particular, it is not strong enough to imply the second part of Theorem 5.0.1. While Kanazawa [37] managed to prove a strong version of the pumping lemma for the sub-class of well-nested m-MCFLs, Kanazawa et al. [38] showed that in fact such a pumping lemma cannot exist for general m-MCFLs by giving a 3-MCFL containing infinitely many words which are not k-pumpable for any given k. Nevertheless, our proof relies heavily on the idea of pumping, thus showing that this technique can be useful even in cases where no pumping lemma is available.

At first glance the material of this chapter seems completely unrelated to self-avoiding walks. However, the main motivation was our wish to generalise Theorem 4.2.1. One of the main ingredients for this theorem was the application of the pumping lemma for context-free languages in the proof of Proposition 4.2.6. In a similar way, Theorem 5.0.1 will be applied in the proof of Lemma 6.4.8, which is basically a generalisation of Proposition 4.2.6.

5.1 Main result

In this chapter we focus on languages defined as follows: A binary relation $\preceq$ on a set M is called a *preorder*, if it is reflexive and transitive. In contrast to partial orders, preorders need not be antisymmetric, that is, it is possible that $a \preceq b$ and $b \preceq a$ for different elements a, b. A preorder $\preceq$ is called *total* if for all $a, b \in M$ we have $a \preceq b$ or $b \preceq a$. The *comparability graph* of a preorder is the simple undirected graph with vertex set M, where two different vertices u and v are connected by an edge if they are comparable. We call a preorder *connected*, if its comparability graph is connected. Note that any total preorder is connected, but a connected preorder does not have to be total.

For a positive integer m and a preorder $\preceq$ on $[m] = \{1, 2, \ldots, m\}$ define the language $L_\preceq$ over the alphabet $\Sigma = \{a_1, \ldots, a_m\}$ by

$$L_\preceq = \{a_1^{n_1} a_2^{n_2} \cdots a_m^{n_m} \mid i \preceq j \Rightarrow n_i \leq n_j\}.$$

A preorder $\preceq'$ on M is said to be a *totalisation* of a preorder $\preceq$ on M, if it is total and extends $\preceq$, that is, whenever $a \preceq b$ also $a \preceq' b$. Let $T_\preceq$ be the set of totalisations of $\preceq$.

Remark 5.1.1. Observe that

$$L_\preceq = \bigcup_{\preceq' \in T_\preceq} L_{\preceq'}.$$

This is a consequence of the fact that for any given word $w = a_1^{n_1} a_2^{n_2} \cdots a_m^{n_m} \in L_{\preceq}$, the binary relation $\preceq'$ on $[m]$ defined by $i \preceq' j$ if and only if $n_i \leq n_j$ is a totalisation of $\preceq$.

A natural way of specifying a language is by giving a grammar generating it. Here we focus on multiple context-free languages and the grammars generating them. Important definitions and basic facts about them are collected in Section 2.2.

By the following lemma it is enough to consider MCFGs in a certain normal form.

Lemma 5.1.2 (Seki et al. [52, Lemma 2.2])**.** *Every m-MCFL L is generated by an m-MCFG $\mathcal{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ satisfying the following conditions.*

(i) If $A(\alpha_1, \ldots, \alpha_r) \leftarrow A_1(x_{1,1}, \ldots, x_{1,r_1}), \ldots, A_n(x_{n,1}, \ldots, x_{n,r_n})$ is a non-terminating rule in $\mathbf{P}$, then the string $\alpha_1 \cdots \alpha_r$ contains each $x_{i,j}$ exactly once and does not contain elements of $\mathbf{\Sigma}$.

(ii) If $A(w_1, \ldots, w_r) \leftarrow$ is a terminating rule, then the string $w_1 \cdots w_r$ has length at most 1. The only rule where this string is allowed to have length 0 is the rule $S(\epsilon) \leftarrow$, which is present in $\mathbf{P}$ if and only if $\epsilon \in L$.

We split the proof of our main result into two parts, covered by Theorem 5.1.3 and Theorem 5.1.4, respectively. Together, these two results clearly imply Theorem 5.0.1; it is also worth pointing out that in fact they cover the (much larger) class of languages $L_{\preceq}$ as introduced in the previous section.

Theorem 5.1.3. *For every preorder $\preceq$ the language $L_{\preceq} = \{a_1^{n_1} a_2^{n_2} \cdots a_m^{n_m} \mid i \preceq j \Rightarrow n_i \leq n_j\}$ over the alphabet $\mathbf{\Sigma} = \{a_1, \ldots, a_m\}$ is a $\lceil m/2 \rceil$-MCFL.*

Proof. It is known (see for instance [52]) that the class of k-MCFLs is closed under substitution and taking finite unions. Thus it is enough to consider the case where $m = 2k$ is even, the case $m = 2k - 1$ follows by substituting ϵ for a_{2k}. Additionally, by Remark 5.1.1 we may assume that $\preceq$ is a total preorder.

We show that $L_{\preceq}$ is generated by the k-MCFG $\mathcal{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{N} = \{S, A\}$, the non-terminal A has rank k and $\mathbf{P}$ consists of the rules

$$\begin{aligned} S(x_1 x_2 \cdots x_k) &\leftarrow A(x_1, x_2, \ldots, x_k) \\ A(\epsilon, \epsilon, \ldots, \epsilon) &\leftarrow \end{aligned}$$

and for every $j \in [2k]$ the additional rule ρ_j given by

$$A(y_1 x_1 y_2, y_3 x_2 y_4, \ldots, y_{2k-1} x_k y_{2k}) \leftarrow A(x_1, x_2, \ldots, x_k),$$

where

$$y_i = \begin{cases} a_i & \text{if } j \preceq i, \\ \epsilon & \text{otherwise.} \end{cases}$$

First note that if $\vdash A(w_1, \ldots, w_k)$, then each w_l has the form $w_l = a_{2l-1}^{n_{2l-1}} a_{2l}^{n_{2l}}$, and it holds that $n_i \leq n_j$ whenever $i \preceq j$. This is clearly true for $A(\epsilon, \epsilon, \ldots, \epsilon)$ and it is preserved when applying the rule ρ_j, which adds one instance of the letter a_j and every letter a_i with $j \preceq i$. Every word w generated by $\mathcal{G}$ is in $L_{\preceq}$ since it is the concatenation $w_1 \cdots w_k$ of strings w_l such that $\vdash A(w_1, \ldots, w_k)$.

Next we show that any given word in $L_{\preceq}$ is generated by $\mathcal{G}$. Assume for a contradiction that there is a word in $L_{\preceq}$ which is not generated by $\mathcal{G}$. Pick such a word $w = a_1^{n_1} a_2^{n_2} \cdots a_{2k}^{n_{2k}}$ with minimal $n_{\max} := \max\{n_l \mid l \in [2k]\}$. As $\mathcal{G}$ generates the empty word, $w \neq \epsilon$ and $n_{\max} \geq 1$. For $l \in [2k]$ let $n'_l = n_l$ if $n_l < n_{\max}$, and let $n'_l = n_{\max} - 1$ otherwise. Since $w \in L_{\preceq}$ we have $n'_i \leq n'_j$ whenever $i \preceq j$, and thus $w' = a_1^{n'_1} a_2^{n'_2} \cdots a_{2k}^{n'_{2k}} \in L_{\preceq}$. By minimality of w the word w' is generated by $\mathcal{G}$, and in particular $\vdash A(a_1^{n'_1} a_2^{n'_2}, \ldots, a_{2k-1}^{n'_{2k-1}} a_{2k}^{n'_{2k}})$. Pick some minimal j with respect to $\preceq$ from the set $\{l \in [2k] \mid n_l = n_{\max}\}$. Applying the rule ρ_j to $A(a_1^{n'_1} a_2^{n'_2}, \ldots, a_{2k-1}^{n'_{2k-1}} a_{2k}^{n'_{2k}})$ yields $\vdash A(a_1^{n_1} a_2^{n_2}, \ldots, a_{2k-1}^{n_{2k-1}} a_{2k}^{n_{2k}})$; consequently $\mathcal{G}$ generates w, contradicting our assumption. □

Theorem 5.1.4. *For every connected preorder $\preceq$ the language $L_{\preceq} = \{a_1^{n_1} a_2^{n_2} \cdots a_m^{n_m} \mid i \preceq j \Rightarrow n_i \leq n_j\}$ over the alphabet $\mathbf{\Sigma} = \{a_1, \ldots, a_m\}$ is not a $(\lceil m/2 \rceil - 1)$-MCFL.*

Proof. Assume that there is a MCFG $\mathcal{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ generating $L_{\preceq}$, and assume that $\mathcal{G}$ is given in normal form as in Lemma 5.1.2.

For a derivation tree D and $i \in [m]$ denote by $|D|_i$ the total number of letters a_i occurring in all substrings contained in the term $\ell(D)$ (the label of the root of D) and by $|D| = \sum_{i=1}^{m} |D|_i$ the combined length of all substrings. Since $\mathcal{G}$ is in normal form, if $\ell(D)$ is not a terminating rule and $D_1, \ldots, D_k$ are the derivation trees rooted at the k children of the root of D, we have

$$|D|_i = \sum_{j=1}^{k} |D_j|_i \,. \tag{5.1}$$

Moreover, if $\ell(D)$ is a terminating rule, then

$$|D| = 1. \tag{5.2}$$

Call a rule a *combiner*, if its right side contains at least 2 non-terminals. Note that a vertex of any derivation tree labelled by a combiner has at least 2 children. Furthermore there is an upper bound K such that the right side of any combiner contains at most K non-terminals.

Fix $n > K^{2C}$, where C is the number of combiners in $\mathbf{P}$, and let D be a derivation tree of $S(a_1^n a_2^n \cdots a_m^n)$. Then D contains a path starting at the root containing at least $2C + 1$ vertices labelled with combiners. If not, then (5.1) and (5.2) imply $|D| \leq K^{2C}$, contradicting our choice of n. By the pigeon hole principle there is some combiner ρ such that this path contains at least 3 vertices labelled ρ. Denote the subtrees rooted at these three vertices by D_1, D_2, and D_3 such that $D_3 \subseteq D_2 \subseteq D_1$.

We claim that for any $i \preceq j$ we have $|D_1|_j - |D_2|_j = |D_1|_i - |D_2|_i$, and that an analogous statement holds for D_2 and D_3.

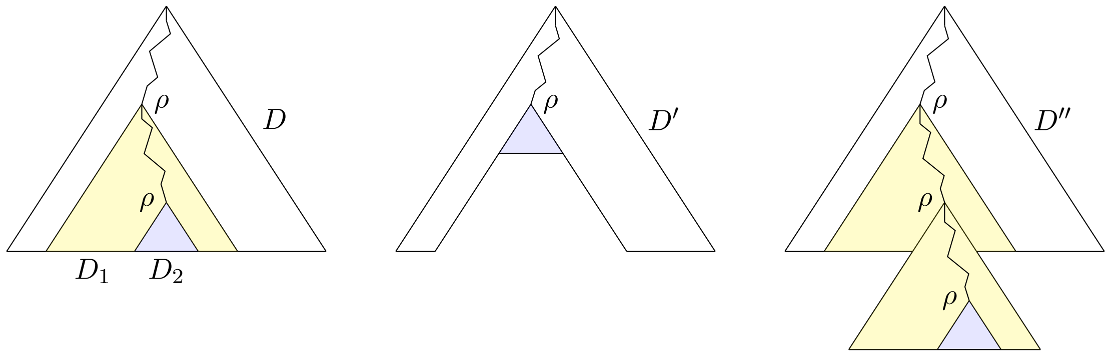


Figure 5.1: Replacing D_1 with D_2 yields D' and replacing D_2 with D_1 yields D''.

Assume that $|D_1|_j - |D_2|_j > |D_1|_i - |D_2|_i$. By (5.1) the derivation tree D' obtained by replacing D_1 with D_2 (compare Remark 2.2.7, also see Figure 5.1) satisfies

$$|D'|_j - |D'|_i = |D|_j - (|D_1|_j - |D_2|_j) - |D|_i + (|D_1|_i - |D_2|_i) < 0,$$

because $|D|_j = |D|_i = n$. This is a contradiction, as the word $w(D')$ is not in $L_{\preceq}$. If $|D_1|_j - |D_2|_j < |D_1|_i - |D_2|_i$, then the derivation tree D'' obtained by replacing D_2 by D_1 satisfies

$$|D''|_j - |D''|_i = |D|_j + (|D_1|_j - |D_2|_j) - |D|_i - (|D_1|_i - |D_2|_i) < 0,$$

a contradiction for the same reason as before. This completes the proof of our claim.

If $i, j \in [m]$ are comparable in $\preceq$, then $|D_1|_j - |D_1|_i = |D_2|_j - |D_2|_i$. By connectedness of the comparability graph this is true for any pair i, j.

Since ρ is a combiner, $|w(D_1)| > |w(D_2)|$. In particular $|D_1|_i > |D_2|_i$ for some and thus for every $i \in [m]$. Analogously we obtain $|D_2|_i > |D_3|_i$; in particular $|D_2|_i > 0$ holds for every $i \in [m]$.

Assume now for a contradiction the Grammar $\mathcal{G}$ is $(\lceil m/2 \rceil - 1)$-MCF. Then $w(D_2)$ consists of at most $\lceil m/2 \rceil - 1$ strings and each of them is a substring of $a_1^n a_2^n \cdots a_m^n$ because $\mathcal{G}$ is in normal form. Every letter of $\mathbf{\Sigma}$ appears in $w(D_2)$, hence one of the strings must contain at least 3 different letters and thus be of the form $a_{i-1}^{n_1} a_i^n a_{i+1}^{n_2}$ for some $i \in \{2, \ldots, m-1\}$. As this contradicts the fact that $n \geq |D_1|_i > |D_2|_i = n$, the grammar $\mathcal{G}$ cannot be $(\lceil m/2 \rceil - 1)$ multiple context-free. □

6 Multiple context-free language of self-avoiding walks

As mentioned in the introduction of this thesis, explicit computation of $\mu(G)$ can be a very challenging task, even in seemingly harmless instances such as two-dimensional lattices.

The special case of SAWs on one-dimensional lattices turns out to be a lot more manageable. The reason for this is that the large scale structure of one-dimensional lattices resembles a line, that is, they can be decomposed into infinitely many pairwise isomorphic finite parts such that each part only intersects with two others (its predecessor and its successor). By considering restrictions of SAWs to single parts and analysing how these restrictions fit together, Alm and Janson [1] showed that SAW-generating functions on these lattices are always rational. While this approach fails for higher-dimensional lattices, analogous techniques have been successfully applied to other graph classes, in particular to graphs exhibiting some kind of large scale tree structure, see for instance the note [19] by Gilch and Müller.

In this chapter, we follow a similar approach to study SAWs on graphs with only thin ends, thus extending the results in Chapter 4. Our first main result concerns the SAW-generating function.

Theorem 6.0.1. *Let G be a simple, locally finite, connected, quasi-transitive graph having only thin ends and let $o \in V(G)$. Then $F_{\mathrm{SAW},o}(z)$ is algebraic over $\mathbb{Q}$. In particular the connective constant $\mu(G)$ is an algebraic number.*

Our second main result connects SAWs to formal languages. As in Chapter 4, let G be deterministically edge-labelled, that is, every (directed) edge e of G is assigned a label $\ell(e)$ from some given alphabet $\mathbf{\Sigma}$ such that different edges with the same initial vertex have different labels. As before, edge-labelled graphs are assumed to be quasi-transitive, that is, the group $\mathrm{AUT}(G, \ell)$ of automorphisms of G preserving ℓ acts with finitely many orbits on G.

Recall that the edge-labelling is extended to walks $p = (v_0, e_1, v_1, \ldots, e_n, v_n)$ by setting

$$\ell(p) = \ell(e_1) \ldots \ell(e_n).$$

In this way, any set $\mathcal{P}$ of walks gives rise to a language $L(\mathcal{P}) = \{\ell(p) \mid p \in \mathcal{P}\}$, thus allowing us to study properties of $\mathcal{P}$ via properties of the corresponding language.

This is particularly fruitful when $L(\mathcal{P})$ belongs to a well understood family of languages. Besides the well-known classes of regular and context-free languages, the class of multiple context free languages (MCFLs) plays an important role. These were introduced by Seki et al. [52] as a generalisation of context free languages capable of modelling cross-serial dependencies occurring in some natural languages such as Swiss German. A concise definition of MCFLs can be found in Section 2.2.

We are interested in the *language of self-avoiding walks* defined by

$$L_{\mathrm{SAW},o}(G) = L(\mathcal{P}_{\mathrm{SAW},o}) = \{\ell(p) \mid p \in \mathcal{P}_{\mathrm{SAW},o}\},$$

where $\mathcal{P}_{\mathrm{SAW},o}$ is the set of all SAWs of length at least 1 on G starting at o. In his computation of $F_{\mathrm{SAW},o}(z)$ for the infinite ladder graph, Zeilberger [63] implicitly used that the language of self-avoiding walks of this graph is context free. More generally, in Chapter 4 we showed that $L_{\mathrm{SAW},o}(G)$ on a locally finite, connected, quasi-transitive deterministically edge-labelled graph G is regular if and only if all ends of G have size 1, and that it is context-free if and only if all ends have size at most 2. In both of these cases $F_{\mathrm{SAW},o}(z)$ can be computed using an appropriate grammar generating $L_{\mathrm{SAW},o}(G)$. Our second main result generalises these results.

Theorem 6.0.2. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph and let $o \in V(G)$. Then $L_{\mathrm{SAW},o}(G)$ is an MCFL if and only if all ends of G are thin.*

In fact, what we prove is slightly stronger. Every MCFL can be assigned a rank (see Section 2.2 for details); an MCFL is called k-multiple context free if its rank is at most k. It is worth noting that the families of k-MCFLs form a strictly increasing hierarchy, and that 1-MCFLs are exactly the context free languages. We show that the maximal size of an end of G tells us exactly where $L_{\mathrm{SAW},o}(G)$ lies in this hierarchy.

Theorem 6.0.3. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph and let $o \in V(G)$. Then $L_{\mathrm{SAW},o}(G)$ is k-multiple context-free if and only if every end of G has size at most $2k$.*

Applied to Cayley graphs of groups, Theorem 6.0.2 states that the language of self-avoiding walks on a Cayley graph of a group is multiple context-free if and only if the group is virtually free. In particular, the property of having a multiple context-free language of self-avoiding walks is a group invariant.

As mentioned above, we are following a similar approach as Alm and Janson in [1]. There are two key ingredients to this approach: firstly, decomposing the graph into finite parts, and secondly, analysing the restrictions of self-avoiding walks to these parts.

The decomposition into finite parts is formalised by the notion of tree decompositions which will be the subject of Section 6.1. Roughly speaking these are decompositions of a graph into parts intersecting in a tree-like manner. This notion was introduced by Halin [27] in 1976; later it was rediscovered by Robertson and Seymour [49] and plays a central role in the proof of the celebrated Graph Minor Theorem. For our applications it is crucial that the tree decompositions are invariant under some quasi-transitive group of automorphisms. Such tree decompositions have been constructed by Dunwoody and Krön [15], inspired by a similar construction based on edge cuts introduced by Dunwoody in [14].

The restriction of self avoiding walks to the parts of a tree decomposition is captured by the notion of configurations introduced and studied in Section 6.2. Among other things we show that there is a bijection between SAWs and a specific class of configurations called *bounded consistent configurations*, in other words, any SAW can be obtained by piecing together appropriate configurations.

This bijection is central to the rest of the paper because it allows us to work with configurations rather than self avoiding walks; this turns out to be beneficial since configurations (unlike SAWs) carry a recursive structure. In Section 6.3 we use this recursive structure to show that the set of bounded consistent configurations is in bijection with a context free language; Theorem 6.0.1 follows from this fact. In Section 6.4, again using the recursive structure, we show that $L_{\mathrm{SAW},o}(G)$ is an MCFL, thus proving the first half of Theorem 6.0.3. Finally, we combine techniques from Chapter 4 with Theorem 5.0.1 to complete the proof of Theorem 6.0.3.

6.1 Graphs and tree decompositions

Recall from Chapter 2 that we consider the edges of our graph G as being directed and with an involution $e \mapsto \bar{e}$ which inverts the direction.

In addition to the usual walks $p = (v_0, e_1, v_1, \dots, e_n, v_n)$, we also make use of the following more general concept. A *multi-walk* p is a sequence of vertices and edges obtained by stringing together the sequences of vertices and edges corresponding to walks $p_1, \dots, p_k$; the p_i are called the *walk components* of p. In other words, a multi-walk is a sequence of vertices and edges, such that every edge in the sequence is preceded by its initial vertex and succeeded by its terminal vertex. Note that the walk components of a multi-walk are uniquely determined.

For a (multi-)walk p on G and $A \subseteq V(G) \cup E(G)$ we denote by $p \cap A$ the sequence obtained from p by deleting all elements not in A and by $p - A$ the sequence obtained by deleting all elements of A. For a subgraph H of G we write $p \cap H$ for the sequence $p \cap (V(H) \cup E(H))$. In general the sequences $p \cap A$ and $p - A$ need not be multi-walks, but we note that $p - E$ is a multi-walk for $E \subseteq E(G)$, as is $p \cap H$ for any subgraph H of G.

Let us provide a simple example of a Cayley graph used as a running example to demonstrate various constructions throughout this paper. Let C_n denote the cyclic group of order n and consider the group $\Gamma = (C_2 * C_2 * C_2) \times C_3$, that is, the direct product of C_3 and a free product of three copies of C_2. Let a, b and c be the generators of the copies of C_2 and let r be the generator of C_3. Then Γ can be presented as $\langle a, b, c, r \mid a^2 = b^2 = c^2 = r^3 = arar^{-1} = brbr^{-1} = crcr^{-1} = 1_\Gamma \rangle$. Figure 6.1 shows the Cayley graph G of Γ with respect to the symmetric generating set $S = \{a, b, c, r, r^{-1}\}$.

A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \mathcal{V})$, consisting of a tree T and a function $\mathcal{V} : V(T) \to 2^{V(G)}$ assigning a non-empty subset of $V(G)$ to every vertex of T, such that the following three conditions are satisfied:

(T1) $V(G) = \bigcup_{t \in V(T)} \mathcal{V}(t)$.

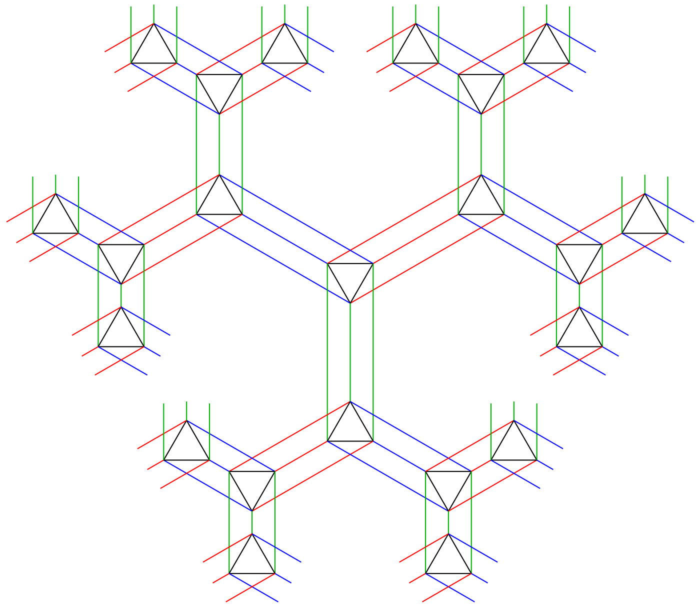


Figure 6.1: A Cayley graph of the group $(C_2 * C_2 * C_2) \times C_3$. Different edge colours correspond to different generators.

(T2) For every $e \in E(G)$ there is a $t \in V(T)$ such that $\mathcal{V}(t)$ contains both vertices that are incident to e.

(T3) $\mathcal{V}(s) \cap \mathcal{V}(t) \subseteq \mathcal{V}(r)$ for every vertex r on the unique s–t-path in T.

The set $\mathcal{V}(t)$ is called the *part* of t. For an edge $e = st$ of T, the intersection $\mathcal{V}(e) = \mathcal{V}(s,t) = \mathcal{V}(s) \cap \mathcal{V}(t)$ $(= \mathcal{V}(t,s))$ is called the *adhesion set* of e. A tree decomposition $(T, \mathcal{V})$ of G is called Γ-*invariant* for a group $\Gamma \leq \mathrm{AUT}(G)$, if every $\gamma \in \Gamma$ maps parts onto parts and thereby induces an automorphism of T. More precisely there is an action of Γ on T by automorphisms such that for every $\gamma \in \Gamma$ and $t \in V(T)$, it holds that $\gamma(\mathcal{V}(t)) = \mathcal{V}(\gamma t)$.

The tree decomposition $\mathcal{T}$ is said to *distinguish* two given ends ω_1 and ω_2 of G if there is some edge e of T such that the adhesion set $\mathcal{V}(e)$ separates ω_1 and ω_2. Moreover $\mathcal{T}$ distinguishes the two ends *efficiently*, if one of its adhesion sets has the smallest size of all subsets of vertices of G separating ω_1 and ω_2. We call $\mathcal{T}$ *reduced* if every adhesion set efficiently distinguishes some pair of ends of G and no two parts corresponding to adjacent vertices of T coincide.

Recall that a locally finite graph G is accessible if there is a natural number k such that any two ends can be separated by a set of vertices of size at most k. The following theorem is closely related to the works of Dicks and Dunwoody (see [14] and [10]) and is a direct consequence of Theorem 6.4 in [28].

Theorem 6.1.1. *Let G be a simple, locally finite, connected, accessible graph and let Γ be a group acting quasi-transitively on G. Then there is a Γ-invariant tree decomposition*

$(T, \mathcal{V})$ *of* G *efficiently distinguishing all ends of* G *and an action of* Γ *on* T *witnessing the* Γ*-invariance of* $(T, \mathcal{V})$ *with only finitely many* Γ*-orbits on* $E(T)$.

For our purpose, it is necessary for tree decompositions to additionally be reduced. However, this is not really a restriction, as the following construction shows.

Let $\mathcal{T} = (T, \mathcal{V})$ be a tree decomposition of G and F be a subset of edges of T. The contraction of F in $\mathcal{T}$ is the pair $\mathcal{T}/F = (T/F, \mathcal{V}/F)$ defined in the following way. The tree T/F consists of a single vertex for every component of the graph $(V(T), F)$ obtained from T by only keeping the edges in F; for a vertex t of T/F let $[t]_F$ denote the vertex set of the corresponding component. Two different vertices $[s]_F$ and $[t]_F$ of T/F are connected by an edge if and only if there are $s' \in [s]_F$ and $t' \in [t]_F$, which are adjacent in T. The part corresponding to $[t]_F \in V(T/F)$ is $(\mathcal{V}/F)([t]_F) = \bigcup_{s \in [t]_F} \mathcal{V}(s)$. It is not hard to see that $\mathcal{T}/F$ is a tree decomposition of G.

Starting from a tree decomposition $\mathcal{T} = (T, \mathcal{V})$ provided by the previous theorem, we can construct a reduced tree decomposition as follows. Let the set F consist of all edges e of T such that the adhesion set $\mathcal{V}(e)$ does not minimally separate any pair of ends of G. It is easy to check that the contraction $\mathcal{T}/F$ is a tree decomposition retaining all properties mentioned in Theorem 6.1.1 and additionally every adhesion set minimally separates two ends of G. In a second step we contract all edges of T/F connecting two vertices whose parts coincide to obtain a reduced tree decomposition as in the following corollary.

Corollary 6.1.2. *Let* G *be a simple, locally finite, connected, accessible graph and let* Γ *be a group acting quasi-transitively on* G*. Then there is a reduced* Γ*-invariant tree decomposition* $\mathcal{T} = (T, \mathcal{V})$ *of* G *efficiently distinguishing all ends of* G *such that there are only finitely many* Γ*-orbits on* $E(T)$.

Let us again look at the Cayley graph G of the group $\Gamma = (C_2 * C_2 * C_2) \times C_3$ given in Figure 6.1. We already mentioned that Γ acts freely on G by left multiplication. A reduced Γ-invariant tree decomposition $(T, \mathcal{V})$ of G as provided by the previous corollary is shown in Figure 6.2.

We require the following four important properties of tree decompositions $\mathcal{T} = (T, \mathcal{V})$ obtained from this corollary.

(P1) The size of all adhesion sets is bounded from above by some $k \in \mathbb{N}$.

(P2) For every $K \subseteq V(G)$ there are only finitely many edges e of T with $K = \mathcal{V}(e)$.

(P3) All parts are finite if and only if all ends of G have finite size.

(P4) If all ends of G have finite size, then T is locally finite.

Firstly, $\mathcal{T}$ is reduced, so every adhesion set minimally separates some pair of ends of G. The graph G is accessible, so (P1) holds.

For the proof of the other properties, we need some notation. A *separation* of G is a pair (A, B) of vertex sets such that $G[A] \cup G[B] = G$, which means that there are no edges between $A \setminus B$ and $B \setminus A$. The intersection $A \cap B$ is called the *separator* of

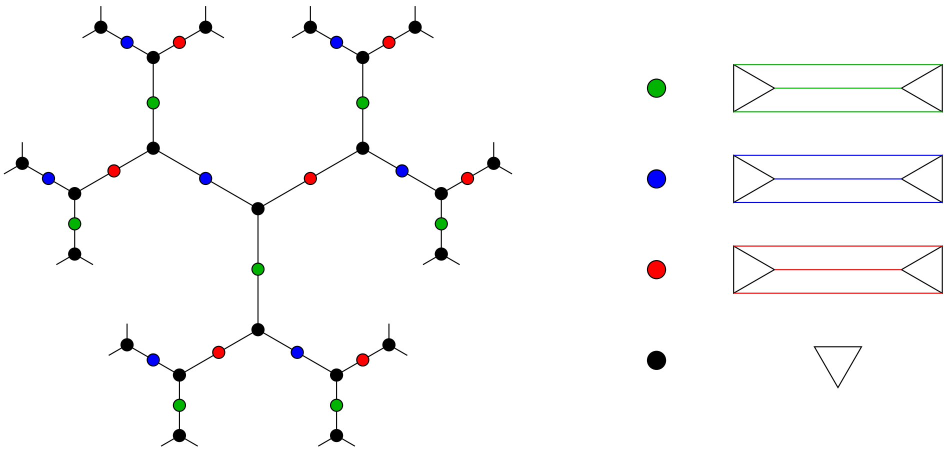


Figure 6.2: Decomposition tree T of the Cayley graph G. The vertex-colouring indicates the four orbits of the group action on the vertices of T. The subgraphs of G induced by the different parts are shown at the right side.

(A, B). Note that every edge e of the tree decomposition $\mathcal{T}$ corresponds to a separation of G with separator $\mathcal{V}(e)$. Removal of e splits T into two components T_1 and T_2 and $\left(\bigcup_{t\in V(T_1)} \mathcal{V}(t), \bigcup_{t\in V(T_2)} \mathcal{V}(t)\right)$ is the separation of G induced by e.

Clearly any given finite set K of vertices of a locally finite graph G can occur only finitely many times as the separator of a separation (A, B) of G. Indeed, $G - K$ has only finitely many components and each of these components has to be fully contained in either A or B. This observation together with the following lemma yields property (P2).

Lemma 6.1.3. *Let G, Γ and $\mathcal{T} = (T, \mathcal{V})$ be as in Corollary 6.1.2. Then every separation (A, B) of G corresponds to at most two edges of T.*

Proof. Let e_1, e_2 be two edges of T inducing the separation (A, B). We first show that e_1 e_2 share a common vertex. For $i = 1, 2$ let T_i^A and T_i^B be the components of $T - e_i$ corresponding to A and B, respectively. The separation (A, B) is induced by an edge of T and every adhesion set separates two ends of G, so A and B are both infinite sets and $\mathcal{V}(e_1) = \mathcal{V}(e_2) = A \cap B$ is finite. We may assume without loss of generality that $T_1^A \subseteq T_2^A$ and $T_2^B \subseteq T_1^B$. Let s be a vertex on the unique shortest path P connecting e_1 and e_2 in T. Property (T3) of tree decompositions yields

$$A \cap B = \mathcal{V}(e_1) \cap \mathcal{V}(e_2) \subseteq \mathcal{V}(s).$$

Moreover s is a vertex of T_1^B and T_2^A, so in particular

$$\mathcal{V}(s) \subseteq \bigcup_{t\in V(T_1^B)} \mathcal{V}(t) \cap \bigcup_{t\in V(T_2^A)} \mathcal{V}(t) = A \cap B,$$

implying that the part $\mathcal{V}(s)$ is equal to $A \cap B$ for every vertex s of P. Reducedness of $\mathcal{T}$ implies that P consists of a single vertex and thus both e_1 and e_2 contain s.

Let e_3 be any edge of T inducing the separation (A, B) and let T_3^A and T_3^B be the components of $T - e_3$ corresponding to A and B, respectively. Then e_3 intersects e_1 and e_2 and thus contains their common vertex r. Finally, if e_3 is different from e_1 and e_2, then either $T_2^B \subseteq T_3^A$ or $T_1^A \subseteq T_3^B$, leading to a contradiction. □

The basis for the proof of (P3) is the following lemma, which is closely related to Proposition 4.5 in [28].

Lemma 6.1.4. *Let G, Γ and $(T, \mathcal{V})$ be as in Theorem 6.1.1. Then for every vertex t of T, the induced subgraph $G[\mathcal{V}(t)]$ is a quasi-transitive graph.*

Proof. We show that the set-wise stabiliser $\Gamma_{\mathcal{V}(t)}$ of $\mathcal{V}(t)$ in Γ acts quasi-transitively on $G[\mathcal{V}(t)]$. If $u \in \mathcal{V}(t)$ does not lie in any adhesion set, then neither does any image of u under a graph automorphism. In particular, any $\gamma \in \Gamma$ mapping u to some vertex $v \in \mathcal{V}(t)$ fixes $\mathcal{V}(t)$ and thus, under the action of the stabiliser of $\mathcal{V}(t)$ there are only finitely many orbits of vertices in $\mathcal{V}(t)$ not contained in any adhesion set.

Let m be the finite number of Γ-orbits on $E(T)$. Whenever $\gamma \in \Gamma$ fixes t and maps a neighbour s of t onto some other neighbour s', γ lies in $\Gamma_{\mathcal{V}(t)}$ and maps the adhesion set $\mathcal{V}(s, t)$ onto $\mathcal{V}(s', t)$. Therefore the number of orbits of adhesion sets under the action of $\Gamma_{\mathcal{V}(t)}$ is at most $2m$. As every adhesion set contains at most k elements, $\Gamma_{\mathcal{V}(t)}$ acts with at most $2mk$ orbits on vertices of $\mathcal{V}(t)$ lying in adhesion sets of the tree decomposition. We conclude that $\Gamma_{\mathcal{V}(t)}$ acts with finitely many orbits on $\mathcal{V}(t)$. □

Our goal is to apply the following proposition to a part $\mathcal{V}(t)$, but in general $G[\mathcal{V}(t)]$ need not be connected, so some additional work is necessary.

Proposition 6.1.5 (Thomassen [55, Proposition 5.6])**.** *If G is a locally finite, connected, quasi-transitive graph with only one end, then this end is thick.*

In order to prove the first implication of (P3), assume that there is some vertex t of T such that the part $\mathcal{V}(t)$ is infinite. Let H be the subgraph of G obtained from the induced subgraph $G[\mathcal{V}(t)]$ in the following way. For every edge e of T incident to t add all shortest paths between any pair of vertices in the adhesion set $\mathcal{V}(e)$. Note that since the stabiliser of $\mathcal{V}(t)$ acts quasi-transitively, the length of these paths is bounded by some constant $m \in \mathbb{N}$.

Any walk on G connecting two vertices of $\mathcal{V}(t)$ consists of sub-walks on $G[\mathcal{V}(t)]$ and detours leaving $\mathcal{V}(t)$ via some adhesion set $\mathcal{V}(e)$ and re-entering via the same set. These detours can be replaced by a shortest detour, which is by definition a walk on H, so H is connected. Furthermore, $\Gamma_{\mathcal{V}(t)}$ acts quasi-transitively on H because it acts with finitely many orbits on the edges of T and thus on the adhesion sets contained in $\mathcal{V}(t)$.

Assume for a contradiction that H has more than one end. Then there must be a separation (A, B) of H with finite separator $A \cap B$ such that both A and B are infinite. Let K be the union of $A \cap B$ and all adhesion sets $\mathcal{V}(s, t)$ containing both a vertex a of $A \setminus B$ and a vertex b of $B \setminus A$. Due to construction of H it contains an a–b-path of length at most m and any such path must intersect the separator $A \cap B$. Thus any vertex of K lies at distance at most m from $A \cap B$, implying that K is finite.

It is not hard to see that $G - K$ contains no $(A \setminus B)$–$(B \setminus A)$-path. Indeed, assume that P is a $(A \setminus B)$–$(B \setminus A)$-path in G which does not intersect $A \cap B$. Then P is a detour leaving and re-entering $G[\mathcal{V}(t)]$ via some adhesion set $\mathcal{V}(s,t)$ intersecting $A \setminus B$ and $B \setminus A$ and thus contains at least one vertex in K.

Finally, let R_A and R_B be rays in $H[A]$ and $H[B]$, respectively. Then the ends ω_A and ω_B of G containing R_A and R_B are different, as R_A and R_B are separated by K. On the other hand, each of R_A and R_B contains infinitely many vertices of $\mathcal{V}(t)$, so they are not separated by any adhesion set of the tree decomposition $\mathcal{T}$. This contradicts the fact that $\mathcal{T}$ distinguishes all ends of G. We conclude that the infinite connected graph H has precisely one end. By Proposition 6.1.5 the end of the one-ended quasi-transitive graph H is thick. The graph G contains H as a subgraph and thus inherits the thick end of H.

On the other hand, it is not hard to see that all ends of G are thin, if all parts of $\mathcal{T}$ are finite. For any set of disjoint rays in the same end of G there must be some adhesion set intersecting each of the rays. The size of adhesion sets is at most k, so every end of G has size at most k.

Finally (P4) is a consequence of (P2) and (P3). Every edge e incident to a vertex s of T corresponds to some adhesion set $\mathcal{V}(e)$ which is a subset of the part $\mathcal{V}(t)$. But the finite part $\mathcal{V}(t)$ has only finitely many different subsets and each of them occurs only finitely often as an adhesion set in $(T, \mathcal{V})$.

6.1.1 Rooted tree decompositions

A *rooted graph* is a pair (G, o) consisting of a graph G and a designated root vertex $o \in V(G)$. Let (G, o) be a simple, locally finite, connected, rooted graph. A *rooted tree decomposition* $\mathcal{T} = (T, \mathcal{V}, r)$ of (G, o) consists of a tree decomposition $(T, \mathcal{V})$ of G and a fixed vertex r of T such that o is contained in $\mathcal{V}(r)$; note that there can be multiple valid choices for r since o can be contained in more than one part. We call r the *root* of T and $\mathcal{V}(r)$ the *root part* of the decomposition. We use the definitions and notations for rooted trees provided in Chapter 2.

For every $t \in V(T)$ we introduce a graph $\mathcal{G}(t)$ on the vertex set $\mathcal{V}(t)$. Let us start by defining a map $\mathcal{E} : V(T) \to 2^{E(G)}$ by $\mathcal{E}(r) = E(G[\mathcal{V}(r)])$ and

$$\mathcal{E}(t) = E(G[\mathcal{V}(t)]) \setminus E(G[\mathcal{V}(t^{\uparrow})]) \quad \text{for } t \neq r,$$

where $t^{\uparrow}$ denotes the parent of t in the rooted tree T. Edges in $\mathcal{E}(t)$ are called *(non-virtual) t-edges*. Property (T2) of tree decompositions implies that for every edge e of G there is some $t \in V(T)$ such that $e \in \mathcal{E}(t)$. Fix some edge e of G and let S be the set of all vertices s of T such that $\mathcal{V}(s)$ contains both endpoints of e. By property (T3) the induced subgraph $T[S]$ is connected and thus the forefather t of S is contained in S. It is easy to see that t is the unique vertex of T with $e \in \mathcal{E}(t)$, so the edge set of G is the disjoint union $E(G) = \biguplus_{t \in V(T)} \mathcal{E}(t)$.

Additionally we introduce for every edge $e = st$ of T a new set of *virtual e-edges* $\mathcal{E}(e) = \mathcal{E}(st)$, such that every pair of vertices of $\mathcal{V}(e)$ is connected by an edge in $\mathcal{E}(e)$. In other words, the *e-graph* $\mathcal{G}(e) = (\mathcal{V}(e), \mathcal{E}(e))$ is a complete graph. In order to enhance readability, we usually write $\mathcal{E}(s,t)$ instead of $\mathcal{E}(st)$ and $\mathcal{G}(s,t)$ instead of $\mathcal{G}(st)$.

Finally, we assign to every vertex t of T the *t-graph*

$$\mathcal{G}(t) = \left(\mathcal{V}(t)\ ,\ \mathcal{E}(t) \uplus \biguplus_{e\colon e^-=t} \mathcal{E}(e) \right).$$

Note that $\mathcal{G}(t)$ generally is not a simple graph since $\mathcal{E}(t)$ and the various sets $\mathcal{E}(s,t)$ potentially contain edges with the same endpoints.

For convenience we extend the definition of $\mathcal{G}$ to subsets S of the vertex set $V(T)$ by taking the union of all graphs $\mathcal{G}(t)$ for $t \in S$ and removing all virtual edges corresponding to edges of $T[S]$, so that only virtual edges corresponding to edges of T with exactly one endpoint in S remain. In terms of sets,

$$\mathcal{G}(S) = \left(\bigcup_{t\in S} \mathcal{V}(t)\ ,\ \bigcup_{t\in S} \mathcal{E}(t) \uplus \biguplus_{e\colon e^-\in S,\, e^+\notin S} \mathcal{E}(e) \right).$$

Again, we visualise these concepts using the Cayley graph G from Figure 6.1 and its tree decomposition $(T, \mathcal{V})$ shown in Figure 6.2. For a given root o of G, denote by r the unique vertex of T such that the part $\mathcal{V}(r)$ has cardinality 3 and contains o. Then $(T, \mathcal{V}, r)$ is a rooted tree decomposition of the rooted graph (G, o). Figure 6.3 shows a portion of the decomposition tree T and the t-graphs for vertices t contained in it. Compare this to Figure 6.2 and note that the t-graphs on the parts are generally neither subgraphs (due to virtual edges) nor supergraphs (due to some missing non-virtual edges) of the induced graphs on the parts.

6.1.2 Cones and cone types

As for rooted trees, we define for a rooted tree decomposition $(T, \mathcal{V}, r)$ the *cone* at a vertex $s \in V(T)$ of T as the set K_s containing all descendants of s, that is, all vertices t such that s lies on the t–r-path in T.

Let $\Gamma \subseteq \mathrm{AUT}(G)$ be a group acting on G. We say that two vertices s and t of T different from r are *cone-equivalent* and write $s \sim_K t$, if there is a $\gamma \in \Gamma$ mapping s to t and the parent $s^\uparrow$ of s to the parent $t^\uparrow$ of t. The root r is only cone equivalent to itself. Clearly $\sim_K$ is an equivalence relation and we call the equivalence classes of vertices *cone types* of the rooted tree decomposition $\mathcal{T}$.

Note that if $\gamma \in \Gamma$ witnesses the cone equivalence of s and t, then γ maps the cone K_s onto the cone K_t; in this case we also call the cones K_s and K_t *equivalent*. The following lemma tells us that the graphs $\mathcal{G}(s)$ and $\mathcal{G}(t)$ are isomorphic whenever s and t are cone equivalent.

Lemma 6.1.6. *Any $\gamma \in \Gamma$ witnessing the cone equivalence of two vertices s and t of T can be extended to a graph isomorphism between the graphs $\mathcal{G}(s)$ and $\mathcal{G}(t)$.*

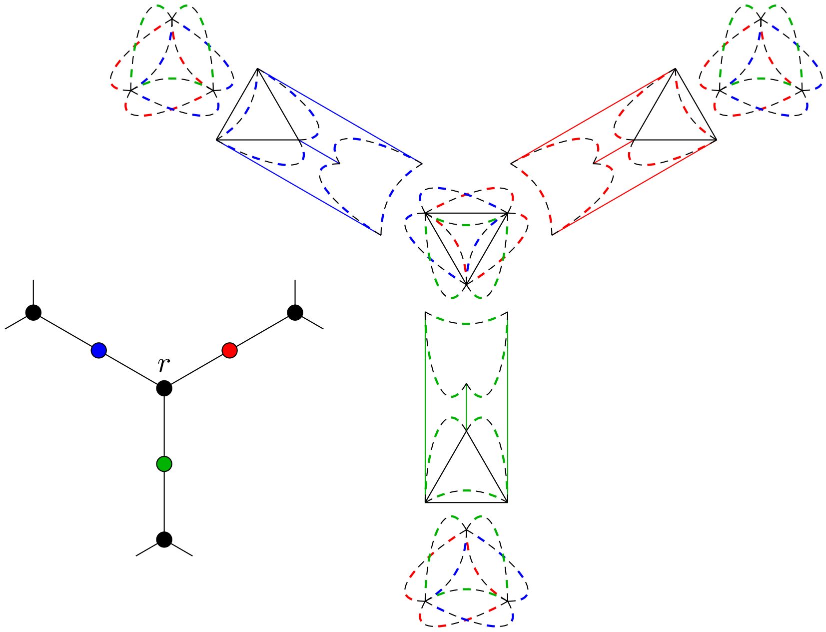


Figure 6.3: Decomposition tree T of the Cayley graph G and its corresponding t-graphs. Dashed edges are virtual e-edges; if they are shared by different t-graphs, they have the same shape in these graphs.

Proof. Recall that by definition

$$\mathcal{G}(s) = \left(\mathcal{V}(s), \mathcal{E}(s) \cup \bigcup_{e|e^-=s} \mathcal{E}(e) \right),$$

where $\mathcal{E}(s) = E(G[\mathcal{V}(s)]) \setminus E(G[\mathcal{V}(s^\uparrow)])$. Note that $e \in \mathcal{E}(s)$ if and only if $\gamma(e) \in \mathcal{E}(t)$, because γ maps $s^\uparrow$ onto $t^\uparrow$. We extend γ in the natural way to virtual edges in $\mathcal{E}(s, u)$ by mapping e onto the edge $\gamma(e) \in \mathcal{E}(t, \gamma(u))$ connecting $\gamma(e^-)$ and $\gamma(e^+)$. The result is a graph isomorphism between $\mathcal{G}(s)$ and $\mathcal{G}(t)$. □

Lemma 6.1.7. *Let $(T, \mathcal{V}, r)$ be a reduced, rooted tree decomposition of a locally finite graph G. If there is a subgroup $\Gamma \leq \mathrm{AUT}(G)$ such that $(T, \mathcal{V})$ is Γ-invariant and the action of Γ on $E(T)$ has finitely many orbits, then the number of cone types of $(T, \mathcal{V}, r)$ is finite.*

Proof. Let $s \sim_K t$ be two vertices in $V(T) \setminus \{r\}$. Then there is a $\gamma \in \Gamma$ mapping the edge $s^\uparrow s$ onto $t^\uparrow t$ while preserving direction. This implies that the number of cone types can be at most two times the number of edge orbits of Γ acting on T plus one, where the additional type is the type of the root r, the only vertex without a parent. In particular the number of cone types is finite. □

6.2 Configurations

Let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree decomposition of a simple, locally finite, connected, rooted graph (G, o). A *configuration* on $S \subseteq V(T)$ with respect to $\mathcal{T}$ is a map $C = (P, X)$ assigning to each vertex $s \in S$ a pair $C(s) = (P(s), X(s))$, such that for every $s \in S$ one of the following alternatives holds.

(a) $X(s) \in V(T)$ is either s or a neighbour of s in T, and $P(s)$ is a self-avoiding walk on $\mathcal{G}(s)$ starting in $\mathcal{V}(s^{\uparrow}, s)$ (or at o if $s = r$) and, if $X(s) \neq s$, ending in $\mathcal{V}(s, X(s))$. Moreover, if $X(s) = s$ then $P(s)$ must end with a non-virtual edge.

(b) $X(s) = s^{\uparrow}$ and $P(s) = \emptyset$ is the empty walk; this is called the *empty configuration* and can only occur for $s \neq r$.

We call $X(s)$ the *exit direction* of s. A *configuration* on a vertex s of T is an image pair $C(s) = (P(s), X(s))$ of a configuration $C = (P, X)$ on the set $S = \{s\}$. Note that by Lemma 6.1.6, the sets of configurations on two cone equivalent vertices are the same up to isomorphism.

Intuitively, configurations model the behaviour of SAWs on single parts of the tree decomposition in the following way. Let p be a self-avoiding walk on G starting at the root o. For any $t \in V(T)$ we define a projection p_t of p onto the graph $\mathcal{G}(t)$. First take all vertices and edges of p contained in $\mathcal{G}(t)$ to obtain the multi-walk $p \cap \mathcal{G}(s, t)$. Every detour of p in some other part $\mathcal{V}(s)$ with s adjacent to t in T corresponds to a virtual edge of $\mathcal{E}(s, t)$ connecting the same endpoints as the detour. By replacing these detours by their "shortcuts", we end up with a walk p_t on $\mathcal{G}(t)$. Note that p_t might be the empty walk for many vertices t. Let u be the vertex of T such that the final edge of p is contained in $\mathcal{G}(u)$. Let $x_u = u$ and for $t \neq u$ let x_t be the neighbour of t on the unique t–u-path in T. Then the function C defined by $C(t) = (p_t, x_t)$ defines a configuration on $V(T)$ with respect to $\mathcal{T}$. This shows that starting from a SAW, we can give a configuration describing the behaviour of the walk when restricted to single parts.

In order to reverse the above construction, we would like to combine configurations on the single parts into SAWs on G. To this end, two more properties are needed. Firstly, since SAWs are finite, only finitely many parts can make non-trivial contributions. Secondly, configurations on the parts that contribute non-trivially must fit together in a certain way. These two properties are implied by the notions of boundedness and consistency of configurations defined below. In what follows, let $C = (P, X)$ be a configuration on $S \subseteq V(T)$.

The *weight* $\|C\|$ of C is the total number of non-virtual edges contained in all the walks $P(s)$ for $s \in S$, so $\|C\| = \sum_{s \in S} \|C(s)\|$, where $\|C(s)\|$ denotes the number of non-virtual edges in $P(s)$. The configuration C is called *boring on* $s \in S \setminus \{r\}$ (we also say that $C(s)$ is *boring*) if $X(s) = s^{\uparrow}$ and $P(s)$ contains only edges in $\mathcal{E}(s^{\uparrow}, s)$. In particular, the empty configuration is boring and all boring configurations have weight 0. We call a configuration C *bounded*, if $C(s)$ is boring for all but finitely many $s \in S$.

Let $s, t \in S$ be adjacent vertices; without loss of generality assume $s = t^{\uparrow}$. The configurations $C(s)$ and $C(t)$ are called *compatible*, if either $P(s) \cap \mathcal{V}(s, t) = \emptyset$ and $C(t)$ is the empty configuration, or if they satisfy the following four conditions.

(C1) The ordered sequences of vertices obtained by intersecting the walks $P(s)$ and $P(t)$ with $\mathcal{V}(s,t)$ coincide,

$$P(s) \cap \mathcal{V}(s,t) = (v_1, \dots, v_l) = P(t) \cap \mathcal{V}(s,t).$$

(C2) For every $i \in \{1, \dots, l-1\}$

$$v_i P(s) v_{i+1} \cap \mathcal{E}(s,t) = \emptyset \iff v_i P(t) v_{i+1} \cap \mathcal{E}(s,t) \neq \emptyset.$$

(C3) $X(s) = t \iff X(t) \neq s$.

(C4) If $X(s) = t$, then $P(s)$ ends in v_l, otherwise $P(t)$ ends in v_l.

The configuration C is called *consistent*, if the configurations $C(t^\uparrow)$ and $C(t)$ are compatible whenever both t and $t^\uparrow$ are in S.

Configurations on the complete vertex set $V(T)$ of the tree decomposition $\mathcal{T}$ are called configurations on $\mathcal{T}$ and the set of all bounded consistent configurations on $\mathcal{T}$ is denoted by $\mathcal{C}_\mathcal{T}$.

Remark 6.2.1. By (C3), a consistent configuration $C = (P, X) \in \mathcal{C}_\mathcal{T}$ induces an orientation of the edges of T. Clearly any vertex $s \in V(T)$ can be incident to at most one vertex t with $X(t) \neq s$, namely the vertex $X(s)$ in the case $X(s) \neq s$. It is not hard to see that if there is a vertex s with $X(s) = s$, then for every other vertex t of T, $X(t)$ lies on the unique t–s-path in T; in other words, $X(t)$ points towards s. In particular there can be at most one such vertex s. Also note that in the case where C is bounded there is exactly one vertex s with $X(s) = s$. This vertex s can be found by starting at any vertex of T and following exit directions.

Let us go back to the Cayley graph G from Figure 6.1. Using the decomposition tree T and the t-graphs from Figure 6.3, an example of a bounded consistent configuration (W, X) on T is shown in Figure 6.4. Note that there are only 3 vertices carrying non-boring configurations and that all exit directions point towards the unique vertex s of T with $X(s) = s$.

The following extension lemma can be seen as the reason why boring configurations are indeed not interesting to us. More precisely, it shows that a bounded consistent configuration on $\mathcal{T}$, is uniquely determined by the (finitely many) non-boring configurations. Moreover, it tells us that under certain conditions a consistent configuration on a finite set $S \subseteq V(T)$ can be extended to a bounded consistent configuration on $\mathcal{T}$.

Lemma 6.2.2. *Let $s, t \in V(T)$ such that $s = t^\uparrow$ and let $c_s = (p_s, x_s)$ be a configuration on s such that $p_s \cap \mathcal{E}(s,t) = \emptyset$ and $x_s \neq t$. Then there is a unique configuration c_t on t compatible with c_s, and this configuration c_t is boring.*

Proof. Suppose $c_t = (p_t, x_t)$ is a configuration on t such that c_t and c_s are compatible. Then $x_t = s$ by (C3) and $p_t \cap \mathcal{V}(s,t) = (v_1, \dots, v_l) = p_s \cap \mathcal{V}(s,t)$ by (C1). Property (C2) implies that the sub-walks $v_i p_t v_{i+1}$ contain only the virtual edge $v_i v_{i+1}$ in $\mathcal{E}(s,t)$ and

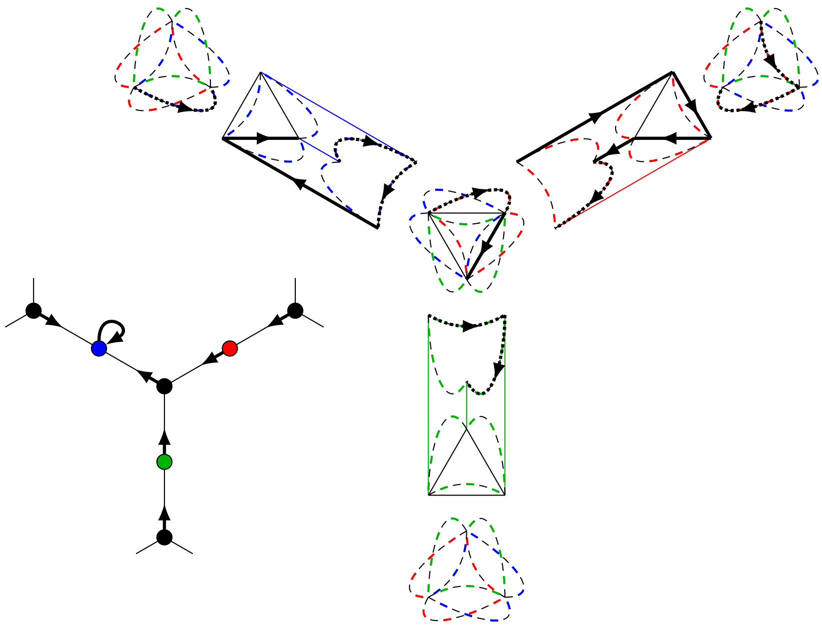


Figure 6.4: A bounded consistent configuration (W, X) on the tree decomposition $\mathcal{T}$ of the Cayley graph G. Edges of the walk $W(t)$ on the t-graph $\mathcal{G}(t)$ are drawn bold and decorated with arrows according to their direction. Exit directions of vertices of T are also denoted by arrows pointing from a vertex t to a vertex s if $X(t) = s$.

by the definition of configurations and (C4) the walk p_t starts at v_1 and ends at v_l. We conclude that the configuration c_t on t is unique and boring. Moreover, the considerations above can be used to construct such a configuration, and in particular such a configuration exists. □

Our goal in this section is to establish a one-to-one correspondence ψ_r between bounded consistent configurations $\mathcal{C}$ on the rooted tree decomposition $\mathcal{T} = (T, \mathcal{V}, r)$ and self-avoiding walks of length at least 1 on the underlying graph G starting at its root o. The main idea is to contract the sub-tree induced by all vertices of T carrying non-boring configurations to a single vertex. By also contracting the corresponding configurations, only a single non-boring configuration remains; its walk is a walk on G and we denote it by $\psi_r(C)$.

In Section 6.1 we already discussed how to contract a set F of edges of a tree decomposition $\mathcal{T}$ to obtain a contracted tree decomposition $\mathcal{T}/F$. Let us repeat this process for rooted tree decompositions and configurations on those tree decompositions. As we are only interested in contractions of finite sets of edges, we first focus on the special case where a single edge is contracted.

The following definition of the contraction of a rooted tree decomposition coincides with our earlier definition of contractions of tree decompositions; the root part of the contraction is simply the equivalence class of r. We still give a detailed definition in order to introduce some notation.

Let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree decomposition of a simple, locally finite, connected, rooted graph (G, o) and let $f \in E(T)$. We may without loss of generality assume that $f^- = (f^+)^\uparrow$ (if not, use the reversed edge). Define the contraction $\mathcal{T}/f = (T/f, \mathcal{V}/f, r/f)$ as follows.

The tree T/f is obtained from T by identifying the two endpoints f^- and f^+ of f and deleting the edge f. More precisely, T/f can be described as follows. The vertex set of T/f is obtained from the vertex set of T by replacing f^- and f^+ by a single vertex t_f. Every edge $e \in E(T) \setminus \{f\}$ not incident to f corresponds to an edge in T/f with the same endpoints. Every edge $e = st$ of T where t is an endpoint of f corresponds to an edge connecting s and t_f in T/f. We abuse notation and denote the edge corresponding to e in T/f by e as well. The part $\mathcal{V}/f(t_f)$ is defined as $\mathcal{V}(f^-) \cup \mathcal{V}(f^+)$; for every other vertex of T/f we define $\mathcal{V}/f(t) = \mathcal{V}(t)$. Finally, if r is incident to f, then let $r/f = t_f$, otherwise let $r/f = r$.

Denote the parent of $t \in V(T/f)$ by $t^{\uparrow/f}$. From the assumption $f^- = (f^+)^\uparrow$ it follows that $(t_f)^{\uparrow/f} = (f^-)^\uparrow$, unless $f^- = r$, in this case $t_f = r/f$ has no parent. For every other vertex of T/f we have $t^{\uparrow/f} = t_f$ if $t^\uparrow \in \{f^-, f^+\}$, and $t^{\uparrow/f} = t^\uparrow$ otherwise. Note that if an edge $e \in E(T) \setminus \{f\}$ connects t to $t^\uparrow$ (or f^- to $(f^-)^\uparrow$), then the corresponding edge in T/f that is also denoted by e connects t to $t^{\uparrow/f}$ (or t_f to $(t_f)^{\uparrow/f}$).

For $e \in E(T/f)$ let $\mathcal{V}/f(e)$ and $\mathcal{E}/f(e)$ denote the adhesion set corresponding to e and the set of e-edges with respect to the tree decomposition $\mathcal{T}/f$, respectively. For $t \in V(T/f)$ let $\mathcal{E}/f(t)$ and $\mathcal{G}/f(t)$ denote the set of t-edges and the t-graph with respect to the tree decomposition $\mathcal{T}/f$, respectively. Using property (T3) of tree decompositions, it is not hard to see that $\mathcal{V}/f(e) = \mathcal{V}(e)$, $\mathcal{E}/f(e) = \mathcal{E}(e)$, $\mathcal{E}/f(t_f) = \mathcal{E}(f^-) \cup \mathcal{E}(f^+)$, and $\mathcal{E}/f(t) = \mathcal{E}(t)$ for $t \neq t_f$. It follows that

$$\mathcal{G}/f(t) = \begin{cases} (\mathcal{G}(f^-) \cup \mathcal{G}(f^+)) - \mathcal{E}(f) & \text{if } t = t_f, \\ \mathcal{G}(t) & \text{otherwise.} \end{cases}$$

Next we define contractions of configurations. Let $C = (P, X)$ be a bounded consistent configuration on $\mathcal{T}$. For the definition of the contracted configuration C/f, assume again without loss of generality that $f^- = (f^+)^\uparrow$, and let $P(f^-) \cap \mathcal{V}(f) = (v_1, \dots, v_l) = P(f^+) \cap \mathcal{V}(f)$, where the last equality follows from (C1). Let $t_0 = f^-$. For $1 \leq j \leq l-1$, let $t_j \in \{f^-, f^+\}$ be such that $P(t_j) \cap \mathcal{E}(f) = \emptyset$; note that this uniquely defines a vertex by (C2). If $X(f^-) = f^+$, then let $t_l = f^+$, otherwise let $t_l = f^-$. Define the walk p_f as the concatenation

$$P(t_0) v_1 P(t_1) v_2 \dots v_l P(t_l).$$

In other words, p_f is obtained from $P(f^-)$ and $P(f^+)$ by deleting all edges in $\mathcal{E}(f)$ and then piecing the walk components of the resulting multi-walks together in a consistent manner, see Figure 6.5.

The contraction $C/f = (P/f, X/f)$ of the configuration C is defined as follows. For

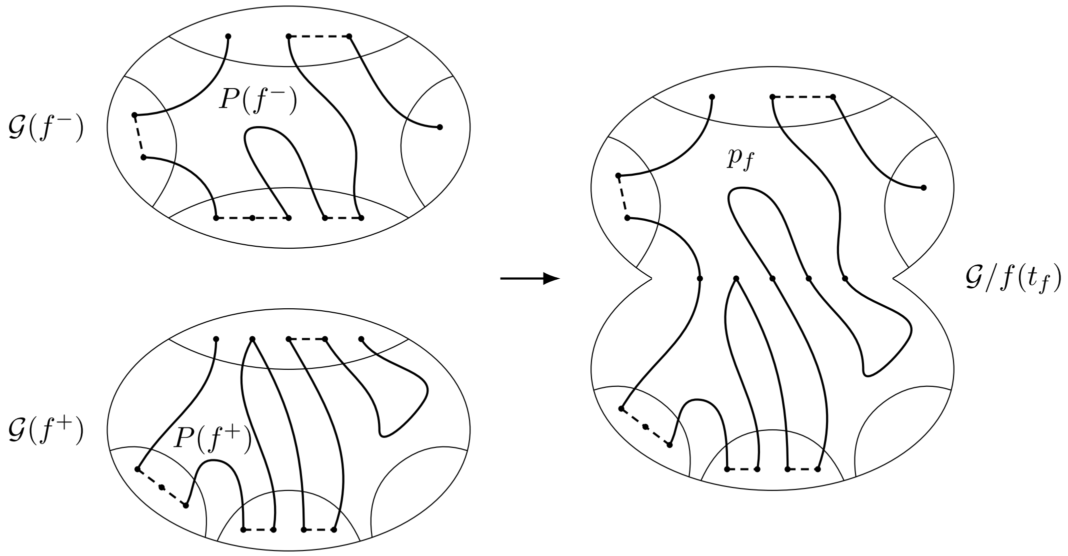


Figure 6.5: Combining walks $P(f^-)$ and $P(f^+)$ of compatible configurations on the endpoints of f into a walk p_f on $\mathcal{G}/f(t_f)$.

the contracted vertex t_f, let

$$P/f(t_f) = p_f \qquad \text{and} \qquad X/f(t_f) = \begin{cases} X(f^-) & \text{if } X(f^-) \notin \{f^-, f^+\}, \\ X(f^+) & \text{if } X(f^+) \notin \{f^-, f^+\}, \\ t_f & \text{otherwise.} \end{cases}$$

Note that by (C3), the conditions in the first two cases in the definition of X/f cannot be satisfied simultaneously, and in the third case $X(f^-) = X(f^+) \in \{f^-, f^+\}$ holds. For $t \neq t_f$ we define

$$P/f(t) = P(t) \qquad \text{and} \qquad X/f(t) = \begin{cases} X(t) & \text{if } X(t) \notin \{f^-, f^+\}, \\ t_f & \text{otherwise.} \end{cases}$$

Note the similarity between the definition of $X/f(t)$ and our observations about $t^{\uparrow/f}$ above; clearly, if $X(t) = t^{\uparrow}$, then $X/f(t) = t^{\uparrow/f}$.

Lemma 6.2.3. *The walk p_f is a self-avoiding walk on $\mathcal{G}/f(t_f)$ satisfying $p_f \cap \mathcal{G}(f^-) = P(f^-) - \mathcal{E}(f)$ and $p_f \cap \mathcal{G}(f^+) = P(f^+) - \mathcal{E}(f)$. In particular, the set of edges contained in p_f consists of the edge sets of $P(f^-) - \mathcal{E}(f)$ and $P(f^+) - \mathcal{E}(f)$.*

Proof. If $P(f^+)$ is the empty walk, then $p_f = P(f^-)$ and all claimed properties are trivially satisfied, so assume that $P(f^+) \neq \emptyset$. Since C is a configuration, $P(f^+)$ must start in v_1, that is, $P(f^+)v_1$ is a trivial walk only consisting of v_1. By (C2), if $t_j = f^-$, then $P(f^+)$ contains the edge $v_j v_{j+1} \in \mathcal{E}(f)$, and if $t_j = f^+$, then $P(f^-)$ contains the edge $v_j v_{j+1} \in \mathcal{E}(f)$. Properties (C3) and (C4) imply that if $t_l = f^+$, then $v_l P(t^-)$ is trivial and vice versa. Combining these observations with the fact that $P(f^-)$ can be decomposed as $P(f^-) = P(f^-)v_1 P(f^-)v_2 \dots v_l P(f^-)$ we conclude that

$$p_f \cap \mathcal{G}(f^-) = P(f^-) - \mathcal{E}(f),$$

and similarly for f^+. This implies that p_f uses no vertex more than once: for vertices in $\mathcal{V}(f)$, this holds by definition, for vertices outside of $\mathcal{V}(f)$, this follows from the fact that $P(f^-)$ and $P(f^+)$ are self-avoiding. Hence p_f is self-avoiding. □

The following lemma shows that C/f as defined above is indeed a bounded consistent configuration on $\mathcal{T}/f$.

Lemma 6.2.4. *Let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree decomposition of the simple, locally finite, connected, rooted graph (G, o), let $C \in \mathcal{C}_\mathcal{T}$, and let $f \in E(T)$. Then $C/f \in \mathcal{C}_{\mathcal{T}/f}$ and $\|C/f\| = \|C\|$.*

Proof. We start by showing that $C/f = (P/f, X/f)$ is a configuration on $\mathcal{T}/f$. First consider $t \neq t_f$. If $C(t)$ is empty, then $P/f(t) = P(t) = \emptyset$ and $X(t) = t^\uparrow$, so by the above observation $X/f(t) = t^{\uparrow/f}$. This shows that $C/f(t)$ is the empty configuration. If $C(t)$ is non-empty, then $P(t) = P/f(t)$ is a non-empty self-avoiding walk on $\mathcal{G}(t) = \mathcal{G}/f(t)$ starting in $\mathcal{V}(t^\uparrow, t) = \mathcal{V}/f(t^{\uparrow/f}, t)$, or in o, if $t = r$, and ending in $\mathcal{V}(t, X(t)) = \mathcal{V}/f(t, X/f(t))$. In case $X/f(t) = t$, clearly also $X(t) = t$, so in this case $P(t) = P/f(t)$ ends in a non-virtual edge. We conclude that $C/f(t) = (P/f(t), X/f(t))$ is a configuration on t.

Now consider the contracted vertex t_f. As before, without loss of generality assume that $f^- = (f^+)^\uparrow$. If $P/f(t_f) = p_f$ is the empty walk, then $P(f^-)$ is the empty walk and thus $C(f^-)$ must be the empty configuration. In particular $X(f^-) = (f^-)^\uparrow \notin \{f^-, f^+\}$, and thus $X/f(t_f) = X(f^-) = (t_f)^{\uparrow/f}$, showing that $C/f(t_f)$ is the empty configuration.

So we may assume that p_f is non-empty. By Lemma 6.2.3, p_f is a self-avoiding walk on $\mathcal{G}/f(t_f)$; it only remains to show that the first and last vertex of p_f lie in the appropriate adhesion sets. The first vertex of p_f is the same as the first vertex of $P(f^-)$, consequently it lies in $\mathcal{V}((f^-)^\uparrow, f^-) = \mathcal{V}/f((t_f)^{\uparrow/f}, t_f)$, or it is equal to o if $t_f = r/f$ and thus $f^- = r$. The last vertex of p_f is the last vertex of $P(t_l)$. If $X(f^+) \notin \{f^-, f^+\}$, then $X(f^-) = f^+$, and thus $t_l = f^+$. It follows that the last vertex of p_f lies in $\mathcal{V}(f^+, X(f^+)) = \mathcal{V}/f(t_f, X/f(t_f))$. If $X(f^-) \notin \{f^-, f^+\}$ an analogous argument applies. If both $X(f^-)$ and $X(f^+)$ are contained in $\{f^-, f^+\}$, then $X/f(t_f) = t_f$; in this case $P(t_l)$ ends with a non-virtual edge, and consequently p_f does not end in a virtual edge if $X/f(t_f) = t_f$. We conclude that $C/f(t_f) = (P/f(t_f), X/f(t_f))$ is a configuration on t_f.

By construction the number of non-boring parts with respect to C/f is at most the number of non-boring parts with respect to C, so C/f is bounded. Moreover C and C/f use the same non-virtual edges, so $\|C/f\| = \|C\|$ holds.

It remains to show that C/f is consistent, or in other words, that $C/f(s)$ and $C/f(t)$ are compatible for any edge $st \in E(T/f)$. If st is not incident with f, then this follows from the fact that C is consistent, so we may without loss of generality assume that

$t = t_f$. We only treat the case where s is a neighbour of f^- in T, the case where s and f^+ are neighbours is completely analogous.

Note that $\mathcal{V}/f(s,t_f) = \mathcal{V}(s,f^-) \subseteq V(\mathcal{G}(f^-))$. Thus Lemma 6.2.3 implies that $p_f \cap \mathcal{V}/f(s,t_f) = P(f^-) \cap \mathcal{V}(s,f^-)$ which in turn implies (C1). Next, note that $\mathcal{E}/f(s,t_f) = \mathcal{E}(s,f^-) \subseteq E(\mathcal{G}(f^-))$. By Lemma 6.2.3 we thus have $up_fv \cap \mathcal{E}/f(s,t_f) = uP(f^-)v \cap \mathcal{E}(s,f^-)$ for any pair of vertices u, v in $\mathcal{V}/f(s,t_f)$, and (C2) follows. For condition (C3) observe that

$$X/f(t_f) = s \iff X(f^-) = s \iff X(s) \neq f^- \iff X/f(s) \neq t_f.$$

Finally, note that if $X/f(t_f) = s$, then $X(f^+) = f^-$, and consequently p_f ends in the same vertex as $P(f^-)$, so (C4) is satisfied. □

Lemma 6.2.5. *Let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree decomposition of the rooted graph (G, o) and $f \in E(T)$. Then the function $C \mapsto C/f$ bijectively maps $\mathcal{C}_\mathcal{T}$ to $\mathcal{C}_{\mathcal{T}/f}$.*

Proof. As before, denote by t_f the contracted vertex in T/f. Let $C' = (P', X') \in \mathcal{C}_{\mathcal{T}/f}$. We show that for any $t \in V(T)$, there is a unique choice $C(t) = (P(t), X(t))$ such that C is a consistent configuration and $C/f = C'$.

First consider $t \notin \{f^-, f^+\}$. Necessarily $P(t) = P'(t)$, otherwise $P/f(t) \neq P'(t)$ by the definition of contraction. Similarly, if $X'(t) \notin \{f^-, f^+\}$, then $X(t) = X'(t)$ as otherwise $X/f(t) \neq X'(t)$. If $X'(t) = t_f$, then $X(t)$ must be either f^- or f^+. Moreover, for C to be a configuration, $X(t)$ must be adjacent to t, and since T is a tree, t cannot be adjacent to both f^- and f^+. So we have shown that $X(t)$ must be the unique neighbour of t in $\{f^-, f^+\}$.

By Lemma 6.2.3, we know that if we want $C/f = C'$, we have to make sure that $P'(t_f) \cap \mathcal{G}(f^-) = P(f^-) - \mathcal{E}(f)$. So $P(f^-)$ can only differ from the multi-walk $P'(t_f) \cap \mathcal{G}(f^-)$ by edges in $\mathcal{E}(f)$. Let $q_1, \ldots, q_l$ be the walk components of $P'(t_f) \cap \mathcal{G}(f^-)$. Note that each q_j for $j > 1$ starts in $\mathcal{V}(f)$, and each q_j for $j < l$ ends in $\mathcal{V}(f)$. In particular, it is possible to define a walk $P(f^-) = q_1e_1q_2 \ldots e_{l-1}q_l$, where $e_j \in \mathcal{E}(f)$ is a virtual edge connecting the last vertex of q_j to the first vertex of q_{j+1}. By the above discussion, this is the only choice of $P(f^-)$ for which $P/f(t_f) = P'(t_f)$ can possibly hold. A completely analogous argument applies to $P(f^+)$.

Finally, let us consider the exit directions of f^- and f^+. If $X'(t_f) \neq t_f$, then by (C2), there is a unique neighbour x of t_f in T/f such that $X'(x) \neq t_f$ and thus $X(x) \notin \{f^-, f^+\}$. If x is a neighbour of f^- in T, then necessarily $X(f^-) = x$ and $X(f^+) = f^-$, otherwise C is not consistent. Similarly, if x is a neighbour of f^+ in T, then necessarily $X(f^+) = x$ and $X(f^-) = f^+$. If $X'(t_f) = t_f$, then $X(f^-) = X(f^+) \in \{f^-, f^+\}$, since otherwise either $C/f \neq C'$, or C is not consistent. Note that in this case $P'(t_f)$ ends in a non-virtual edge e because C' is a configuration. If $e \in \mathcal{E}(f^-)$, then $X(f^+) = X(f^-) = f^-$, otherwise C is either not a configuration (if both endpoints of e lie in $\mathcal{V}(f)$), or it is inconsistent due to (C4). If $e \in \mathcal{E}(f^+)$, then analogously $X(f^+) = X(f^-) = f^+$.

A straight forward check (left to the reader) shows that the above construction indeed gives a bounded consistent configuration $C = (P, T) \in \mathcal{C}_\mathcal{T}$ with $C/f = C'$. □

Our next goal is to define contraction of finite sets of edges. For this purpose, let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree-decomposition of a simple, locally finite, connected, rooted graph (G, o) and let $F = \{f_1, \dots, f_k\}$ be a finite subset of $E(T)$. Then we define

$$\mathcal{T}/F = \mathcal{T}/f_1/f_2/\dots/f_k.$$

We note once again that this definition is consistent with the definition of $\mathcal{T}/F$ given in Section 6.1. If the set F induces a connected subgraph of T, then there is a unique contracted vertex in T/F; we denote it by t_F. Analogously, for a configuration C on $\mathcal{T}$, we define $C/F = (P/F, X/F)$ by

$$C/F = C/f_1/f_2/\dots/f_k.$$

We would like these definitions to be independent of the order in which the edge contractions are carried out. In order to make sense of this statement, we first need to clarify when we consider two tree decompositions and configurations on them to be the same. Let $\mathcal{T}_1 = (T_1, \mathcal{V}_1, r_1)$ and $\mathcal{T}_2 = (T_2, \mathcal{V}_2, r_2)$ be rooted tree decompositions of the same rooted graph (G, o). We say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are *isomorphic* if there is an isomorphism $\iota\colon T_1 \to T_2$ such that $\iota(r_1) = r_2$ and $\mathcal{V}_1 = \mathcal{V}_2 \circ \iota$. We call two configurations $C_1 = (P_1, X_1)$ on $\mathcal{T}_1$ and $C_2 = (P_2, X_2)$ on $\mathcal{T}_2$ isomorphic, if there is an isomorphism ι as above additionally satisfying $P_1 = P_2 \circ \iota$ and $\iota \circ X_1 = X_2 \circ \iota$. Since we only care about tree decompositions and configurations up to isomorphism, we write $\mathcal{T}_1 = \mathcal{T}_2$ and $C_1 = C_2$ to denote the fact that the respective tree decompositions and configurations are isomorphic. Inductive application of the following lemma shows that $\mathcal{T}/F$ and C/F (up to isomorphism) indeed do not depend on the order in which edges are contracted.

Lemma 6.2.6. *Let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree-decomposition of a simple, locally finite, connected, rooted graph (G, o), let $C = (P, X)$ be a configuration on $\mathcal{T}$ and let $F = \{f_1, f_2\} \subseteq E(T)$. Then $\mathcal{T}/f_1/f_2 = \mathcal{T}/f_2/f_1$ and $C/f_1/f_2 = C/f_2/f_1$.*

Proof. Before we get started, we need to discuss a notational issue. Recall that when we defined contractions, we abused notation so that we could refer to vertices and edges of T and T/f by the same names. When considering contractions of different edges, this is a potential source of confusion. For example, if there is an edge e connecting f_1^- to f_2^-, then e refers to the edge connecting t_{f_1} to f_2^- in T/f_1, as well as to the edge connecting t_{f_2} to f_1^- in T/f_2.

However, for double contractions as considered in this lemma, this abuse of notation works in our favour, that is, the function mapping every vertex of $T/f_1/f_2$ to the vertex of $T/f_2/f_1$ with the same name is an isomorphism (which will play the role of ι). More precisely, there are two cases to consider: if f_1 and f_2 are not incident, then $T/f_1/f_2$ and $T/f_2/f_1$ both contain two contracted vertices denoted by t_{f_1} and t_{f_2}. In this case, any edge incident to f_1^- or f_1^+ in T is incident to t_{f_1} in $T/f_1/f_2$ and $T/f_2/f_1$ and any edge incident to f_2^- or f_2^+ in T is incident to t_{f_2} in $T/f_1/f_2$ and $T/f_2/f_1$. If f_1 and f_2 are incident, then $T/f_1/f_2$ and $T/f_2/f_1$ both contain a unique contracted vertex which we will denote by t_F. In this case, any edge incident to f_1^-, f_1^+, f_2^-, or f_2^+ in T is incident to t_F in $T/f_1/f_2$ and $T/f_2/f_1$. The endpoints of all other edges are the same in T, $T/f_1/f_2$, and $T/f_2/f_1$ thus giving the desired isomorphism. In light of the above discussion, we will from now on treat $T/f_1/f_2$ and $T/f_2/f_1$ as the same tree and denote it by T/F.

The claim $\mathcal{T}/f_1/f_2 = \mathcal{T}/f_2/f_1$ follows directly from the definition of contraction. First note that by definition, if $t \notin \{f_1^-, f_1^+, f_2^-, f_2^+\}$, then $\mathcal{V}/f_1/f_2(t) = \mathcal{V}(t) = \mathcal{V}/f_2/f_1(t)$. If f_1 and f_2 are not incident, then

$$\mathcal{V}/f_1/f_2(t_{f_1}) = \mathcal{V}/f_1(t_{f_1}) = \mathcal{V}(f_1^-) \cup \mathcal{V}(f_1^+) = \mathcal{V}/f_2(f_1^-) \cup \mathcal{V}/f_2(f_1^+) = \mathcal{V}/f_2/f_1(t_{f_1}),$$

and analogous arguments show that $\mathcal{V}/f_1/f_2(t_{f_2}) = \mathcal{V}/f_2/f_1(t_{f_2})$. In case f_1 and f_2 are incident, the same line of reasoning leads to

$$\mathcal{V}/f_1/f_2(t_F) = \mathcal{V}/f_2/f_1(t_F) = \mathcal{V}(f_1^-) \cup \mathcal{V}(f_1^+) \cup \mathcal{V}(f_2^-) \cup \mathcal{V}(f_2^+),$$

where two of the sets in the union on the right-hand side are the same. It is also easily verified that $r/f_1/f_2 = r/f_2/f_1$, thus showing that indeed $\mathcal{T}/f_1/f_2 = \mathcal{T}/f_2/f_1$.

Our next goal is to show that $P/f_1/f_2 = P/f_2/f_1$. If $t \notin \{f_1^-, f_1^+, f_2^-, f_2^+\}$, then $P/f_1/f_2(t) = P(t) = P/f_2/f_1(t)$ by definition, so it only remains to consider the contracted vertices.

If f_1 and f_2 are not incident, then $P/f_1/f_2(t_{f_1}) = P/f_1(t_{f_1})$ and Lemma 6.2.3 tells us that this walk contains exactly the edges of $P(f_1^-) - \mathcal{E}(f_1)$ and $P(f_1^+) - \mathcal{E}(f_1)$. On the other hand, $P/f_2/f_1(t_{f_1})$ contains exactly the edges of $P/f_2(f_1^-) - \mathcal{E}/f_2(f_1)$ and $P/f_2(f_1^+) - \mathcal{E}/f_2(f_1)$. Since $P/f_2(f_1^-) = P(f_1^-)$, $P/f_2(f_1^+) = P(f_1^+)$, and $\mathcal{E}/f_2(f_1) = \mathcal{E}(f_1)$, the two edge sets coincide, and using that a self avoiding walk is uniquely determined by its set of edges we conclude that $P/f_1/f_2(t_{f_1}) = P/f_2/f_1(t_{f_1})$. An analogous argument shows that $P/f_1/f_2(t_{f_2}) = P/f_2/f_1(t_{f_2})$.

If f_1 and f_2 are incident, then we may assume without loss of generality that $f_1^- = f_2^-$, in particular the edge f_2 connects f_2^+ to t_{f_1} in T/f_1. By Lemma 6.2.3, the edge set of $P/f_1/f_2(t_F)$ consists of the edges of $P/f_1(f_2^+) - \mathcal{E}/f_1(f_2) = P(f_2^+) - \mathcal{E}(f_2)$ and $P/f_1(t_{f_1}) - \mathcal{E}/f_1(f_2)$. Again by Lemma 6.2.3, the edge set of the latter multi-walk consists of the edge sets of $P(f_1^-) - (\mathcal{E}(f_1) \cup \mathcal{E}(f_2))$ and $P(f_1^+) - (\mathcal{E}(f_1) \cup \mathcal{E}(f_2))$. Since f_1 is not incident to f_2^+, the graph $\mathcal{G}(f_2^+)$ and thus also the walk $P(f_2^+)$ is disjoint from $\mathcal{E}(f_1)$, and we conclude that the edge set of the walk $P/f_1/f_2(t_F)$ consists of the edge sets of $P(f_1^-) - (\mathcal{E}(f_1) \cup \mathcal{E}(f_2))$, $P(f_1^+) - (\mathcal{E}(f_1) \cup \mathcal{E}(f_2))$, and $P(f_2^+) - (\mathcal{E}(f_1) \cup \mathcal{E}(f_2))$. Since $f_1^- = f_2^-$, this is symmetric in f_1 and f_2, and an analogous argument shows that the edge set of $P/f_2/f_1(t_F)$ is the same. Thus the two walks coincide.

Finally, we need to show that $X/f_1/f_2 = X/f_2/f_1$. By Lemma 6.2.4, both $C/f_1/f_2$ and $C/f_2/f_1$ are bounded consistent configurations, thus by Remark 6.2.1 it suffices to show that the unique vertex $t \in T/F$ with $X/f_1/f_2(t) = t$ also satisfies $X/f_2/f_1(t) = t$. This clearly follows from the definition of X/f. □

Recall that the goal of this section is relating bounded consistent configurations on $\mathcal{T}$ to self-avoiding walks of length at least 1 starting at the root o of G. In this sense the upcoming Theorem 6.2.7 is the main result of this section. In preparation of this theorem, for each vertex t, we define a map ψ_t mapping bounded consistent configurations C on $\mathcal{T}$ to SAWs on the graph $\mathcal{G}(K_t)$ corresponding to the cone K_t as follows.

First, recall the definition of $\mathcal{G}(S)$ for S subset of $V(T)$. In particular, when $S = K_s$ is a cone,

$$\mathcal{G}(K_s) = \left(\bigcup_{t \in K_s} \mathcal{V}(t) \; , \; \bigcup_{t \in K_s} \mathcal{E}(t) \uplus \mathcal{E}(s, s^{\uparrow}) \right).$$

Let $S \subseteq K_t$ consist of the vertex t and all vertices of K_t carrying non-boring configurations. Note that Lemma 6.2.2 implies that $T[S]$ is connected and thus a finite subtree of K_t. Let $F = E(T[S])$ be the set of its edges. We define

$$\psi_t(C) = P/F(t_F),$$

where t_F is the unique contracted vertex in T/F. In other words, $\psi_t(C)$ is the self-avoiding walk on the finite graph $\mathcal{G}/F(t_F)$ obtained by contracting all edges of $T[K_t]$ connecting two vertices carrying non-boring configurations. By (C2) all its virtual edges must be in $\mathcal{E}/F(t_F^{\uparrow}, t_F) = \mathcal{E}(t^{\uparrow}, t)$, because all other neighbours of t_F carry boring configurations. In particular, $\psi_t(C)$ is a SAW on $\mathcal{G}(K_t)$ as claimed.

Let us illustrate this definition using the bounded consistent configuration depicted in Figure 6.4. In Figure 6.6 we iteratively contract edges incident to the root vertex r until only a single vertex carrying a non-boring configuration remains. This only takes two steps. Any further contraction, for example the one done in the third step, does not change the walk $\psi_r(C)$ anymore.

Theorem 6.2.7. *Let (G, o) be a simple, locally finite, connected graph rooted at $o \in V(G)$, and let $\mathcal{T} = (T, \mathcal{V}, r)$ be a rooted tree decomposition of (G, o). Then ψ_r is a bijection between the set $\mathcal{C}_{\mathcal{T}}$ and the set of self-avoiding walks of length at least 1 on G starting at o and for every $C \in \mathcal{C}_{\mathcal{T}}$, the weight $\|C\|$ coincides with the length of $\psi_r(C)$.*

Proof. Let $C \in \mathcal{C}_{\mathcal{T}}$. As above, let S be the set of all vertices of T carrying non-boring configurations and let $F = E(T[S])$. Note that the root r is contained in S, and consequently t_F is the root of T/F. By the above discussion, $\psi_r(C) = P/F(t_F)$ is a self-avoiding walk on G starting at the vertex o. Furthermore, by inductive application of Lemma 6.2.4, the weight $\|C\|$ is equal to the weight $\|C/F\|$, which is the length of the walk $\psi_r(C)$, because $\psi_r(C)$ contains no virtual edges. Finally, note that $X/F(t_F) = t_F$ and thus $P/F(t_F)$ ends with a non-virtual edge, in particular $\psi_r(C)$ has length at least 1.

It remains to show that ψ_r is bijective. We first show that it is injective. For $i = 1, 2$ let $C_i = (P_i, X_i) \in \mathcal{C}_{\mathcal{T}}$ such that $\psi_r(C_1) = \psi_r(C_2)$. Let S consist of all vertices s of T such that at least one of $C_1(s)$ and $C_2(s)$ is non-boring and let $F = E(T[S])$. Then $X_1/F(t_F) = X_2/F(t_F) = t_F$ because all neighbours of t_F carry boring configurations. While S could potentially contain some vertices s of T such that $C_i(s)$ is boring, this does not influence the result of $P_i/F(t_F)$. Thus by assumption

$$P_1/F(t_F) = \psi_r(C_1) = \psi_r(C_2) = P_2/F(t_F)$$

and this walk does not contain virtual edges, so Lemma 6.2.2 implies that $C_1/F = C_2/F$. Inductive application of Lemma 6.2.5 yields $C_1 = C_2$, so ψ_r is injective.

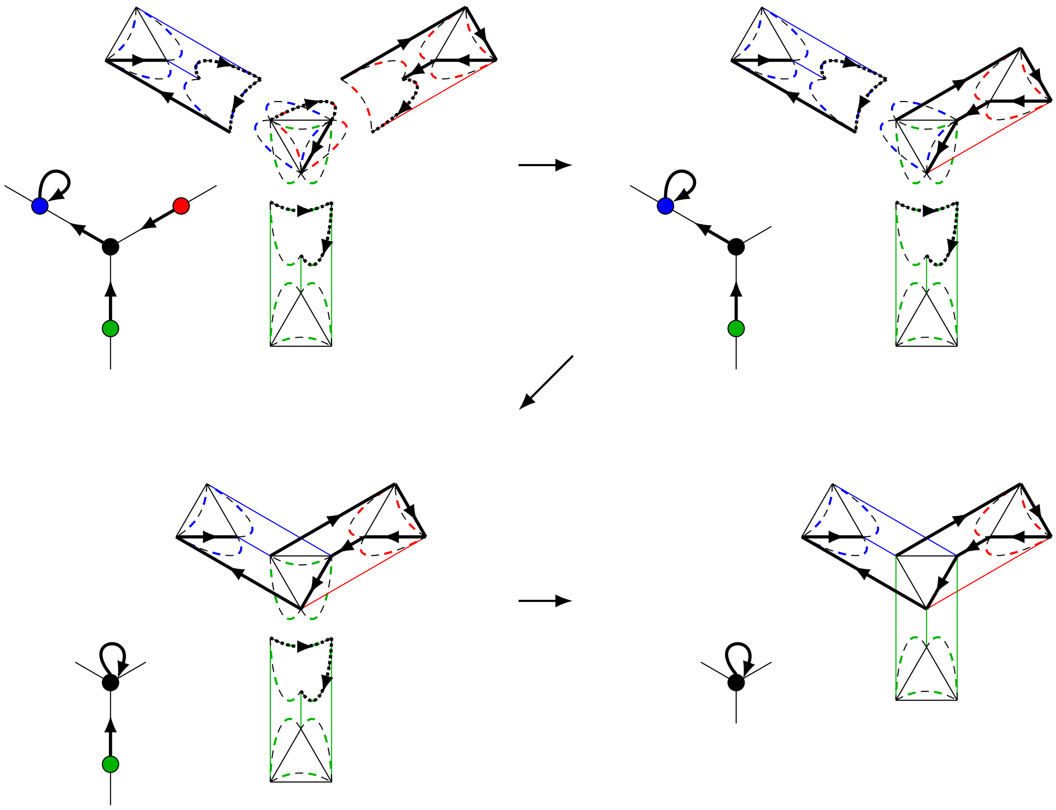


Figure 6.6: Contraction of the bounded consistent configuration C on T. The walk $\psi_r(t)$ is reached after 2 steps. It contains only non-virtual edges and does not change anymore during the third contraction.

To prove that ψ_r is surjective, let p be a SAW of length at least 1 on G starting at o. There is a finite subset $S \subseteq V(T)$ such that all edges of p are contained in $\mathcal{G}(S)$ and $T[S]$ is connected. As before, let $F = E(T[S])$. Then (p, t_F) is a configuration on t_F and p does not contain virtual edges, thus Lemma 6.2.2 provides a bounded consistent configuration $C \in \mathcal{C}_{\mathcal{T}/F}$ such that $C(t_F) = (p, t_F)$. Lemma 6.2.5 yields a configuration $C \in \mathcal{C}_{\mathcal{T}}$ with $\psi_r(C) = P$ and therefore ψ_r is surjective. $\square$

Remark 6.2.8. While ψ_t maps bounded consistent configurations on the cone K_t to walks on the corresponding graph $\mathcal{G}(K_t)$, it is (in general) only a bijection in the case $t = r$. This is due to the fact that for $t \neq r$ the exit direction $X(t)$ at vertex t is in general not uniquely determined by the walk $\psi_t(C)$; in many cases it can be either t or $t^{\uparrow}$.

6.3 A grammar for bounded consistent configurations

Throughout this section let (G, o) be a rooted, simple, locally finite, connected graph having only thin ends and let Γ be a group acting quasi-transitively on G. Corollary 5.5 in [56] by Thomassen and Woess states that any transitive graph without thick ends is accessible. Using the fact that any quasi-transitive graph is quasi-isometric to some transitive graph (see Remark 2.1.1), and additionally that quasi-isometries preserve accessibility (see also Lemma 4.1.1), we obtain that the graph G is accessible.

By Corollary 6.1.2 there is a reduced, Γ-invariant, rooted tree decomposition $\mathcal{T} = (T, \mathcal{V}, r)$ of (G, o) such that there are only finitely many Γ-orbits on $E(T)$. By property (P3) in Section 6.1 the parts of such a tree decomposition are finite and by (P4) the tree T is locally finite.

Our goal in this section is to reveal a recursive structure in the set of bounded consistent configurations, which we later use to define a context-free grammar. To this end we first show that there are only finitely many essentially different configurations on vertices of T. The letters in $\mathbf{\Sigma}$ will correspond to these different configurations, and the production rules will reflect the ways in which individual configurations can be combined in a compatible way.

6.3.1 Choosing representatives of configurations

We would like to define a function ρ assigning one of finitely many representatives to each vertex t of T and each configuration $c = (p, x)$ on t. This function ρ is chosen in a way that for neighbouring vertices s and t of T and configurations c_s and c_t on them compatibility only depends on $\rho(c_s)$ and $\rho(c_t)$.

We start by choosing representatives of the cone equivalence classes of vertices of T.

Lemma 6.3.1. *There is a finite subset R of $V(T)$ containing exactly one vertex of every cone type such that $T[R]$ is connected.*

Proof. Choose a subset R of $V(T)$ such that R contains at least one vertex of every cone type, $T[R]$ is connected, and R has minimal cardinality among all such sets. Clearly R is finite as there are only finitely many cone types. Assume that R contains two vertices s and t in the same cone type and let $\gamma \in \Gamma$ map K_s onto K_t. Then the set $R' = (R \cup \gamma(R \cap K_s)) \setminus (R \cap K_s)$ still satisfies the condition that $T[R']$ is connected and has smaller cardinality than R, because $t \in R \cap \gamma(R \cap K_s)$. $\square$

Let us now fix a set of representatives R of the cone types of $\mathcal{T}$ as in the previous lemma. The representative of a vertex t of T is denoted by $\rho(t) \in R$.

For every vertex t of T we define an automorphism $\delta_t \in \Gamma$ mapping K_t to $K_{\rho(t)}$. For $t \in R$ let $\delta_t = 1_\Gamma$, the neutral element in Γ, which acts as the identity on G. For any vertex $t \in V(T) \setminus R$ with $t^\uparrow \in R$ fix some arbitrary automorphism $\delta_t \in \Gamma$ mapping K_t to $K_{\rho(t)}$. Finally for all other vertices t of T inductively define

$$\delta_t = \delta_{\delta_{t^\uparrow}(t)} \circ \delta_{t^\uparrow}. \tag{6.1}$$

This is well defined because $\delta_{t^\uparrow}(t)$ must be a child of a vertex of R for every t; in particular, if $\delta_{t^\uparrow}(t) \in R$, then $\delta_{\delta_{t^\uparrow}(t)} = 1_\Gamma$. Note that equation (6.1) in fact holds for all vertices t of T besides the root r.

We use these maps to extend the map ρ to configurations. Let $c = (p, x)$ be a configuration on a vertex t of T. Then ρ maps c onto a configuration on the representative $\rho(t) = \delta_t(t)$ of t by

$$\rho(c) = \delta_t(c) = (\delta_t(p), \delta_t(x)).$$

Additionally for every $t \in T$ let $t_1^\downarrow, \ldots, t_{k(t)}^\downarrow$ be the $k(t)$ children of t ordered in a way such that

$$\delta_t(t_i^\downarrow) = \delta_t(t)_i^\downarrow = \rho(t)_i^\downarrow. \tag{6.2}$$

This can be achieved by fixing any order of the children of vertices in R and then ordering the children of any vertex t accordingly. From the definition (6.1) of δ_t it is immediate that

$$\delta_{t_i^\downarrow} = \delta_{\delta_t(t_i^\downarrow)} \circ \delta_t. \tag{6.3}$$

Note that the representative of the i-th child of t is exactly the representative of the i-th child of $\rho(t)$:

$$\rho(t_i^\downarrow) = \delta_{t_i^\downarrow}(t_i^\downarrow) = \delta_{\delta_t(t_i^\downarrow)}(\delta_t(t_i^\downarrow)) = \delta_{\rho(t)_i^\downarrow}(\rho(t)_i^\downarrow) = \rho(\rho(t)_i^\downarrow). \tag{6.4}$$

In the above equation, the first and last equalities use the definition of ρ on $t_i^\downarrow$ and $\rho(t)_i^\downarrow$ respectively, the other equalities follow from (6.3) and (6.2).

Moreover we can use δ_t to map a consistent configuration C on t and its children to a consistent configuration $\delta_t \circ C \circ \delta_t^{-1}$ on the representative $\rho(t)$ and its children. Note that δ_t^{-1} is applied to vertices of T whereas δ_t is applied to configurations on the corresponding parts. Similar to (6.4), we get that the configuration assigned by C to the i-th child of t and the configuration assigned by $\delta_t \circ C \circ \delta_t^{-1}$ to the i-th child of the representative of t have the same image under ρ:

$$\rho(C(t_i^\downarrow)) = \delta_{t_i^\downarrow}(C(t_i^\downarrow)) = \delta_{\delta_t(t_i^\downarrow)}(\delta_t(C(t_i^\downarrow))) = \rho(\delta_t(C(t_i^\downarrow))) = \rho(\delta_t \circ C \circ \delta_t^{-1}(\rho(t)_i^\downarrow)). \tag{6.5}$$

Similarly to (6.4), the first and second equalities in (6.5) follow from the definition of ρ and equation (6.3), respectively. For the third equality, we use the definition of ρ and the fact that $\delta_t(C(t_i^\downarrow))$ is a configuration on $\delta_t(t_i^\downarrow)$. The last equality follows from (6.2).

Let us illustrate the above definitions using the tree decomposition $(T, \mathcal{V})$ shown in Figure 6.2. As before, we choose the central vertex corresponding to a part of cardinality 3 as the root r. It is not hard to see that Γ acts with six orbits on the set of directed edges of T, so we obtain seven cone types and the set R has to contain seven vertices (see also Lemma 6.1.7). Figure 6.7 shows a valid choice of the set R. Note that we could also have taken the seven vertices in the first three layers of the rooted tree T as our set R.

Let us sketch the recursive definition of δ_t using the vertices t_1, t_2 and t_3 in Figure 6.7. First, note that the vertex t_1 is contained in R, so by definition δ_{t_1} is the identity map. The parent of t_2 is in R, so we may choose an arbitrary automorphism γ in Γ mapping t_2

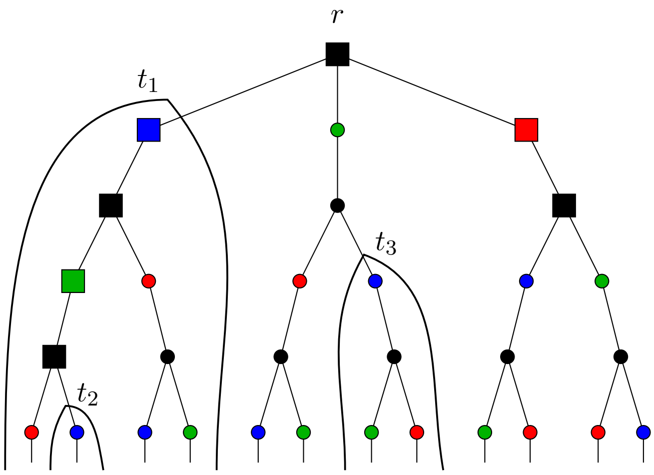


Figure 6.7: Cone types of the tree decomposition T. The square nodes give one possible valid choice for the set R. The marked cones rooted at t_1, t_2 and t_3 have the same type.

to its representative t_1 in R and set $\delta_{t_2} = \gamma$. Regarding t_3, note that the representative of $t_3^\uparrow$ in R is $t_2^\uparrow$, thus by definition $\delta_{t_3^\uparrow}$ maps the cone $K_{t_3^\uparrow}$ onto $K_{t_2^\uparrow}$ and thus t_3 onto t_2. The map δ_{t_3} is obtained by first applying $\delta_{t_3^\uparrow}$ (which maps t_3 to t_2), and then applying $\delta_{\delta_{t_3^\uparrow}} = \delta_{t_2}$ (which maps t_2 to t_1).

6.3.2 Construction of the grammar

The goal of this section is to construct a context-free language, whose words are in one-to-one correspondence with bounded consistent configurations on the rooted tree decomposition $\mathcal{T}$. Essentially, this language is obtained by going through the vertices of the decomposition tree carrying non-boring configurations in the order given by depth-first search and noting down the corresponding representatives of configurations (that is, their images under ρ).

Instead of constructing a context free grammar for this language, we construct a 1-multiple context free grammar. Clearly the two notions are equivalent; recall that by Remark 2.2.4 every context-free grammar stands in one-to-one correspondence to a 1-multiple context-free grammar generating the same language. However, in anticipation of Section 6.4, where a multiple context free grammar for SAWs is constructed in a very similar way, it makes more sense to consider multiple context-free grammars in the present section as well.

The 1-multiple context-free grammar $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, A_r)$ generating our desired language is defined as follows.

The alphabet is

$$\mathbf{\Sigma} = \{a_c \mid c \text{ configuration on some } t \in R\}.$$

The set of non-terminals is

$$\mathbf{N} = \{A_r\} \cup \{A_{t,c} \mid t \in R,\ c \text{ configuration on } t\}.$$

For each $t \in R$ and every consistent configuration C on the set $\{t, t_1^{\downarrow}, \dots, t_{k(t)}^{\downarrow}\}$, the set $\mathbf{P}$ contains a production rule as follows. If $C(t)$ is non-boring, then the production

$$A_{t,C(t)}(a_{C(t)} x_1 \dots x_{k(t)}) \leftarrow \Big(A_{\rho(t_i^{\downarrow}),\rho(C(t_i^{\downarrow}))}(x_i)\Big)_{i\in[k(t)]}$$

is in $\mathbf{P}$. If $C(t)$ is boring, then $\mathbf{P}$ contains the terminal rule

$$A_{t,C(t)}(\epsilon) \leftarrow .$$

Additionally $\mathbf{P}$ contains for every configuration $C(r)$ on r the production

$$A_r(x) \leftarrow A_{r,C(r)}(x).$$

Note that any production rule of this grammar is uniquely determined by its head and tail. This means that we do not lose any information by using simplified derivation trees, where every vertex is labelled by the head of its corresponding rule instead of the complete rule. To shorten notation, we henceforth work with simplified labels.

Remark 6.3.2. Observe that for any simplified derivation tree D of $\mathbf{G}$ whose root d is labelled $A_{t,c}$, the following three conditions hold.

(i) The number of children of d is uniquely given by the pair (t, c); we denote it by k.

(ii) If c is non-boring, the word corresponding to D is $w(D) = a_c w(D_1) \dots w(D_k)$, where D_i is the sub-tree of D rooted at the i-th child of d and $k > 0$. Otherwise c is boring, $w(D) = \epsilon$ and d is the only vertex of D.

(iii) Let $d = v_1, v_2, \dots, v_n$ be the vertices of D in DFS-order and let $\lambda(v_i) = A_{t_i,c_i}$ be their labels. Then

$$w(D) = x_1 \dots x_n, \quad \text{where} \quad x_i = \begin{cases} a_{c_i} \text{ if } c_i \text{ is non-boring} \\ \epsilon \text{ otherwise} \end{cases}$$

Observations (i) and (ii) are direct consequences of the structure of $\mathbf{G}$. For (iii) use induction on the number of vertices n of D. If $n = 1$, then (ii) implies that c_1 is boring and $w(D) = \epsilon$, so (iii) holds. Let now $n > 1$ and suppose (iii) holds for every derivation tree with at most $n-1$ vertices. Then c_1 is non-boring and the claim follows from (ii) by applying the induction hypothesis on the sub-trees $D_1, \dots, D_k$ rooted at the k children of v_1.

Lemma 6.3.3. *The grammar* $\mathbf{G}$ *is unambiguous 1-multiple context-free.*

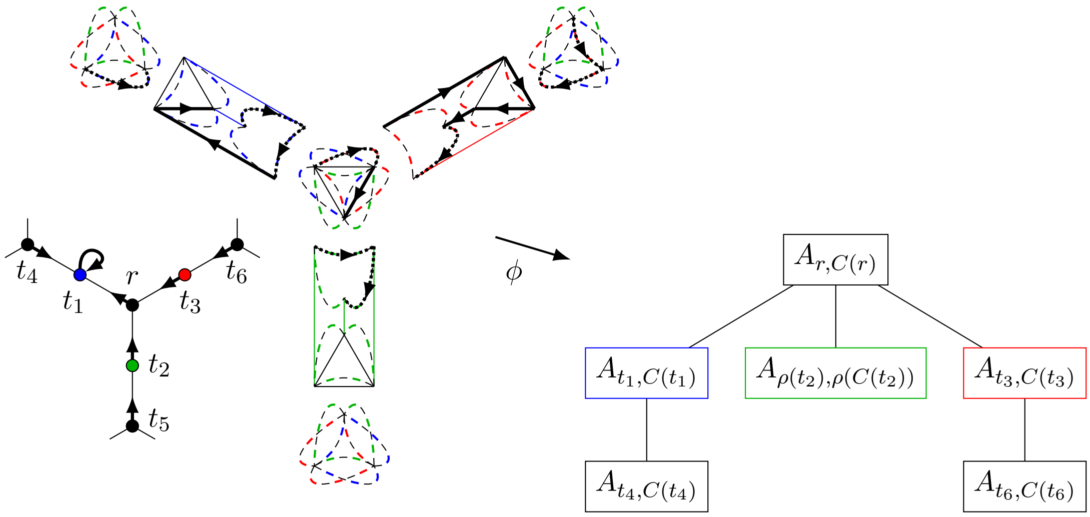


Figure 6.8: The map ϕ transforms the configuration C on the decomposition tree T into the derivation tree $\phi(C)$ over the grammar $\mathbf{G}$. The generated word is $w(\phi(C)) = a_{C(r)}a_{C(t_1)}a_{C(t_4)}a_{\rho(C(t_2))}a_{C(t_3)}a_{C(t_6)}$.

Proof. Let D and D' be different non-trivial (consisting of at least 2 vertices) derivation trees. Then there is a smallest positive number m, such that the m-th vertices u and u' in the DFS-orders on D and D' either have a different number of children or have a different label. Remark 6.3.2 (i) implies that in any case $\lambda(u) = A_{t,c} \neq \lambda(u') = A_{t',c'}$, so in particular $c \neq c'$. By minimality of m, the parents of u and u' have the same label, so by Lemma 6.2.2 the configurations c and c' must be non-boring. Thus Remark 6.3.2 (iii) implies that $w(D) \neq w(D')$ and $\mathbf{G}$ is unambiguous. □

Bounded consistent configurations on $\mathcal{T}$ are closely related to derivation trees of $\mathbf{G}$. Let us define a map ϕ assigning to any given $C = (P, X) \in \mathcal{C}_\mathcal{T}$ a derivation tree of our grammar $\mathbf{G}$ as follows. Let $S \subseteq V(T)$ consist of all vertices s carrying non-boring configurations $C(s)$ and their neighbours. Then $T[S]$ is connected and can be seen as an ordered tree with root r, where the order of the children is inherited from the tree T. We label every vertex s of $T[S]$ with $\lambda(s) = A_{\rho(s),\rho(C(s))}$.

As an example, we provide in Figure 6.8 the derivation tree $\phi(C)$ of the configuration C given in Figure 6.4. Note that the vertex t_2 is not contained in the set R from Figure 6.7, so we need to apply ρ to the respective configuration. For all other t_i, the map ρ is the identity map and can be omitted. The vertex t_5 is not contained in $\phi(C)$ because its parent t_2 carries a boring configuration.

Lemma 6.3.4. *The map ϕ is a bijection between the set $\mathcal{C}_\mathcal{T}$ of bounded consistent configurations on $\mathcal{T}$ and the set of derivation trees whose root is labelled by $A_{r,c}$ for some configuration c on r.*

Proof. Observe that an ordered tree labelled with non-terminals of $\mathbf{N}$ is a simplified derivation tree of $\mathbf{G}$ if and only if for every vertex t and its children $t_1, \ldots, t_k$ there is a rule in $\mathbf{P}$ with head $\lambda(t)$ and tail $(\lambda(t_1), \ldots, \lambda(t_k))$.

Let S be as above and s be a vertex of $T[S]$. If $C(s)$ is boring, then s is a leaf in $T[S]$ and $A_{\rho(s),\rho(C(s))}(\epsilon) \leftarrow$ is a rule in $\mathbf{P}$.

Otherwise $C(s)$ is non-boring and we consider the children $s_1^{\downarrow}, \ldots, s_{k(s)}^{\downarrow}$ of s in T, which are also the children of s in $T[S]$. Then the production

$$A_{\rho(s),\rho(C(s))}(a_{\rho(C(s))}x_1 \ldots x_{k(s)}) \leftarrow \left(A_{\rho(s_i^{\downarrow}),\rho(C(s_i^{\downarrow}))}(x_i)\right)_{i\in[k(s)]}$$

is in $\mathbf{P}$ because $\delta_s \circ C \circ \delta_s^{-1}$ is a consistent configuration on $\{\delta_s(s), \delta_s(s_i^{\downarrow}), \ldots, \delta_s(s_{k(t)}^{\downarrow})\}$ and (6.4) and (6.5) hold. We conclude that $\phi(C)$ is a derivation tree of $\mathbf{G}$.

Our next step is to show that ϕ is surjective. Let D be a derivation tree of $\mathbf{G}$ whose root d is labelled $A_{r,c}$. We recursively construct an embedding ι of D into T and a bounded consistent configuration C on T such that every vertex u of D has the label

$$\lambda(u) = A_{\rho(\iota(u)),\rho(C(\iota(u)))}. \tag{6.6}$$

We start the top-down construction by setting $\iota(d) = r$ and $C(r) = c$. Then clearly d satisfies (6.6). Suppose now $\iota(u) = t$ is already defined for a vertex u of D and that u satisfies (6.6). Let $u_1, \ldots, u_k$ be the $k > 0$ children of u in D and $\lambda(u_i) = A_{s_i,c_i}$ their labels. Then

$$A_{\rho(t),\rho(C(t))}(a_{\rho(C(t))}x_1 \ldots x_k) \leftarrow (A_{s_i,c_i}(x_i))_{i\in[k]} \tag{6.7}$$

is a rule in $\mathbf{P}$. This implies that t has precisely k children $t_1^{\downarrow}, \ldots, t_k^{\downarrow}$ in T and moreover that $\rho(t_i^{\downarrow}) = s_i$ holds for every i. We define $\iota(u_i) = t_i^{\downarrow}$ and $C(t_i^{\downarrow}) = \delta_{t_i}^{-1}(c_i)$. Then $C(t_i^{\downarrow})$ is compatible with $C(t)$ for every $i \in [k]$; the production rule (6.7) is in $\mathbf{P}$ if and only if $\rho(C(t)) = \delta_t(C(t))$ and $\delta_{\rho(t)_i^{\downarrow}}^{-1}(c_i) = \delta_t(\delta_{t_i}^{-1}(c_i))$ are compatible. In this way we have constructed a consistent configuration C on the sub-tree $\iota(D)$ of T. Note that by definition of the set $\mathbf{P}$, configurations on leaves of $\iota(D)$ are boring. By Lemma 6.2.2 the configuration C can be (uniquely) extended to a bounded consistent configuration on T. Moreover it follows directly from (6.6) that $\phi(C) = D$, so ϕ is surjective.

Finally, it is not hard to see that ϕ is injective. Any two different configurations $C_1 \neq C_2$ have to differ on some vertex t of T; we pick such a t with minimal distance to the root r. Then $C_1(t^{\uparrow}) = C_2(t^{\uparrow})$, so Lemma 6.2.2 yields that $C_1(t^{\uparrow})$ is non-boring. Thus t is a vertex of $\phi(C_1)$ and $\phi(C_2)$. For $i \in \{1, 2\}$ the label of t in $\phi(C_i)$ is $A_{\rho(t),\rho(C_i(t))}$, so in particular $\phi(C_1) \neq \phi(C_2)$ and we conclude that ϕ is injective. □

It is clear that ϕ also describes a bijection between the set $\mathcal{C}_{\mathcal{T}}$ of bounded consistent configurations on $\mathcal{T}$ and derivation trees of $\mathbf{G}$ whose roots are labelled A_r. Moreover, the number of occurrences of a given letter a_c in the word $w(\phi(C)) \in L(\mathbf{G})$ corresponding to $\phi(C)$ is equal to the number of vertices t of T with $\rho(C(t)) = c$.

Combining the previous results, the composition of the bijection ϕ mapping configurations onto derivation trees and the natural bijection w between derivation trees and their corresponding words of an unambiguous 1-multiple context-free grammar $\mathbf{G}$ is a bijection between $\mathcal{C}_{\mathcal{T}}$ and words in $L(\mathbf{G})$, as stated in the following corollary.

Corollary 6.3.5. *There is a bijection θ between the set $\mathcal{C}_\mathcal{T}$ of bounded consistent configurations on $\mathcal{T}$ and the words in the unambiguous 1-multiple context-free language $L(\mathbf{G})$ such that the number of occurrences of a given letter a_c in $\theta(C)$ coincides with the number of vertices t of T with $\rho(C(t)) = c$.*

Further combining Corollary 6.3.5 with the connection between bounded consistent configurations and SAWs established in Theorem 6.2.7, we obtain the proof of our first main result, Theorem 6.0.1. Before we turn to the proof, let us recall the statement of the theorem.

Theorem 6.0.1. *Let G be a locally finite, connected, quasi-transitive graph having only thin ends and let $o \in V(G)$. Then $F_{\mathrm{SAW},o}(z)$ is algebraic over $\mathbb{Q}$. In particular the connective constant $\mu(G)$ is an algebraic number.*

Proof. Let Γ be a group acting quasi-transitively on G and $\mathcal{T} = (T, \mathcal{V}, r)$ be a reduced, Γ-invariant, rooted tree decomposition of (G, o) such that there are only finitely many Γ-orbits on $E(T)$. Then by Theorem 6.2.7 the generating function of self-avoiding walks coincides with the generating function of the set $\mathcal{C}_\mathcal{T}$,

$$F_{\mathrm{SAW},o}(z) = \sum_{C \in \mathcal{C}_\mathcal{T}} z^{\|C\|}.$$

Let $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, S)$ be the unambiguous 1-multiple context-free grammar over the alphabet $\mathbf{\Sigma} = \{a_{c_1}, \dots, a_{c_m}\}$ as defined at the start of Section 6.3.2, where the c_i are configurations on vertices in R, and let $F_{L(\mathbf{G})}(a_{c_1}, \dots, a_{c_m})$ be the commutative language generating function of $L(\mathbf{G})$. Chomsky and Schützenberger showed in [7] that commutative language generating functions of unambiguous context-free languages are algebraic over $\mathbb{Q}$, that is, there is an irreducible polynomial P in $m+1$ variables with coefficients in $\mathbb{Q}$ such that

$$P(F_{L(\mathbf{G})}(a_{c_1}, \dots, a_{c_m}), a_{c_1}, \dots, a_{c_m}) = 0.$$

Corollary 6.3.5 yields that the generating function of $\mathcal{C}_\mathcal{T}$ coincides with the generating function obtained by substituting every variable a_{c_i} in $F_{L(\mathbf{G})}$ by the monomial $z^{\|c_i\|}$,

$$F_{\mathrm{SAW},o}(z) = F_{L(\mathbf{G})}\left(z^{\|c_1\|}, \dots, z^{\|c_m\|}\right)$$

In particular $F_{\mathrm{SAW},o}(z)$ is algebraic over $\mathbb{Q}$; it satisfies the equation

$$Q\left(F_{\mathrm{SAW},o}(z), z\right) = 0,$$

where $Q(y, z) = P\left(y, z^{\|c_1\|}, \dots, z^{\|c_m\|}\right)$ is a polynomial with coefficients in $\mathbb{Q}$. The connective constant $\mu(G)$ is the reciprocal of the radius of convergences of the algebraic function $F_{\mathrm{SAW},o}(z)$ and thus an algebraic number. □

6.4 The multiple context-free language of self-avoiding walks

In this section, we prove the second main result of this paper, which we briefly recall for convenience.

Theorem 6.0.3. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph and let $o \in V(G)$. Then $L_{\mathrm{SAW},o}(G)$ is k-multiple context-free if and only if every end of G has size at most $2k$.*

The proofs of the two implications are quite different from one another and will be discussed separately in the two subsections of this section. In order to show that bounded end size implies that $L_{\mathrm{SAW},o}(G)$ is a MCFL, we adopt a similar approach as in the previous section and construct a MCFG which is very closely related to the grammar $\mathbf{G}$ defined in Section 6.3. For the converse implication we essentially follow the approach of Section 4.2; note that this part does not depend on the other results of the current chapter and can be read independently.

6.4.1 Bounded end size implies multiple context-freeness

We use again the assumptions, notation and definition from the previous section. In particular $\mathcal{T} = (T, \mathcal{V}, r)$ again denotes a rooted tree decomposition of a rooted graph (G, o) fixed throughout this section; we also fix a map ρ as constructed in Section 6.3. Our aim is proving the following theorem, a stronger version of one of the two implications of Theorem 6.0.3.

Theorem 6.4.1. *Let G be a simple, locally finite, connected, quasi-transitive edge-labelled graph having only ends of size at most k and let o be a given vertex of G. Then the language of self-avoiding walks $L_{\mathrm{SAW},o}(G)$ is $\lceil k/2 \rceil$-multiple context-free.*

If the edge-labelling is deterministic, then $L_{\mathrm{SAW},o}(G)$ is unambiguous $\lceil k/2 \rceil$-multiple context-free.

To prove the above theorem, we give an MCFG $\mathbf{G} = (\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, A_r)$ and show that it generates the language $L_{\mathrm{SAW},o}(G)$. As mentioned above, $\mathbf{G}$ is a refinement of the 1-MCFG from Section 6.3.

Obviously, the alphabet $\mathbf{\Sigma}$ has to consist of all edge-labels. Note that this is a finite set since G is locally finite and the group of label preserving automorphisms is assumed to act quasi-transitively.

Like in Section 6.3, the set of non-terminals is

$$\mathbf{N} = \{A_r\} \cup \{A_{t,c} \mid t \in R,\ c \text{ configuration on } t\}.$$

However, since we are constructing a MCFG, we need to assign a rank to each non-terminal; to this end some additional definitions are necessary. For a vertex t of T and a configuration $c = (p, x)$ on t let $\mu(c)$ denote the number of walk components of $p - \mathcal{E}(t^{\uparrow}, t)$ containing at least two vertices of $\mathcal{V}(t^{\uparrow}, t)$. Furthermore let $r(c) = \mu(c) + 1$ if $x \neq t^{\uparrow}$ and the final walk component of $p - \mathcal{E}(t^{\uparrow}, t)$ contains only a single vertex of $\mathcal{V}(t^{\uparrow}, t)$ and let

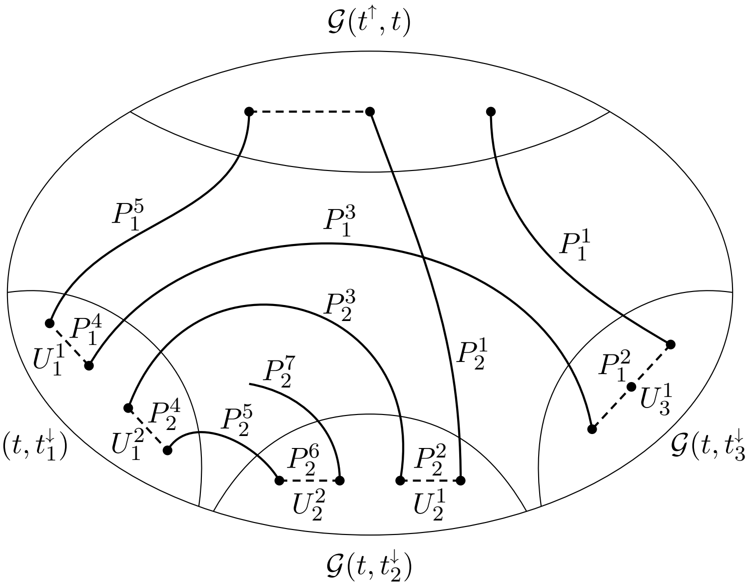


Figure 6.9: Decomposition of $P(t)$ into sub-walks P_h^l. For even l, the walk P_h^l is equal to some U_i^j; for instance $P_2^4 = U_1^2$ means that the fourth sub-walk in the second walk component of $P(t) - \mathcal{E}(t^{\uparrow}, t)$ coincides with the second walk component of $P(t) \cap \mathcal{G}(t, t_1^{\downarrow})$.

$r(c) = \mu(c)$ otherwise. Define the rank of the non-terminal $A_{t,c}$ to be $r(c)$. Note that $r(c) = 0$ if and only if c is boring. For a configuration $c = (p, x)$ on the root part, we define $r(c) = 1$; note that this is consistent with the above definition in the sense that the root has no parent and there is exactly one walk component of p.

Next we turn to the set $\mathbf{P}$ of production rules. For every boring configuration c on a vertex $t \in R$, $\mathbf{P}$ contains the rule

$$A_{t,c}(\emptyset) \leftarrow .$$

For non-boring configurations, we need more involved production rules that require some preliminary definitions. Let $t \in R$ and let $C = (P, X)$ be a consistent configuration on $\{t, t_1^{\downarrow}, \dots, t_{k(t)}^{\downarrow}\}$ such that $C(t)$ is non-boring. Let $P_1, \dots, P_{\mu(C(t))}$ be the walk components of $P(t) - \mathcal{E}(t^{\uparrow}, t)$ containing at least 2 vertices of $\mathcal{V}(t^{\uparrow}, t)$, and if $r(C(t)) > \mu(C(t))$ let $P_{r(C(t))}$ be the (possibly trivial) final walk component of $P(t) - \mathcal{E}(t^{\uparrow}, t)$. Moreover, for each $i \in [k(t)]$ let $U_i^1, \dots, U_i^{\mu_i}$ be the non-trivial walk components of $P(t) \cap \mathcal{G}(t, t_i^{\downarrow})$, that is, the walk components that contain more than one vertex. Every P_h admits a unique decomposition into an odd number of sub-walks $P_h = P_h^1 P_h^2 \dots P_h^{2m+1}$ such that P_h^l is a (possibly trivial) non-virtual walk if l is odd and equal to some U_i^j if l is even. Observe that $P_h^l = U_i^j$ means that the l-th part in the decomposition of P_h is the j-th walk component of $P(t) \cap \mathcal{G}(t, t_i^{\downarrow})$, that is, the notation P_h^l indicates which walk component of $P(t)$ the walk lies in, whereas the notation Q_i^j tells us which adhesion set the virtual edges belong to, see Figure 6.9 for an example.

For $h < r(C(t))$, the string α_h corresponding to P_h is

$$\alpha_h = \ell(P_h^1)z(P_h^2)\ell(P_h^3)z(P_h^4)\ldots\ell(P_h^{2m+1}),$$

where $z(P_h^l) = z_{i,j}$ with i, j chosen such that $P_h^l = U_i^j$; in other words, α_h is obtained from P_h by concatenating labels of non-virtual walks P_h^l, and variables $z_{i,j}$ for virtual walks P_h^l. For $h = r(C(t))$, we define

$$\alpha_h = \ell(P_h^1)z(P_h^2)\ell(P_h^3)z(P_h^4)\ldots\ell(P_h^{2m+1})\beta,$$

where $z(P_h^l) = z_{i,j}$ as above, and $\beta = \epsilon$ unless there is some $i \in [k(t)]$ such that $X(t) = t_i^{\downarrow}$ and $P(t)$ does not end with an edge in $\mathcal{E}(t, t_i^{\downarrow})$, in which case β is equal to the single variable z_{i,μ_i+1}.

For every $t \in R$ and each configuration C on t and its children such that $C(t)$ is non-boring, $\mathbf{P}$ includes the production rule

$$A_{t,C(t)}(\alpha_1, \ldots, \alpha_{r(C(t))}) \leftarrow \left(A_{\rho(t_i^{\downarrow}),\rho(C(t_i^{\downarrow}))}(z_{i,1}, \ldots, z_{i,r(C(t_i^{\downarrow}))})\right)_{i\in[k(t)]} \tag{6.8}$$

where the strings $\alpha_1, \ldots, \alpha_{r(C(t))}$ are defined as above.

Finally, for every configuration c on the root r the production rule

$$A_r(z) \leftarrow A_{r,c}(z)$$

is in $\mathbf{P}$; this is a well formed production rule since $A_{r,c}$ has rank 1.

Before showing that this grammar indeed generates the language $L_{\mathrm{SAW},o}(G)$, let us discuss why this intuitively should be true.

Let us extend the label function ℓ to walks p on the graph $\mathcal{G}(K_t)$ corresponding to the cone K_t in the natural way by mapping p onto the tuple of labels $\ell(p) = (\ell(p_1), \ldots, \ell(p_m))$ of the non-trivial walk components $p_1, \ldots, p_m$ of $p - \mathcal{E}(t^{\uparrow}, t)$. If p is the empty walk, then $\ell(p) = \emptyset$. For a bounded consistent configuration $C \in \mathcal{C}_{\mathcal{T}}$ with $C(t) = c$, we want the term $A_{t,c}(\ell(\psi_t(C)))$ to be derivable in $\mathbf{G}$, where $\psi_t(C)$ is the walk on $\mathcal{G}(K_t)$ corresponding to the configuration C as defined on page 85. This is inductively taken care of by production rules of type (6.8): sub-walks of P_h consisting of virtual edges correspond to variables $z_{i,j}$ in the string α_h which are subsequently replaced by strings corresponding to walk components of $\psi_s(C) - \mathcal{E}(t, s)$ where s is some child of t. To see that this intuitively makes sense, recall that by definition $\psi_t(C)$ is obtained by contracting all edges vertices carrying non-boring configurations and then taking the walk $P(t_F)$, where t_F is the unique contracted vertex. By Lemma 6.2.6, the order of edge contractions does not matter, hence we can first contract all edges not incident to t; in particular, the walk $\psi_t(C)$ can be obtained by replacing sub-walks consisting of virtual edges in $\mathcal{E}(t, s)$ by appropriate sub-walks of $\psi_s(C)$, see Figure 6.10. This replacement procedure is essentially captured by rules of type (6.8).

When making the above intuition precise, there are some technical issues that need to be addressed, leading to the fairly involved definition of $r(c)$ and to the subtle difference between α_h for $h < r(C(t))$ and $\alpha_{r(C(t))}$. These are due to the fact that walk components

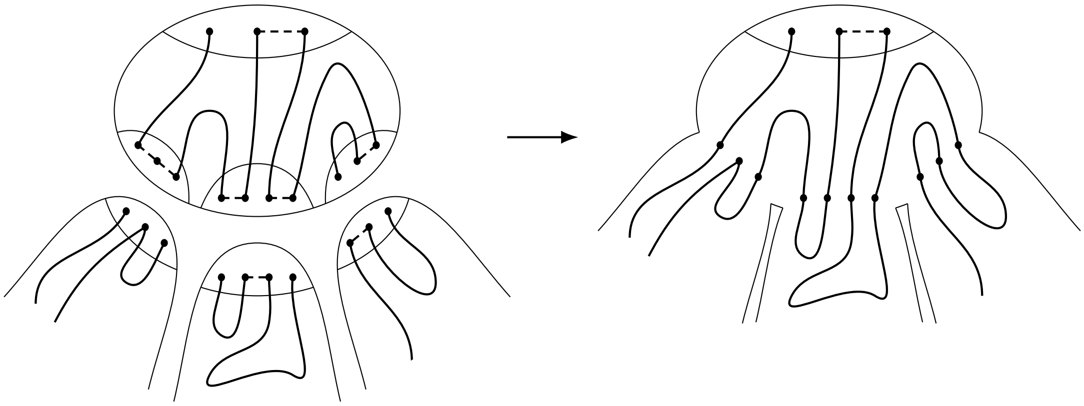


Figure 6.10: Iterated contraction shows that we can obtain $\psi_t(C)$ by combining walks $P(t)$ and $\psi_s(C)$ for all children s of t.

of $\psi_t(C) - \mathcal{E}(t^\uparrow, t)$ come in two different flavours. Recall that the walk $\psi_t(C)$ starts in $\mathcal{V}(t^\uparrow, t)$, and so does every walk component of $\psi_t(C) - \mathcal{E}(t^\uparrow, t)$. Call a walk component a *U-walk* if it returns to $\mathcal{V}(t^\uparrow, t)$, and an *I-walk* if it doesn't. We now sketch how these two types of walks relate to the definitions of $r(c)$ and α_h.

Firstly, recall that the rank $r(C(t))$ of $A_{t,C(t)}$ should equal the number of walk components of $\psi_t(C) - \mathcal{E}(t^\uparrow, t)$. It is not hard to see that the projection of any U-walk is a non-trivial walk component of $P(t) - \mathcal{E}(t^\uparrow, t)$ containing at least two vertices in $\mathcal{V}(t^\uparrow, t)$. I-walks on the other hand may or may not use edges of $\mathcal{G}(t)$. Note however that if there is an I-walk, then it is necessarily the last walk component of $\psi_t(C) - \mathcal{E}(t^\uparrow, t)$, and it is not hard to see that in this case $X(t) \neq t^\uparrow$. In particular, we have $r(C(t)) = \mu(C(t)) + 1$ if and only if there is an I-walk. Since $\mu(C(t))$ clearly counts the number of U-walks, we conclude that $r(C(t))$ is indeed the number of walk components.

I-walks are also the reason why we need to include β in the definition of $\alpha_{r(C(t))}$. If t has a child $t_i^\downarrow$ such that $\psi_{t_i^\downarrow}(C) - \mathcal{E}(t, t_i^\downarrow)$ contains an I-walk, then $\psi_t(C)$ ends in this I-walk. Note that this happens if and only if $X(t) = t_i^\downarrow$ and the last edge of $P(t)$ is not contained in $\mathcal{E}(t, t_i^\downarrow)$. We would like the production rules to reflect this possibility, but the I-walk only intersects $\mathcal{V}(t)$ in its starting point and thus it does not correspond to any non-trivial walk component of $P(t) \cap \mathcal{G}(t, t_i^\downarrow)$. Adding z_{i,μ_i+1} to the end of the string $\alpha_{r(C(t))}$ allows us to replace the trivial walk consisting of the last vertex of $P(t)$ by such an I-walk. We once again point out that an I-walk always sits at the end of $\psi_t(C)$, so we only have to consider this in the definition of α_h for $h = r(C(t))$.

With the above intuition and the resulting subtleties in mind, let us start by proving some basic results about the grammar $\mathbf{G}$.

Lemma 6.4.2. *The grammar* $\mathbf{G}$ *is a* $\lceil k/2 \rceil$*-multiple context-free.*

Proof. As mentioned in the previous section, R is a finite set and the number of valid configurations on a given part is finite, so $\mathbf{N}$ and $\mathbf{P}$ are finite sets. For the proof of multiple context-freeness of $\mathbf{G}$, it only remains to verify that every expression (6.8) is a well-formed production rule over $(\mathbf{N}, \mathbf{\Sigma})$. Compatibility of $C(t)$ and $C(t_i^{\downarrow})$ implies that $\mu_i = \mu(C(t_i^{\downarrow})))$ for every i. Additionally, if $X(t) = t_i^{\downarrow}$ for some $i \in [k(t)]$ and $P(t)$ does not end with an edge in $\mathcal{E}(t, t_i^{\downarrow})$, then $X(t_i^{\downarrow}) \neq t$ and the final walk component of $P(t_i^{\downarrow}) - \mathcal{E}(t, t_i^{\downarrow})$ contains only a single vertex of $\mathcal{V}(t, t_i^{\downarrow})$, so $\mu_i + 1 = \mu(C(t_i^{\downarrow})) + 1 = r(C(t_i^{\downarrow}))$. We conclude that every variable $z_{i,j}$ with $j \leq r(C(t_i^{\downarrow}))$ occurs in $\alpha_1 \dots \alpha_{r(C(t))}$ and it follows directly from the construction that none of them occurs more than once. As a consequence, $\mathbf{G}$ is multiple-context-free.

Let $P_1, \dots, P_{r(C(t))}$ be the walk components of $P(t) - \mathcal{E}(t^{\uparrow}, t)$. Then P_i contains at least two vertices of $\mathcal{V}(t^{\uparrow}, t)$ for $i < r(C(t))$ and at least one vertex of $\mathcal{V}(t^{\uparrow}, t)$ for $i = r(C(t))$.

The size of $\mathcal{V}(t^{\uparrow}, t)$ is at most k, so $2r(c) - 1 \leq k$ holds, which for an integer k is equivalent to $r(c) \leq \lceil k/2 \rceil$. We conclude that $\mathbf{G}$ is $\lceil k/2 \rceil$-multiple-context-free. □

While the grammar $\mathbf{G}$ may appear more complicated than the 1-multiple context-free grammar of configurations introduced in Section 6.3, the two grammars share many structural similarities. In particular, production rules are again uniquely determined by their heads and tails, so we can again work with simplified derivation trees, where every vertex is labelled with the head of its production rule. In fact we even show that simplified derivation trees of the two grammars are the same.

To this end, let us again define a map ϕ mapping bounded consistent configurations C on T to simplified derivation trees of $\mathbf{G}$. Let the set S consist of all vertices $s \in V(T)$ carrying non-boring configurations $C(s)$ and the neighbours of such vertices. Then $T[S]$ is an ordered tree with root r, where the order on the children of a vertex s is $s_1^{\downarrow}, \dots, s_{k(t)}^{\downarrow}$. By labelling every vertex s of $T[S]$ with $A_{\rho(s), \rho(C(s))}$, we obtain an ordered tree labelled with elements of $\mathbf{N}$.

The following lemma is analogous to Lemma 6.3.4, the proof is exactly the same and is thus omitted.

Lemma 6.4.3. *The map ϕ is a bijection between the set $\mathcal{C}_{\mathcal{T}}$ of bounded consistent configurations on $\mathcal{T}$ and the set of derivation trees whose root is labelled by $A_{r,c}$ for some configuration c on r.*

It remains to show that for any configuration $C \in \mathcal{C}_{\mathcal{T}}$, the word $\ell(\psi_r(C))$ given by the SAW $\psi_r(C)$ on G coincides with the word corresponding to the derivation tree $\phi(C)$.

Lemma 6.4.4. *Let $C \in \mathcal{C}_{\mathcal{T}}$ be a bounded consistent configuration on $\mathcal{T}$. Then*

$$\ell(\psi_r(C)) = w(\phi(C)).$$

Proof. Let $C \in \mathcal{C}_{\mathcal{T}}$. During the proof we denote for $t \in V(T)$ by $\phi_t(C)$ the cone of $\phi(C)$ rooted at t. We prove that whenever $C(t^{\uparrow})$ is non-boring it holds that

$$\ell(\psi_t(C)) = w(\phi_t(C)). \tag{6.9}$$

We proceed by induction on the number of vertices $s \in K_t$ carrying non-boring configurations $C(s)$.

Let $t \in V(T)$ be such that $C(s)$ is boring for every $s \in K_t$. Then $\psi_t(C)$ is the empty walk, so $\ell(\psi_t(C)) = \emptyset$. Furthermore $\phi_t(C)$ consists only of the vertex t labelled $A_{\rho(t),\rho(C(t))}$ and $\rho(C(t))$ is boring, so $w(\phi_t(C)) = \emptyset$.

For the induction step we first set up some notation. Let r be the number of walk components of $\psi_t(C) - \mathcal{E}(t^\uparrow, t)$. For $h \in [r]$, let Q_h denote the h-th walk component of $\psi_t(C) - \mathcal{E}(t^\uparrow, t)$, and let $w_h = \ell(Q_h)$. By definition $\ell(\psi_t(C)) = (w_1, \ldots, w_r)$. Analogously, for each child $t_i^\downarrow$ of t, we define r_i as the number of walk components of $\psi_{t_i^\downarrow}(C) - \mathcal{E}(t, t_i^\downarrow)$. For $j \in [r_i]$, let Q_i^j be the j-th walk component of $\psi_{t_i^\downarrow}(C) - \mathcal{E}(t, t_i^\downarrow)$ and let w_i^j be the label of Q_i^j. The definition of ℓ together with the induction hypothesis imply that

$$w(\phi_{t_i^\downarrow}(C)) = \ell(\psi_{t_i^\downarrow}(C)) = (w_i^1, \ldots, w_i^{r_i}),$$

in particular $r_i = r(C(t_i^\downarrow))$. We examine the left-hand side and right-hand side of (6.9) independently, and show that they yield the same tuple of words.

For the left-hand side first recall that for any vertex s of T we have that $\psi_s(C) = P/F(t_F)$ where F is the set of edges in K_s incident to vertices with non-boring configurations. By Lemma 6.2.6 the order of edge contractions does not play a role, in particular $\psi_t(C)$ can be obtained by first contracting all such edges not incident to t, and then contracting the edges connecting t to $t_i^\uparrow$ one by one.

This means that we can construct $\psi_t(C)$ from $P(t)$ by performing the following modifications for each $i \in [k(t)]$. For every virtual edge $uv \in P(t) \cap \mathcal{E}(t, t_i^\downarrow)$, the walk $\psi_{t_i^\downarrow}(C)$ contains a sub-walk with the same endpoints entirely consisting of non-virtual edges. Replace every such virtual edge in $P(t)$ by the corresponding walk in $\psi_{t_i^\downarrow}(C)$. If $X(t) = t_i^\downarrow$, then append the sub-walk of $\psi_{t_i^\downarrow}(C)$ after the last vertex in $\mathcal{V}(t, t_i^\downarrow)$ to the resulting walk. Note that equivalently, we can let $U_i^1, \ldots, U_i^{\mu_i}$ be the sequence of non-trivial walk components of $P(t) \cap \mathcal{G}(t, t_i^\downarrow)$, replace every U_i^j by the respective Q_i^j, and append $Q_i^{r_i}$ in case $X(t) = t_i^\downarrow$ and $P(t)$ does not end in a virtual edge in $\mathcal{E}(t, t_i^\downarrow)$.

Let $P_1, \ldots, P_{r(C(t))}$ be the walk components of $P(t) - \mathcal{E}(t^\uparrow, t)$ and for every $h \in [r(C(t))]$ let $P_h = P_h^1 P_h^2 \ldots P_h^{2m+1}$ be the unique decomposition into sub-walks such that P_h^l is a possibly trivial non-virtual walk if l is odd and equal to some U_i^j if l is even. By the above discussion, for $h < r(C(t))$ we thus have

$$\ell(Q_h) = \ell(P_h^1)\tilde{\ell}(P_h^2)\ell(P_h^3)\tilde{\ell}(P_h^4)\ldots\ell(P_h^{2m+1}),$$

where $\tilde{\ell}(P_h^l) = \ell(Q_i^j) = w_i^j$ for the unique indices i, j satisfying $P_h^l = U_i^j$. For the final walk component, that is, $h = r(C(t))$, we analogously obtain

$$\ell(Q_h) = \ell(P_h^1)\tilde{\ell}(P_h^2)\ell(P_h^3)\tilde{\ell}(P_h^4)\ldots\ell(P_h^{2m+1})\beta,$$

where $\tilde{\ell}(P_h^l) = w_i^j$ as above and $\beta = \epsilon$, unless $X(t) = t_i^\downarrow$ and $P(t)$ does not end with an edge in $\mathcal{E}(t, t_i^\downarrow)$, in which case $\beta = \ell(Q_i^{\mu_i+1}) = w_i^{\mu_i+1}$.

We now turn to the right-hand side of (6.9). For every $i \in [k(t)]$, the induction hypothesis implies that $\phi_{t_i^\downarrow}(C)$ is a derivation tree of the term

$$\tau_i := A_{\rho(t_i^\downarrow),\rho(C(t_i^\downarrow))}\left(w_i^1, \ldots, w_i^{r_i}\right).$$

Moreover the root t of $\phi_t(C)$ has label $A_{\rho(t),\rho(C(t))}$, so $\phi_t(C)$ is a derivation tree of the term obtained by application of the rule

$$A_{\rho(t),\rho(C(t))}(\alpha_1,\ldots,\alpha_{r(C(t))}) \leftarrow \Big(A_{\rho(t_i^\downarrow),\rho(C(t_i^\downarrow))}(z_{i,1},\ldots,z_{i,r(C(t_i^\downarrow))})\Big)_{i\in[r(C(t))]}$$

to $(\tau_i)_{i\in[k(t)]}$. By definition of α_h, the h-th entry of this term is obtained from the w_i^j in the exact same way as $\ell(Q_h)$ and we conclude that $\ell(\psi_t(C)) = w(\phi_t(C))$ as claimed. $\square$

We are now able to prove the main result of this section by combining the previous results.

Proof of Theorem 6.4.1. Theorem 6.2.7 yields that the language of self-avoiding walks of the graph G satisfies

$$L_{\mathrm{SAW},o}(G) = \{\ell(\psi_r(C)) \mid C \in \mathcal{C}_{\mathcal{T}}\}.$$

Furthermore Lemma 6.4.3 implies that the language generated by the grammar $\mathbf{G}$ is given by

$$L_{\mathbf{G}} = \{w(\phi(C)) \mid C \in \mathcal{C}_{\mathcal{T}}\}.$$

But by Lemma 6.4.4 these two sets are equal and $\mathbf{G}$ is a $\lceil k/2\rceil$-multiple-context-free grammar generating $L_{\mathrm{SAW},o}(G)$.

Finally, if the edge-labelling of G is deterministic, then ℓ is a bijection between the set of self-avoiding walks on G and $L_{\mathrm{SAW},o}(G)$. Lemma 6.4.4 provides equality of the maps $w \circ \phi = \ell \circ \psi_r$, so in particular $w \circ \phi$ is also a bijection. We conclude that w bijectively maps derivation trees with respect to $\mathbf{G}$ onto words in $L(\mathbf{G})$, so $\mathbf{G}$ is unambiguous. $\square$

6.4.2 Multiple context-freeness implies bounded end size

In this section we prove the second part of our main result, namely

Theorem 6.4.5. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph such that $L_{\mathrm{SAW},o}(G)$ is k-multiple context-free for some $o \in V(G)$. Then every end of G has size at most $2k$.*

As mentioned before, the proof of this statement will mostly follow the approach in Section 4.2.

Recall that any graph automorphism is either elliptic, parabolic or hyperbolic, depending on whether it fixes a finite subset of vertices, a unique end or a unique pair of ends. In what follows, elliptic automorphisms are useless, so as a first step we eliminate the possibility that all label-preserving automorphisms are elliptic.

We remark that there are numerous examples of infinite graphs admitting a transitive group action by only elliptic automorphisms. To see this, note that any non-elliptic automorphism must have infinite order because it cannot fix a finite set of vertices. Therefore some interesting examples arise from the study of the famous Burnside Problem from 1902, asking whether every finitely generated torsion group, that is a group in which every element has finite order, must be finite. While this question remained unsolved for more than 60 years, nowadays various examples of infinite torsion groups are known. Any such group acts transitively on its Cayley graph by only elliptic automorphisms.

However, if $L_{\mathrm{SAW},o}(G)$ is multiple context-free, then there are always non-elliptic automorphisms. The following lemma extends Lemma 4.2.3 to multiple context-free languages; the proof is essentially the same.

Lemma 6.4.6. *Let G be a connected, infinite, locally finite and deterministically edge-labelled graph and let $\Gamma \leq \mathrm{AUT}(G, \ell)$ act quasi-transitively on G. If $L_{\mathrm{SAW},o}(G)$ is multiple context-free for some vertex o of G, then Γ contains a non-elliptic element.*

Proof. The graph G is infinite and connected, so $L_{\mathrm{SAW},o}(G)$ is an infinite language. Thus by Lemma 2.2.8 the k-multiple context-free language $L_{\mathrm{SAW},o}(G)$ contains some word $w = x_1 y_1 x_2 \dots y_{2k} x_{2k+1}$ such that at least one of $y_1, \dots, y_{2k}$ is not the empty word ϵ, and $x_1 y_1^n x_2 \dots y_{2k}^n x_{2k+1} \in L_{\mathrm{SAW},o}(G)$ for every $n \in \mathbb{N}_0$. Let $m = \min\{i \in [2k] \mid y_i \neq \epsilon\}$. Then for every $n \in \mathbb{N}_0$, the word $x_1 \dots x_m y_m^n$ is a prefix of some word in $L_{\mathrm{SAW},o}(G)$ and thus itself contained in $L_{\mathrm{SAW},o}(G)$. Let v_0 be the end-vertex of the unique walk p on G starting at o and having label $\ell(p) = x_1 \dots x_m$. Then for every $n \geq 0$ there is a unique self-avoiding walk p_n of length $n\,|y_m|$ starting at v_0 and having label y_m^n. We denote by v_n the endpoint of the walk p_n. Using the fact that Γ acts quasi-transitively on G, there must be some $\tau \in \Gamma$ and some $0 \leq i < j \leq n$ such that $\tau v_i = v_j$. Since τ is label preserving, $\tau^l v_i = v_{j+(l-1)(j-i)} \neq v_i$ for every $l > 0$ and Proposition 12 in [26] yields that τ is non-elliptic. □

Recall from Section 4.1 that a locally finite, connected graph S is called a strip if it is quasi-transitive and has precisely two ends. It is called a τ-strip for $\tau \in \mathrm{AUT}(S)$, if the cyclic group $\langle \tau \rangle$ generated by τ acts quasi-transitively on S. We use the same notation if S is a subgraph of a graph G invariant under $\tau \in \mathrm{AUT}(G)$. Note that both ends of a strip have the same finite size k which we call the size of S. It follows directly from Theorem 4.1.5 that any τ-strip of size k contains k disjoint τ-invariant double rays.

The following lemma is a combination of Lemma 4.1.8 and Lemma 4.1.9 and provides the existence of τ-strips in certain types of graphs. In particular, it implies that τ-strips exist whenever there is a non-elliptic automorphism.

Lemma 6.4.7. *Let G be a locally finite connected graph and let Γ act quasi-transitively on G.*

1. *If G has a thin end of size k, then it contains a τ-strip of size k for some $\tau \in \Gamma$.*
2. *If Γ contains a parabolic element, then for every $k \geq 1$, the graph G contains a τ-strip of size k for some $\tau \in \Gamma$.*

With these existence results for τ-strips in a graph G in mind, we turn to the relation between strips in a graph G and its language of SAWs. A combination of the previous lemma and the upcoming lemma is already sufficient to treat graphs G without thick ends.

Lemma 6.4.8. *Let G be a connected, infinite, deterministically edge-labelled graph on which* $\mathrm{AUT}(G, \ell)$ *acts quasi-transitively, let o be a vertex of G and let $k \in \mathbb{N}$. If G contains a τ-strip of size $2k+1$ for some $\tau \in \mathrm{AUT}(G, \ell)$, then $L_{\mathrm{SAW},o}(G)$ is not k-multiple context-free.*

Proof. The proof can be outlined as follows. We start by defining an infinite set $\mathcal{P}$ of walks such that firstly, the language $\ell(\mathcal{P})$ is regular, and secondly, the language $\ell(\mathcal{P}_{\mathrm{SAW}})$ of the subset $\mathcal{P}_{\mathrm{SAW}}$ consisting of all self-avoiding walks in $\mathcal{P}$ is not k-multiple context-free. It then follows from the closure properties of k-multiple context free languages that $L_{\mathrm{SAW},o}(G)$ is not k-multiple context-free.

The set $\mathcal{P}$ essentially consists of spiral-shaped walks on the strip S, see Figure 6.11. For a concise definition, first recall that the strip S contains $2k+1$ τ-invariant rays $R_1, \dots, R_{2k+1}$ and that the subgroup $\langle \tau \rangle$ of $\mathrm{AUT}(G, \ell)$ generated by τ acts quasi-transitively on S. After possibly replacing τ with some power τ^l, we can find a set K be a set of orbit representatives of the action of $\langle \tau \rangle$ on S such that the induced subgraphs $S[K]$ and $R_i[K]$ (for every $i \in [2k+1]$) are connected. See Lemma 4.1.7 on how to find such a set K.

Let T_K be a spanning tree of $S[K]$ containing all edges of the paths $R_i[K]$. Such a tree exists because $S[K]$ is connected and the rays R_i are disjoint and acyclic. For $j \in [2k]$ let $T_K(j)$ be the smallest sub-tree of T_K containing the paths $R_j[K], \dots, R_{2k+1}[K]$. We call a ray R_i pendant in $T_K(j)$ if $T_K(j)$ contains precisely one edge connecting a vertex of R_i to a vertex not in R_i. Clearly $T_K(j)$ contains at least two pendant rays and we may relabel the rays in a way such that for every $i \in [2k]$, the rays R_i and R_{i+1} are pendant in the tree $T_K(i)$.

For $i, j \in [2k+1]$ with $i \neq j$ let $W_{i,j}$ be the path connecting R_i to R_j in T_K. Furthermore let $W_{0,2}$ consist of a shortest walk connecting o to some $v_0 \in \tau^n(T_K)$, followed by a walk connecting v_0 and R_2 on $\tau^n(T_K)$.

It will be convenient to slightly abuse notation and define concatenations of walks whose endpoint and starting point do not coincide, but the endpoint of the first walk can be mapped onto onto the starting point by a suitable power of τ. More precisely, let P and Q be two walks on S, let u be the endpoint of P and let v be the starting point of Q. If $v = \tau^i(u)$, we write PQ for the concatenation of P and $\tau^i Q$.

Using this notation, for each $i \in [2k]$ let us define a walk

$$X_i = W_{i-1,i+1} Q W_{i+1,i},$$

where Q is the path connecting the endpoint of $W_{i-1,i+1}$ to the starting point of $\tau(W_{i+1,i})$ on the ray R_{i+1} if i is odd and the endpoint of $\tau(W_{i-1,i+1})$ to the starting point of $W_{i+1,i}$ if i is even. Note that we apply the notation defined above, so X_i consists of the paths $W_{i-1,i+1}$, Q and $\tau(W_{i+1,i})$. In particular X_i is self-avoiding because $W_{i-1,i+1}$ and $W_{i+1,i}$ are fully contained in $S[K]$. Moreover, let $X_{2k+1} = W_{2k,2k+1} Q$, where Q connects the endpoint of $W_{2k,2k+1}$ with its image under τ on the ray R_{2k+1}. Furthermore observe that X_i is contained in $T_K(i-1)$ meets R_{i-1} and R_i only in its endpoints.

Next, for every $i \in [2k+1]$ let r_i be the terminal vertex of X_i. Note that r_i is a vertex of R_i. Moreover for $i \leq 2k$ it lies in the same orbit as the initial vertex of X_{i+1} because

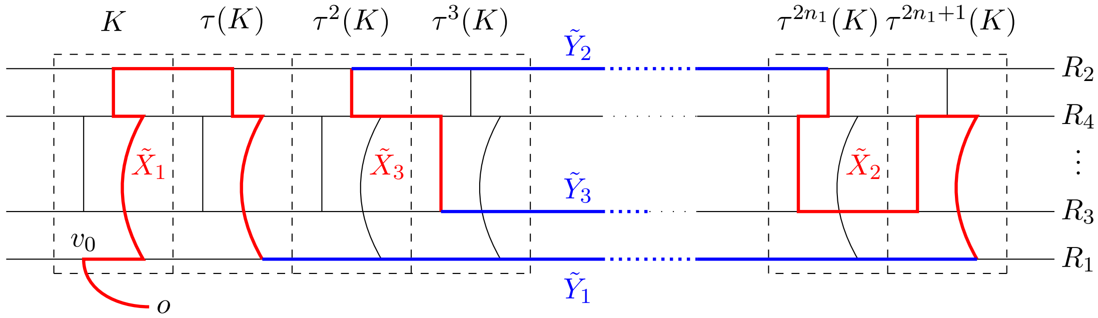


Figure 6.11: Spiral shaped walks in $\mathcal{P}$. The dashed rectangles contain the set K and its respective translates under τ^i.

the rays R_i and R_{i+1} are pendant in the tree $T_K(i)$. Finally let Y_i be the sub-path of R_i connecting r_i with $\tau^2(r_i)$ if i is odd and let Y_i be the sub-path of R_i connecting $\tau^2(r_i)$ with r_i if i is even.

Let $\mathcal{P}$ be the infinite set of walks of the form

$$X_1 Y_1^{n_1} X_2 Y_2^{n_2} \dots X_{2k+1} Y_{2k+1}^{n_{2k+1}}, \tag{6.10}$$

where $n_1, \dots, n_{2k+1} \in \mathbb{N}$. See Figure 6.11 for an illustration of an element of $\mathcal{P}$.

The language $L(\mathcal{P})$ has the form

$$L(\mathcal{P}) = \{x_1 y_1^{n_1} x_2 y_2^{n_2} \dots x_{2k+1} y_{2k+1}^{n_{2k+1}} \mid n_1, \dots, n_{2k+1} \in \mathbb{N}\} \tag{6.11}$$

where the words x_i and y_i are the labels of the walks X_i and Y_i, respectively. Clearly, $L(\mathcal{P})$ is a regular language.

We claim that a walk of type (6.10) is self-avoiding if and only if $n_{i+1} < n_i$ for every $i \in [2k]$. Fix a walk $W = X_1 Y_1^{n_1} X_2 Y_2^{n_2} \dots X_{2k+1} Y_{2k+1}^{n_{2k+1}} \in \mathcal{P}$ and denote by $\tilde{X}_i$ the sub-walk of W corresponding to X_i and by $\tilde{Y}_i$ the sub-walk of W corresponding to the concatenated walk $Y_i^{n_i}$. In the example shown in Figure 6.11 we have $n_2 = n_1 - 1$. Observe that $n_2 \geq n_1$ would yield a self-intersection on R_2.

We say that a vertex $v \in S$ lies on level $l \in \mathbb{Z}$, if $v \in \tau^{2l}(K) \cup \tau^{2l+1}(K)$. First note that the walk $\tilde{X}_1$ does not contain vertices on level $l \geq 1$. Moreover, it follows inductively that $\tilde{X}_i$ contains only vertices on level $l_i = \sum_{j=1}^{i-1} (-1)^{j-1} n_j$ and that $\tilde{Y}_i$ starts on level l_i and ends on level l_{i+1}.

Assume that the walk W is not self-avoiding. Then there is some index $i \in [2k+1]$ such that $\tilde{X}_i$ intersects either $\tilde{X}_j$ for $j > i$ or contains an interior point of $\tilde{Y}_j$ for some $j \in [2k+1]$. For $j < i-1$, the walk $\tilde{X}_i$ does not intersect $\tilde{Y}_j$ because R_j does not intersect $T_K(i-1)$. For $j \in \{i-1, i\}$, the walk $\tilde{X}_i$ contains only a single vertex of R_j, which is an endpoint of $\tilde{Y}_j$. Therefore $j > i$, and in particular $\tilde{Y}_j$ contains a vertex on level l_i. Without loss of generality assume that j is odd, the other case is symmetric. Since $\tilde{Y}_j$ connects levels l_j to $l_{j+1} > l_j$ we conclude that $l_{j+1} \geq l_i \geq l_j$. If i is odd, then

$$0 \geq l_j - l_i = \sum_{l=i}^{j-1} (-1)^{l-1} n_l = (n_i - n_{i+1}) + \dots + (n_{j-2} - n_{j-1}),$$

so there is some index l such that $n_{l-1} \leq n_l$. Otherwise i is even and an analogous argument using $0 \leq l_{j+1} - l_i$ leads to the same conclusion.

For the converse implication assume that there is $i \in [2k]$ such that $n_{i+1} \geq n_i$. We claim that the sub-walk $\tilde{X}_i \tilde{Y}_i \tilde{X}_{i+1} \tilde{Y}_{i+1}$ of W is not self-avoiding. Assume without loss of generality that i is odd, the other case is symmetric. Since $n_i - n_{i+1} \leq 0$ we know that $l_{i+2} \leq l_i < l_{i+1}$. In particular, both $\tilde{X}_i$ and $\tilde{Y}_{i+1}$ contain vertices on level l_i. Moreover, by definition of X_i the walk $\tilde{X}_i$ contains a vertex v of $R_{i+1} \cap \tau^{2l_i+1}(K)$. Finally the sub-path $\tilde{Y}_{i+1}$ of R_{i+1} starts at a vertex in $\tau^{2l_{i+1}}(K)$ and ends at a vertex in $\tau^{2l_{i+2}}(K)$ and thus must contain the vertex v. We conclude that W is not self-avoiding.

Let us now assume that the language $L_{\mathrm{SAW},o}(G)$ is k-multiple context-free. Recall that the class of k-multiple context-free languages is closed under homomorphisms, inverse homomorphisms, and intersection with regular languages. Using these properties and Theorem 5.0.1, we derive a contradiction. First, note that the language

$$L(\mathcal{P}_{\mathrm{SAW}}) = \{x_1 y_1^{n_1} x_2 y_2^{n_2} \dots x_{2k+1} y_{2k+1}^{n_{2k+1}} \mid n_1 > n_2 > \dots > n_{2k+1} > 0\}$$

is the intersection of the regular language $L(\mathcal{P})$ and the k-multiple context-free language $L_{\mathrm{SAW},o}(G)$ and thus must be k-multiple context-free. We define a language homomorphism

$$\phi : \{a_1, b_1, \dots, a_{2k+1}, b_{2k+1}\}^* \to \mathbf{\Sigma}^*$$

by setting $\phi(a_i) = x_i y_i^{2k+2-i}$ and $\phi(b_i) = y_i$ for every $i \in [2k+1]$; we point out that a_i and b_i are single letters, whereas x_i and y_i are labels of walks and thus may consist of multiple letters. The language

$$L_1 = \{a_1 b_1^{n_1} \dots a_{2k+1} b_{2k+1}^{n_{2k+1}} \mid n_1 \geq n_2 \geq \dots \geq n_{2k+1} \geq 0\}$$

is k-multiple context-free because

$$L_1 = \phi^{-1}(L(\mathcal{P}_{\mathrm{SAW}})) \cap \{a_1 b_1^{n_1} \dots a_{2k+1} b_{2k+1}^{n_{2k+1}} \mid n_1, \dots, n_{2k+1} \in \mathbb{N}_0\}.$$

Note that this statement strongly relies on the fact that the edge-labelling ℓ is deterministic: the image $\phi(w)$ of any word $w \in \{a_1 b_1^{n_1} \dots a_{2k+1} b_{2k+1}^{n_{2k+1}} \mid n_1, \dots, n_{2k+1} \in \mathbb{N}_0\}$ is the label of a unique walk in $\mathcal{P}$ and thus has a unique representation of the form (6.11), which lies in $L(\mathcal{P}_{\mathrm{SAW}})$ if and only if $n_1 \geq n_2 \geq \dots \geq n_{2k+1} \geq 0$.

Finally, the language $L_2 = \{c_1^{n_1} \dots c_{2k+1}^{n_{2k+1}} \mid n_1 \geq n_2 \geq \dots \geq n_{2k+1} \geq 0\}$ is the image of L_1 under the obvious homomorphism mapping a_i to ϵ and b_i to c_i and thus must be k-multiple context-free, a contradiction to Theorem 5.0.1. □

The property of having a k-multiple context-free language of SAWs is closed under taking certain subgraphs. The following lemma extends Lemma 4.2.2 to MCFLs; the proof works exactly the same and is thus omitted.

Lemma 6.4.9. *Let H be a subgraph of G which is invariant under a subgroup Γ of $\mathrm{AUT}(G, \ell)$ acting quasi-transitively on H. If $L_{\mathrm{SAW},o}(G)$ is k-multiple context-free, then there is a vertex $o' \in V(H)$ such that $L_{\mathrm{SAW},o'}(H)$ is also k-multiple context-free.*

Knowing that k-multiple context-freeness of the language of self-avoiding walks forbids τ-strips of size at least $2k+1$, we are able to prove the main result of this section. Note that the case of thin ends is already taken care of, so we mainly need to deal with thick ends in the proof.

Proof of Theorem 6.4.5. By Lemma 6.4.8 there is no $\tau \in \Gamma = \mathrm{AUT}(G, \ell)$ such that the graph G contains a τ-strip of size $2k+1$. Lemma 6.4.7 yields that all thin ends of G have size at most $2k$, thus G is accessible.

Assume for a contradiction that G contains a thick end. By Theorem 6.1.1 there is a tree decomposition $(T, \mathcal{V})$ of G efficiently distinguishing all ends of G such that there are only finitely many Γ-orbits on $E(T)$. We have seen in Section 6.1 that G having a thick end implies that there is a vertex t of T such that the part $\mathcal{V}(t)$ is infinite. By Lemma 6.1.4 the set-wise stabiliser $\Gamma_{\mathcal{V}(t)} \leq \Gamma$ of $\mathcal{V}(t)$ acts quasi-transitively on this part.

Let H be the subgraph of G obtained from the induced subgraph $G[\mathcal{V}(t)]$ in the following way. For every edge e of T incident to t add for every pair of vertices in the adhesion set $\mathcal{V}(e)$ all shortest walks connecting these vertices. Then H is connected and $\Gamma_{\mathcal{V}(t)}$ acts quasi-transitively on H because it acts with finitely many orbits on the edges of T and thus on the adhesion sets contained in $\mathcal{V}(t)$. By Lemma 6.4.9 there exists a vertex o' of H such that the language $L_{\mathrm{SAW},o'}(H)$ is k-multiple context-free and by Lemma 6.4.6 the subgroup $\Gamma_{\mathcal{V}(t)}$ contains a non-elliptic graph automorphism γ. As H has only a single (thick) end, γ fixes this end and is parabolic. By Lemma 6.4.7, the graph H contains a τ-strip S of size $2k+1$ for some $\tau \in \Gamma_{\mathcal{V}(t)}$. But S is also a τ-strip in the original graph G, contradicting Lemma 6.4.8. We conclude that all ends of G are thin. □

Bibliography

[1] S. E. Alm and S. Janson. "Random self-avoiding walks on one-dimensional lattices". In: *Commun. Stat., Stochastic Models* 6.2 (1990), pp. 169–212.

[2] A. V. Anisimov. "Group languages". In: *Cybernetics* 7.4 (July 1971), pp. 594–601.

[3] R. Bauerschmidt, H. Duminil-Copin, J. Goodman, and G. Slade. "Lectures on self-avoiding walks". In: *Probability and statistical physics in two and more dimensions.* Vol. 15. Clay Math. Proc. Amer. Math. Soc., Providence, RI, 2012, pp. 395–467.

[4] T. Ceccherini-Silberstein, M. Coornaert, F. Fiorenzi, P. E. Schupp, and N. W. M. Touikan. "Multipass automata and group word problems". In: *Theor. Comput. Sci.* 600 (2015), pp. 19–33.

[5] T. Ceccherini-Silberstein and W. Woess. "Growth and ergodicity of context-free languages". In: *Trans. Am. Math. Soc.* 354.11 (2002), pp. 4597–4625.

[6] T. Ceccherini-Silberstein and W. Woess. "Context-free pairs of groups. I: Context-free pairs and graphs". In: *Eur. J. Comb.* 33.7 (2012), pp. 1449–1466.

[7] N. Chomsky and M. P. Schützenberger. "The algebraic theory of context-free languages". In: *Computer programming and formal systems.* North-Holland, Amsterdam, 1963, pp. 118–161.

[8] M. Dehn. "Über unendliche diskontinuierliche Gruppen". In: *Math. Ann.* 71.1 (1911), pp. 116–144.

[9] M. Dehn. "Transformation der Kurven auf zweiseitigen Flächen". In: *Math. Ann.* 72.3 (1912), pp. 413–421.

[10] W. Dicks and M. J. Dunwoody. *Groups acting on graphs.* Vol. 17. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 1989, pp. xvi+283.

[11] V. Diekert and A. Weiß. "Context-free groups and their structure trees". In: *Int. J. Algebra Comput.* 23.3 (2013), pp. 611–642.

[12] C. Droms, B. Servatius, and H. Servatius. "The structure of locally finite two-connected graphs". In: *Electron. J. Comb.* 2 (1995).

[13] H. Duminil-Copin and S. Smirnov. "The connective constant of the honeycomb lattice equals $\sqrt{2+\sqrt{2}}$". In: *Ann. Math. (2)* 175.3 (2012), pp. 1653–1665.

[14] M. J. Dunwoody. "Cutting up graphs". In: *Combinatorica* 2.1 (1982), pp. 15–23.

[15] M. J. Dunwoody and B. Krön. "Vertex cuts". In: *J. Graph Theory* 80.2 (2015), pp. 136–171.

[16] P. J. Flory. *Principles of Polymer Chemistry.* Cornell University Press, 1953.

[17] H. Freudenthal. “Über die Enden topologischer Räume und Gruppen”. In: *Math. Z.* 33.1 (1931), pp. 692–713.

[18] H. Freudenthal. “Über die Enden diskreter Räume und Gruppen”. In: *Comment. Math. Helv.* 17 (1945), pp. 1–38.

[19] L. A. Gilch and S. Müller. “Counting self-avoiding walks on free products of graphs”. In: *Discrete Math.* 340.3 (2017), pp. 325–332.

[20] G. R. Grimmett and Z. Li. “Connective constants and height functions for Cayley graphs”. In: *Trans. Amer. Math. Soc.* 369.8 (2017), pp. 5961–5980.

[21] G. R. Grimmett and Z. Li. “Self-avoiding walks and amenability”. In: *Electron. J. Combin.* 24.4 (2017), Paper 4.38, 24.

[22] G. R. Grimmett and Z. Li. “Locality of connective constants”. In: *Discrete Math.* 341.12 (2018), pp. 3483–3497.

[23] G. R. Grimmett and Z. Li. “Self-Avoiding Walks and Connective Constants”. In: *Sojourns in Probability Theory and Statistical Physics - III.* Ed. by V. Sidoravicius. Singapore: Springer Singapore, 2019, pp. 215–241.

[24] R. Halin. “Über unendliche Wege in Graphen”. In: *Math. Ann.* 157 (1964), pp. 125–137.

[25] R. Halin. “Über die Maximalzahl fremder unendlicher Wege in Graphen”. In: *Math. Nachr.* 30 (1965), pp. 63–85.

[26] R. Halin. “Automorphisms and endomorphisms of infinite locally finite graphs”. In: *Abh. Math. Semin. Univ. Hamb.* 39 (1973), pp. 251–283.

[27] R. Halin. “S-functions for graphs”. In: *J. Geom.* 8.1-2 (1976), pp. 171–186.

[28] M. Hamann, F. Lehner, B. Miraftab, and T. Rühmann. *A Stallings’ type theorem for quasi-transitive graphs.* 2018. arXiv: `1812.06312 [math.CO]`.

[29] J. M. Hammersley. “Percolation processes. II: The connective constant”. In: *Proc. Camb. Philos. Soc.* 53 (1957), pp. 642–645.

[30] J. M. Hammersley and D. J. A. Welsh. “Further results on the rate of convergence to the connective constant of the hypercubical lattice”. In: *Quart. J. Math. Oxford Ser. (2)* 13 (1962), pp. 108–110.

[31] G. H. Hardy and S. Ramanujan. “Asymptotic formulæ for the distribution of integers of various types [Proc. London Math. Soc. (2) **16** (1917), 112–132]”. In: *Collected papers of Srinivasa Ramanujan.* AMS Chelsea Publ., Providence, RI, 2000, pp. 245–261.

[32] M. A. Harrison. *Introduction to formal language theory.* Addison-Wesley Publishing Co., Reading, Mass., 1978, pp. xiv+594.

[33] M.-C. Ho. “The word problem of $\mathbb{Z}^n$ is a multiple context-free language.” In: *Groups Complex. Cryptol.* 10.1 (2018), pp. 9–15.

[34] W. Imrich and N. Seifter. “A note on the growth of transitive graphs”. In: *Discrete Math.* 73.1-2 (1989), pp. 111–117.

[35] W. Imrich and N. Seifter. “A survey on graphs with polynomial growth”. In: *Discrete Math.* 95.1-3 (1991), pp. 101–117.

[36] H. A. Jung and M. E. Watkins. “Fragments and automorphisms of infinite graphs”. In: *Eur. J. Comb.* 5 (1984), pp. 149–162.

[37] M. Kanazawa. “The pumping lemma for well-nested multiple context-free languages”. In: *Developments in language theory.* Vol. 5583. Lecture Notes in Comput. Sci. Springer, Berlin, 2009, pp. 312–325.

[38] M. Kanazawa, G. M. Kobele, J. Michaelis, S. Salvati, and R. Yoshinaka. “The failure of the strong pumping lemma for multiple context-free languages.” In: *Theory Comput. Syst.* 55.1 (2014), pp. 250–278.

[39] W. Kuich and A. Salomaa. *Semirings, automata, languages.* Vol. 5. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1986, pp. v+374.

[40] F. Lehner and C. Lindorfer. *Self-avoiding walks and multiple context-free languages.* 2020. arXiv: `2010.06974 [math.CO]`.

[41] F. Lehner and C. Lindorfer. “Comparing consecutive letter counts in multiple context-free languages”. In: *Theor. Comput. Sci.* 868 (2021), pp. 1–5.

[42] C. Lindorfer. *The language of self-avoiding walks. Connective constants of quasi-transitive graphs.* Wiesbaden: Springer Spektrum (Master Thesis), 2018, pp. viii + 65.

[43] C. Lindorfer. “A general bridge theorem for self-avoiding walks”. In: *Discrete Math.* 343.12 (2020). Id/No 112092, p. 11.

[44] C. Lindorfer and W. Woess. “The Language of Self-Avoiding Walks”. In: *Combinatorica* 40.5 (Nov. 2020), pp. 691–720.

[45] N. Madras and G. Slade. *The self-avoiding walk.* Boston, MA: Birkhäuser, 1993, pp. xiv + 425.

[46] D. E. Muller and P. E. Schupp. “Groups, the theory of ends, and context-free languages”. In: *J. Comput. Syst. Sci.* 26 (1983), pp. 295–310.

[47] D. E. Muller and P. E. Schupp. “The theory of ends, pushdown automata, and second-order logic”. In: *Theor. Comput. Sci.* 37 (1985), pp. 51–75.

[48] L. Pélecq. “Automorphism groups of context-free graphs”. In: *Theor. Comput. Sci.* 165.2 (1996), pp. 275–293.

[49] N. Robertson and P. D. Seymour. “Graph minors. III. Planar tree-width”. In: *J. Combin. Theory Ser. B* 36.1 (1984), pp. 49–64.

[50] G. Sabidussi. “On a class of fixed-point-free graphs”. In: *Proc. Am. Math. Soc.* 9 (1958), pp. 800–804.

[51] S. Salvati. “MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is captured by the IO and the OI hierarchies.” In: *J. Comput. Syst. Sci.* 81.7 (2015), pp. 1252–1277.

[52] H. Seki, T. Matsumura, M. Fujii, and T. Kasami. “On multiple context-free grammars.” In: *Theor. Comput. Sci.* 88.2 (1991), pp. 191–229.

[53] J.-P. Serre. *Trees.* Translated from the French by John Stillwell. Springer-Verlag, Berlin-New York, 1980, pp. ix+142.

[54] P. M. Soardi and W. Woess. “Amenability, unimodularity, and the spectral radius of random walks on infinite graphs”. In: *Math. Z.* 205.3 (1990), pp. 471–486.

[55] C. Thomassen. “The Hadwiger number of infinite vertex-transitive graphs”. In: *Combinatorica* 12.4 (1992), pp. 481–491.

[56] C. Thomassen and W. Woess. “Vertex-transitive graphs and accessibility”. In: *J. Comb. Theory, Ser. B* 58.2 (1993), pp. 248–268.

[57] V. I. Trofimov. “Automorphism groups of graphs as topological groups”. In: *Math. Notes* 38 (1985), pp. 717–720.

[58] W. T. Tutte. *Graph theory. Foreword by Crispin St. J. A. Nash-Williams.* Vol. 21. Cambridge University Press, Cambridge, 1984.

[59] K. Vijay-Shanker, D. J. Weir, and A. K. Joshi. “Characterizing Structural Descriptions Produced by Various Grammatical Formalisms”. In: *25th Annual Meeting of the Association for Computational Linguistics.* Stanford, California, USA: Association for Computational Linguistics, July 1987, pp. 104–111.

[60] W. Woess. “Graphs and groups with tree-like properties”. In: *J. Combin. Theory Ser. B* 47.3 (1989), pp. 361–371.

[61] W. Woess. *Random walks on infinite graphs and groups.* Vol. 138. Cambridge: Cambridge University Press, 2000, pp. xi + 334.

[62] W. Woess. “Context-free pairs of groups. II: Cuts, tree sets, and random walks”. In: *Discrete Math.* 312.1 (2012), pp. 157–173.

[63] D. Zeilberger. “Self-avoiding walks, the language of science, and Fibonacci numbers”. In: *J. Statist. Plann. Inference* 54.1 (1996), pp. 135–138.