

eScholarship

Combinatorial Theory

Title

Self-avoiding walks and multiple context-free languages

Permalink

<https://escholarship.org/uc/item/8928m3zx>

Journal

Combinatorial Theory, 3(1)

ISSN

2766-1334

Authors

Lehner, Florian
Lindorfer, Christian

Publication Date

2023

DOI

10.5070/C63160431

Supplemental Material

<https://escholarship.org/uc/item/8928m3zx#supplemental>

Copyright Information

Copyright 2023 by the author(s). This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

SELF-AVOIDING WALKS AND MULTIPLE CONTEXT-FREE LANGUAGES

Florian Lehner^{*1} and Christian Lindorfer^{†2}

¹*Department of Mathematics, The University of Auckland, New Zealand
mail@florian-lehner.net*

²*Institut für Diskrete Mathematik, Technische Universität Graz, Austria
lindorfer@math.tugraz.at*

Submitted: May 16, 2022; Accepted: Jan 19, 2023; Published: Mar 15, 2023

© The authors. Released under the CC BY license (International 4.0).

Abstract. Let G be a quasi-transitive, locally finite, connected graph rooted at a vertex o , and let $c_n(o)$ be the number of self-avoiding walks of length n on G starting at o . We show that if G has only thin ends, then the generating function $F_{\text{SAW},o}(z) = \sum_{n \geq 1} c_n(o)z^n$ is an algebraic function. In particular, the connective constant of such a graph is an algebraic number.

If G is deterministically edge-labelled, that is, every (directed) edge carries a label such that no two edges starting at the same vertex have the same label, 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 $L_{\text{SAW},o}$. Assume that the group of label-preserving graph automorphisms acts quasi-transitively. We show that $L_{\text{SAW},o}$ 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 $L_{\text{SAW},o}$ is multiple context-free if and only if the group is virtually free.

Keywords. Self avoiding walk, formal language, multiple context free language, Cayley graph, virtually free group

Mathematics Subject Classifications. 20F10, 68Q45, 05C25

1. Introduction

A *walk* in a graph is an alternating 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 SAW), if its vertices are pairwise different. This notion was introduced in 1953 by the chemist Flory [Flo53] as a model for long-chain polymer molecules and has since attracted considerable interest. A lot

^{*}F. Lehner was supported by FWF (Austrian Science Fund) projects P31889-N35 and J3850-N32

[†]C. Lindorfer was partially supported by FWF (Austrian Science Fund) projects P31889-N35 and DK W1230.

of research has focused on lattices, see for instance the monograph by Madras and Slade [MS96] and also the lecture notes by Bauerschmidt et al. [BDCGS12], but other graphs have also been investigated, see for example the survey of Grimmett and Li [GL19].

One of the most fundamental problems related to this model is determining (exactly or asymptotically) the number of SAWs of a given length. Denote by $c_n(o)$ the number of SAWs with n edges starting at a fixed root vertex o . Hammersley [Ham57] showed that the limit

$$\mu(G) = \lim_{n \rightarrow \infty} c_n(o)^{1/n}$$

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

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

Explicit computation of $\mu(G)$ can be a very challenging task, even in seemingly harmless instances such as two-dimensional lattices. 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 [DCS12] by Duminil-Copin and Smirnov containing the first rigorous calculation of the connective constant of the hexagonal lattice is considered a milestone in the theory.

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: 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 [AJ90] 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 for graphs exhibiting some kind of large scale tree structure, see for instance [GM17] and [LW20].

Our two main results, Theorems 1.1 and 1.2 below, crucially depend on the rich interplay between tree like structures, context free languages, and the corresponding generating functions. Introduced in seminal work by Chomsky and Schützenberger [CS63], the idea to study context free languages through their generating functions is a useful tool in the investigation of context free languages [CSW02, Fla87, SS78] and related structures [FS09, Vie85]; the introduction of [BD15] provides a good overview.

In the present paper, we use formal languages to study SAWs on graphs all of whose ends are thin. Recall that 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*.

Our first main result concerns the SAW-generating function of graphs all of whose ends are thin.

Theorem 1.1. *Let G be a simple, locally finite, connected, quasi-transitive graph having only thin ends and let $o \in V(G)$. Then $F_{\text{SAW},o}$ 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. For this purpose consider the following setup. Let G be deterministically edge-labelled, that is, every (directed) edge e of G is assigned a label $\ell(e)$ from some given alphabet Σ 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 $\text{AUT}(G, \ell)$ of automorphisms of G which preserve ℓ acts with finitely many orbits on G . Important examples of transitive deterministically edge-labelled graphs are the Cayley graphs of finitely generated groups.

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

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

In this way, any set \mathcal{P} of walks gives rise to a language $L(\mathcal{P}) = \{\ell(p) : p \in \mathcal{P}\}$ 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. [SMFK91] as a generalisation of context free languages capable of modelling cross-serial dependencies which occur in some natural languages such as Swiss German. A concise definition of MCFLs will be given in Section 2; for now we only mention that they share many useful traits with context-free languages, including polynomial time parsability, semi-linearity and the closure properties of being a full AFL.

While MCFLs may seem artificial at first, they appear in some natural problems. One of them is the well known word problem, which is equivalent to recognising closed walks in a given Cayley graph. Anisimov [An71] showed that the arising language is regular if and only if the underlying group is finite. Muller and Schupp [MS83] showed that it is context-free if and only if the group is virtually free; they also introduced the class of context free graphs [MS85], which are precisely the edge-labelled graphs for which the language of words corresponding to closed walks is context free. It follows from the above results that 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 [Sal15] for $d = 2$, and this result has since been extended by Ho [Ho18] to all positive integers d . We note that any multiple context free grammar generating these languages must be ambiguous because the corresponding generating functions are not algebraic.

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

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

where $\mathcal{P}_{\text{SAW},o}$ is the set of all SAWs of length at least 1 on G starting at o . In his computation of $F_{\text{SAW},o}$ for the infinite ladder graph Zeilberger [Zei96] implicitly used that $L_{\text{SAW},o}$ for this graph is context free. More generally, Lindorfer and Woess [LW20] showed that $L_{\text{SAW},o}$ 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 $F_{\text{SAW},o}$ can be computed using an appropriate grammar generating $L_{\text{SAW},o}$. Our second main result generalises and completes these results.

Theorem 1.2. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph and let $o \in V(G)$. Then $L_{\text{SAW},o}$ 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 $L_{\text{SAW},o}$ lies in this hierarchy.

Theorem 1.3. *Let G be a simple, locally finite, connected, deterministically edge-labelled quasi-transitive graph and let $o \in V(G)$. Then $L_{\text{SAW},o}$ 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 1.2 implies that the property of having a multiple context-free language of self-avoiding walks is a group invariant and gives a characterisation of virtually free groups in terms of self-avoiding walks on their Cayley graphs; for various other characterisations of virtually free groups see for instance the survey [Ant11].

Corollary 1.4. *The language of self-avoiding walks on some (\iff any) Cayley graph of a group is multiple context-free if and only if the group is virtually free.*

As mentioned above, we follow a similar approach as Alm and Janson in [AJ90]. 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 3. Roughly speaking these are decompositions of a graph into parts which intersect in a tree-like manner. This notion was introduced by Halin [Hal76] in 1976; later it was rediscovered by Robertson and Seymour [RS84] 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 [DK15], inspired by a similar construction based on edge cuts introduced by Dunwoody in [Dun82].

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 4. 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 configurations in a feasible way.

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 5 we use this recursive structure to show that the set of bounded consistent configurations is in bijection with a context free language; Theorem 1.1 follows from this fact. In Section 6, again using the recursive structure, we show that $L_{\text{SAW},o}$ is an

MCFL, thus proving the first half of Theorem 1.3. Finally, we combine techniques from [LW20] with a result from [LL21] to complete the proof of Theorem 1.3.

2. Basic background

Throughout this paper, we denote by \mathbb{N} the set of natural numbers starting at 1 and by $[n]$ the set $\{k \in \mathbb{N} : k \leq n\}$.

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* $e^+ \in V(G)$. 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 direction. In other words, for $e \in E(G)$ the edge $\bar{e} \in E$ satisfies $\bar{e}^- = e^+$ and $\bar{e}^+ = e^-$ and $\bar{\bar{e}} = e$. When drawing a graph, every pair (e, \bar{e}) will usually be represented by and thought of as one undirected edge. 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 degrees.

A *walk* in a graph is an alternating sequence $p = (v_0, e_1, v_1, \dots, e_n, v_n)$ of vertices $v_i \in V(G)$ and edges $e_i \in E(G)$ 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. For two vertices v and w of p we write vpw for the maximal sub-walk of p starting at v and ending at w . If $v = v_0$ or $w = v_n$ we omit the corresponding vertex and denote the sub-walk by pw or vp , respectively. We extend this notation even further and denote for walks p_1, \dots, p_n and vertices v_0, \dots, v_n in the respective walks the concatenation $(v_0p_1v_1)(v_1p_2v_2) \dots (v_{n-1}p_nv_n)$ of the sub-walks $v_{i-1}p_i v_i$ by $v_0p_1v_1p_2 \dots p_nv_n$. 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 closed walk p of length at least 3 such that v_1p is self-avoiding is called *cycle*.

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 each of the walks p_i is a sequence starting and ending with a vertex, so that the final vertex of p_i and the initial vertex of p_{i+1} will appear next to each other in the sequence p . In

fact every appearance of two consecutive vertices in a multi-walk marks the start of a new walk component.

For a (multi-)walk p on G and $A \subset V(G) \cup E(G)$ 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 sub-graph 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 sub-graph H of G .

A graph G is *connected* if any two vertices v, w are the endpoints of some walk p in G . *Components* of G are maximal connected sub-graphs.

A *tree* is a connected and cycle-free graph. A *rooted tree* is a tree in which 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 which is a neighbour of v is called its *parent* and denoted by v^\dagger , 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 none of whose children are ancestors 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 in which an ordering is specified for the children of each vertex; in this case we will denote the i -th child of a vertex v with respect to this order by v_i^\dagger .

A tree consisting only of vertices of degree at most 2 is called 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 SAW. Given two disjoint subsets A and B of vertices of a graph G , an A - B -*path* on G is a sub-graph 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 sub-graph obtained from G by removing K and all edges incident to K . If removing K disconnects G , then K is called a *separating set*. Furthermore, we denote by $G[K]$ the sub-graph of G induced by K , that is the graph $G - (V(G) \setminus K)$.

Two rays in a graph G are called *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. We say that two ends ω_1 and ω_2 of a graph G are *separated* by K if any two rays $R_1 \in \omega_1$ and $R_2 \in \omega_2$ end up in different components of $G - K$. Halin [Hal65] 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 and lies 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 $V(G)$ preserving the neighbourhood relation in G . The set of all automorphisms of G forms a group which is called the *automorphism group* of G and denoted by $\text{AUT}(G)$. For a subgroup $\Gamma \subseteq \text{AUT}(G)$ we can define an equivalence relation on $V(G)$ by $u \sim v \iff \exists \gamma \in \Gamma: u = \gamma v$. 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*. A sub-graph H of G is called γ -invariant

for $\gamma \in \text{AUT}(G)$ if $\gamma(H) = H$.

It is well known, that any infinite, locally finite, connected graph which is quasi-transitive has either one, two, or infinitely many ends. 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 it must have thin ends. These and many more results were given by Halin in [Hal73].

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

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

Halin [Hal73] showed that any graph automorphism is either elliptic, or 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 (\iff every) vertex v of G the sequence $v, \gamma v, \gamma^2 v, \dots$ 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, which are invariant under γ^n for some $n \in \mathbb{N}$. Finally, if γ is parabolic then the unique end fixed by γ is thick and G contains infinitely many double rays invariant under γ^n for some $n \in \mathbb{N}$.

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

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

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 $\psi : Y \rightarrow X$, which is a quasi-isometry such that $\psi\varphi$ and $\varphi\psi$ are at bounded distance from the respective identity mappings. 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 is bijective on the set of ends, and maps thick ends to thick ends and thin ends to thin ends, see for example [Woe00, Lemma 21.4].

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 Σ , 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 $\text{AUT}(G)$. To this end, we denote by $\text{AUT}(G, \ell)$ the group of *label-preserving* graph automorphisms; when speaking of a quasi-transitive edge-labelled graph it will be implicitly assumed that $\text{AUT}(G, \ell)$ acts quasi-transitively. Note that in the case of a deterministic labelling ℓ , the group $\text{AUT}(G, \ell)$ acts freely on G , that is, the identity in $\text{AUT}(G, \ell)$ 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

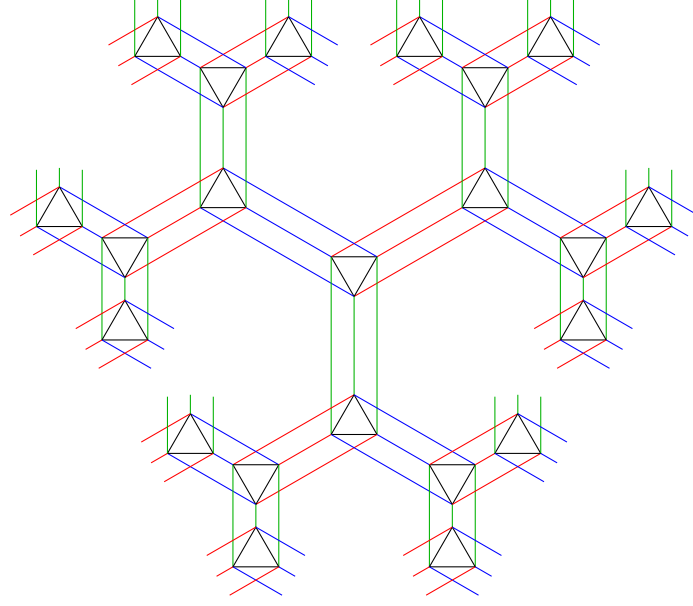


Figure 2.1: Cayley graph of the group $(C_2 * C_2 * C_2) \times C_3$. Different edge colours correspond to different generators.

with a symmetric generating set S of a group Γ , the *Cayley graph* $G = \text{Cay}(\Gamma, S)$ has vertex set $V(G) = \Gamma$. We choose $\Sigma = S$, 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 $\Gamma = \text{AUT}(G, \ell)$.

We finish this section by giving a simple example of a Cayley graph which will be 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 2.1 shows the Cayley graph G of Γ with respect to the symmetric generating set $S = \{a, b, c, r, r^{-1}\}$.

2.2. Formal languages

For an *alphabet* (finite set of letters) Σ we denote by

$$\Sigma^* = \{w = a_1 a_2 \dots a_n \mid n \geq 0, a_i \in \Sigma\}$$

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

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

$$\ell(p) = \ell(e_1)\ell(e_2)\dots\ell(e_n) \in \Sigma^*.$$

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

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

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

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

For two strings $\alpha, \beta \in (\mathbf{N} \cup \Sigma)^*$, we say that β is obtained from α in a *single step of leftmost derivation*, and write $\alpha \Rightarrow \beta$, if $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 \gamma \alpha_2$ for some $\alpha_1 \in \Sigma^*$, $\alpha_2 \in (\mathbf{N} \cup \Sigma)^*$ and additionally $A \vdash \gamma \in \mathbf{P}$. Thus, β is a result of using the rule $A \vdash \gamma$ to replace the leftmost non-terminal in α . A *leftmost derivation* of β from α is a sequence

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

such that $\alpha_{i-1} \Rightarrow \alpha_i$ for every i . We say that β is derived from α and write $\alpha \xRightarrow{*} \beta$. Each non-terminal $A \in \mathbf{N}$ generates a language $L_A = \{\alpha \mid A \xRightarrow{*} \alpha\}$ and the *language generated by the grammar \mathbf{G}* is $L(\mathbf{G}) = L_S$. The grammar is called *unambiguous* if for every $\alpha \in L(\mathbf{G})$ there is a unique leftmost derivation generating α .

For a given context-free grammar \mathbf{G} a *derivation tree* is an ordered tree D together with a labelling $\lambda : V(D) \rightarrow \mathbf{N} \cup \Sigma^*$ such that

- (i) internal vertices have labels in \mathbf{N} ,
- (ii) leaves have labels in Σ^* and
- (iii) whenever v_1, \dots, v_k are the ordered children of u in D ,

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

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

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

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

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 .

It follows from a famous result of Chomsky and Schützenberger [CS63] 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 it is the solution to a system of algebraic equations

over \mathbb{Q} . Elimination theory provides us with an irreducible polynomial P in $m + 1$ variables with coefficients in \mathbb{Q} such that

$$P(F_L(a_1, \dots, a_m), a_1, \dots, a_m) = 0.$$

A proof of this fact can for example be found in the book [KS86] of Kuich and Salomaa.

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 $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, S)$. 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 $A \vdash x_0 A_1 x_1 \dots A_n x_n$ tells us that we can obtain an element of the language L_A generated by A by sticking together the strings $x_0, \dots, x_n \in \Sigma^*$ 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 $w \in \Sigma^*$ has property $A \in \mathbf{N}$ and write $\vdash_{\mathbf{G}} A(w)$ if $w \in L_A$. Then the rule $A \vdash x_0 A_1 x_1 \dots A_n x_n$ is equivalent to the statement

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

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

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

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. Just like in the definition of context-free languages, the first two ingredients are an alphabet Σ and a set of non-terminals \mathbf{N} . As noted above we can intuitively view every non-terminal in a context-free language as a property applying to all strings it generates. Similarly, a non-terminal of a multiple context-free language should be viewed as a property applying to tuples of strings. To this end, 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 Σ are expressions of the form

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

where

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

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 $w_1, \dots, w_r \in \Sigma^*$ 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 $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, S)$, where \mathbf{N} is a finite ranked set of non-terminals, Σ is a finite alphabet, \mathbf{P} is a finite set of production rules over (\mathbf{N}, Σ) 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} and we write $\vdash_{\mathbf{G}} \tau$ 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 $L(\mathbf{G}) = \{w \in \Sigma^* \mid \vdash_{\mathbf{G}} S(w)\}$. We call a language *k-multiple context-free* or an *k-MCFL* if it is generated by an *k-MCFG*.

The following simple example of a 2-MCFG should be beneficial for a better understanding of the concepts above.

Example 2.1. Consider the MCFG $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, S)$, where $\mathbf{N} = \{S, A\}$, $\Sigma = \{a, b, c\}$ and the set \mathbf{P} consists of the rules ρ_1, \dots, ρ_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. 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 $\vdash_{\mathbf{G}} A(\epsilon, \epsilon)$. Applying the rule $A(a z_1, z_2) \leftarrow A(z_1, z_2)$ 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, iterative 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 $S(z_1 z_2) \leftarrow A(z_1, z_2)$ 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 $\vdash_{\mathbf{G}} S(w_1 w_2)$. 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. 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 $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, S)$ 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 $w_1, \dots, w_n \in \Sigma^*$ 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}, Σ) . The resulting 1-MCFG $\mathbf{G}' = (\mathbf{N}, \Sigma, \mathbf{P}', S)$ then generates the same language $L(\mathbf{G})$.

Remark 2.3. 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 homomorphisms, inverse homomorphisms, union, intersection with regular languages and Kleene closure. Furthermore they are parsable in polynomial time and semilinear.

Derivation trees for MCFLs were first defined by Seki et al [SMFK91]; we use a slight variation of their definition. Let $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, S)$ 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:

- (i) The root of D has $n \geq 0$ children and is labelled with a rule $\rho \in \mathbf{P}$.
- (ii) 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 .
- (iii) The rule ρ applied to the sequence $(\tau_i)_{i \in [n]}$ yields τ .

It is not hard to see that $\vdash A(w_1, \dots, w_r)$ if and only if there is a derivation tree D of the term $A(w_1, \dots, w_r)$. 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 an unambiguous MCFG. We denote by $w(D)$ the tuple of strings w_1, \dots, w_r generated by 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.

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 2.4 ([SMFK91, Lemma 3.2]). *For every infinite k -MCFL L there is some $w \in L$, which can be written in the form $w = x_1y_1x_2y_2 \dots x_{2k}y_{2k}x_{2k+1}$ for some $x_i, y_i, \in \Sigma^*$ such that*

- $y_1y_2 \dots y_{2k} \neq \epsilon$ and
- $x_1y_1^n x_2y_2^n \dots x_{2k}y_{2k}^n x_{2k+1} \in L$ 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. In particular it is not strong enough to imply the second part of the following result from [LL21], which is a main pillar of Theorem 1.3.

Theorem 2.5 ([LL21, Theorem 1.1]). *The language $\{a_1^{n_1} a_2^{n_2} \dots a_k^{n_k} \mid n_1 \geq n_2 \geq \dots \geq n_k \geq 0\}$ is a $\lceil k/2 \rceil$ -MCFL, but not a $(\lceil k/2 \rceil - 1)$ -MCFL.*

3. Tree decompositions

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

- (T1) $V(G) = \bigcup_{t \in V(T)} \mathcal{V}(t)$.
- (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 with 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 $\Gamma \leq \text{AUT}(G)$, 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.

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. Stallings'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 [TW93] showed that accessibility of a group is equivalent to accessibility of some (and thus all) of its Cayley graphs. The following theorem is closely related to the work of Dunwoody [Dun82] and is a direct consequence of [HLMR22, Theorem 6.4].

Theorem 3.1. *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 3.1 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 3.2. *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 (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 2.1. 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 3.1.

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

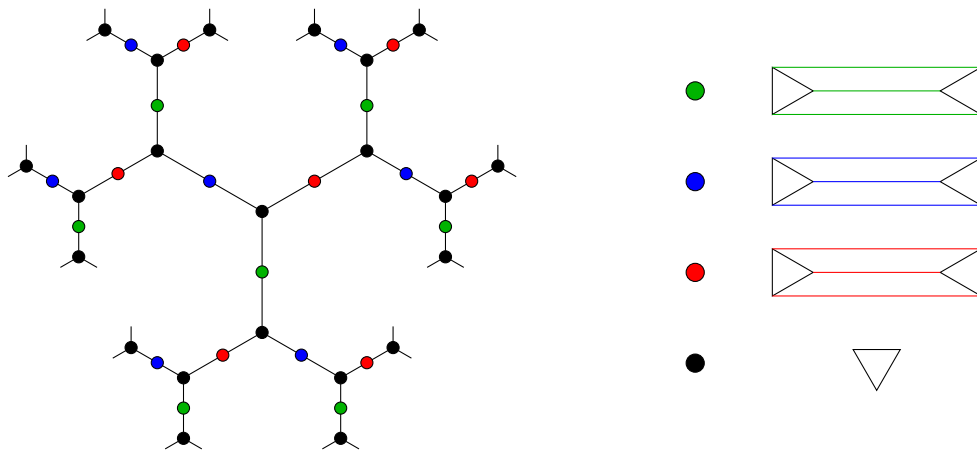


Figure 3.1: Decomposition tree T of the Cayley graph G . The vertex-colouring indicates the four orbits of the group action on the vertices of T . The subgraphs of G induced by the different parts are shown at the right side.

- (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.

The remainder of this section is dedicated to the proof of the four properties stated above.

Proof of (P1)–(P4). 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 (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 $(\bigcup_{t \in V(T_1)} \mathcal{V}(t), \bigcup_{t \in V(T_2)} \mathcal{V}(t))$ 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 3.3. *Let $\mathcal{T} = (T, \mathcal{V})$ be a tree decomposition as in Corollary 3.2. 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 and e_2 share a 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. \square

The basis for the proof of (P3) is the following lemma, which is closely related to [HLMR22, Proposition 4.5].

Lemma 3.4. *Let G, Γ and (T, \mathcal{V}) be as in Theorem 3.1. Then for every vertex t of T , the induced sub-graph $G[\mathcal{V}(t)]$ 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 $\mathcal{V}(t)$, which lie in adhesion sets of the tree decomposition. We conclude that $\Gamma_{\mathcal{V}(t)}$ acts with finitely many orbits on $\mathcal{V}(t)$. \square

Our goal is to apply the following proposition due to Thomassen [Tho92] to a part $\mathcal{V}(t)$, but in general $G[\mathcal{V}(t)]$ need not be connected, so some additional work is necessary.

Proposition 3.5 ([Tho92, 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 sub-graph of G obtained from the induced sub-graph $G[\mathcal{V}(t)]$ 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 3.5 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}) .

This completes the proof of (P1)–(P4) □

3.1. Rooted tree decompositions

Let (G, o) be a simple, locally finite, connected graph rooted at $o \in V(G)$. 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.

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 $\mathcal{E} : V(T) \rightarrow 2^E(G)$ by $\mathcal{E}(r) = E(G[\mathcal{V}(r)])$ and

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

where t^\dagger 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 sub-graph $T[S]$ 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 $E(G) = \bigsqcup_{t \in V(T)} \mathcal{E}(t)$.

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

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

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,

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

Again, we visualise these concepts using the Cayley graph G from Figure 2.1 and its tree decomposition (T, \mathcal{V}) shown in Figure 3.1. 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 3.2 shows a portion of the decomposition tree T and the t -graphs for vertices t contained in it. Compare this to Figure 3.1 and note that the t -graphs on the parts are generally neither sub-graphs (due to virtual edges) nor super-graphs (due to some missing non-virtual edges) of the induced graphs on the parts.

3.2. Cones and cone types

Recall that 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 $\Gamma \subseteq \text{AUT}(G)$ 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^\dagger of s to the parent t^\dagger 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.

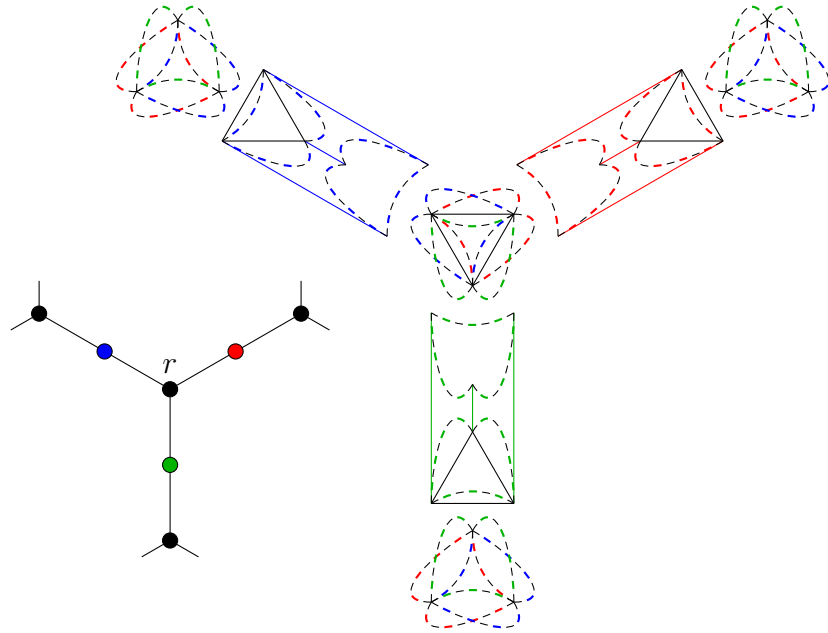


Figure 3.2: Decomposition tree T of the Cayley graph G and its corresponding t -graphs. Dashed edges are virtual e -edges; if they are shared by different t -graphs, they have the same shape in these graphs.

Lemma 3.6. 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 $\mathcal{G}(t)$.

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 $\mathcal{G}(t)$. \square

Lemma 3.7. Let (T, \mathcal{V}, r) be a reduced, rooted tree decomposition of a locally finite graph G . If there is a subgroup $\Gamma \leq \text{AUT}(G)$ 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 onto t^\uparrow 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. \square

4. 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 $s = r$. If $X(s) \neq s$, then $P(s)$ ends in $\mathcal{V}(s, X(s))$. Moreover, if $X(s) = s$ then the last edge of $P(s)$ is 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 3.6, 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 be able 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. 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 $C(t^\dagger)$ and $C(t)$ are compatible whenever both t and t^\dagger 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} will be denoted by $\mathcal{C}_{\mathcal{T}}$.

Remark 4.1. 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 2.1. Using the decomposition tree T and the t -graphs from Figure 3.2, an example of a bounded consistent configuration (W, X) on T is shown in Figure 4.1. 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 4.2. *Let $s, t \in V(T)$ such that $s = t^\dagger$ 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 which is 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 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. \square

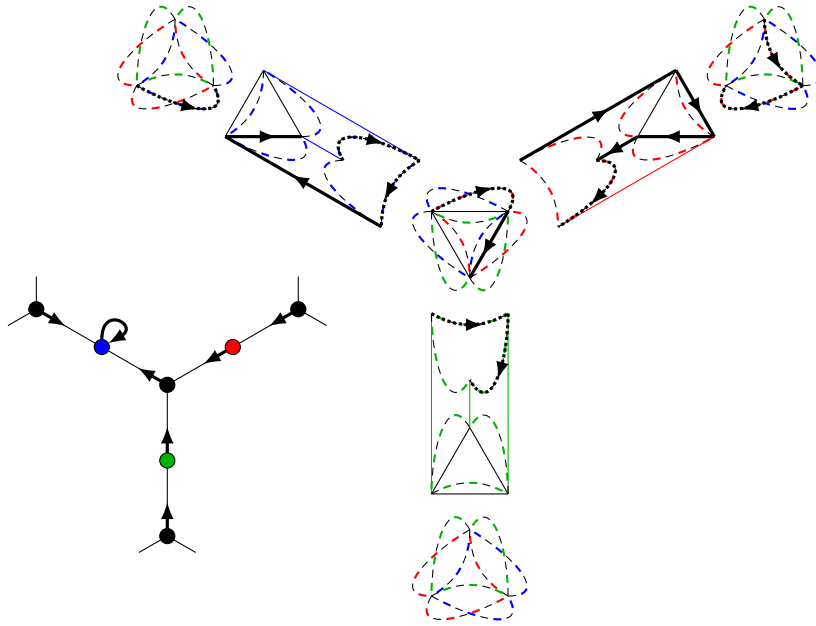


Figure 4.1: A bounded consistent configuration (W, X) on the tree decomposition \mathcal{T} of the Cayley graph G . Edges of the walk $W(t)$ on the t -graph $\mathcal{G}(t)$ are drawn bold and decorated with arrows according to their direction. Exit directions of vertices of T are also denoted by arrows pointing from a vertex t to a vertex s if $X(t) = s$.

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 will be denoted $\psi_r(\mathcal{C})$.

In Section 3 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 since we would like 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 $f^- = (f^+)^{\dagger}$ (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

$$P(t_0)v_1P(t_1)v_2 \dots v_lP(t_l).$$

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 4.2.

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

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

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

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

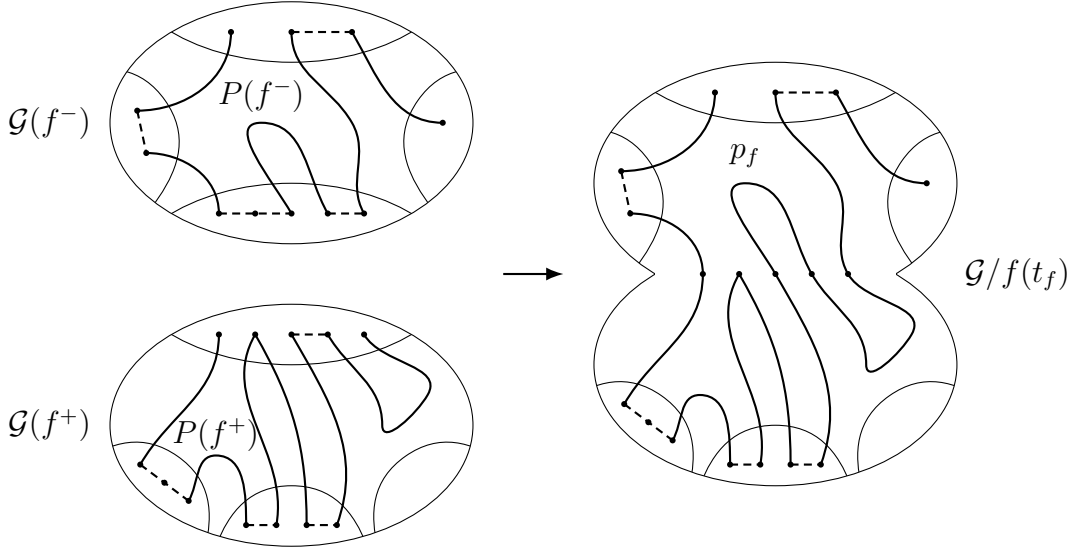


Figure 4.2: Combining walks $P(f^-)$ and $P(f^+)$ of compatible configurations on the endpoints of f into a walk p_f on $\mathcal{G}/f(t_f)$.

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

Lemma 4.3. *The walk p_f is a self-avoiding walk on $\mathcal{G}/f(t_f)$ satisfying*

$$p_f \cap \mathcal{G}(f^-) = P(f^-) - \mathcal{E}(f) \quad \text{and} \quad p_f \cap \mathcal{G}(f^+) = P(f^+) - \mathcal{E}(f).$$

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. \square

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

Lemma 4.4. *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 $X(t) = t^\dagger$, so by the above observation $X/f(t) = t^{\dagger/f}$. 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 $\mathcal{V}(t^\dagger, t) = \mathcal{V}/f(t^{\dagger/f}, t)$, 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 $f^- = (f^+)^\dagger$. 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 $X(f^-) = (f^-)^\dagger \notin \{f^-, f^+\}$, and thus $X/f(t_f) = X(f^-) = (t_f)^{\dagger/f}$, showing that $C/f(t_f)$ is the empty configuration.

So we may assume that p_f is non-empty. By Lemma 4.3, 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 $\mathcal{V}((f^-)^\dagger, f^-) = \mathcal{V}/f((t_f)^{\dagger/f}, t_f)$, 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 $\mathcal{V}/f(s, t_f) = \mathcal{V}(s, f^-) \subseteq V(\mathcal{G}(f^-))$. Consequently, Lemma 4.3 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 4.3 we thus have $u p_f v \cap \mathcal{E}/f(s, t_f) = u P(f^-) v \cap \mathcal{E}(s, f^-)$ 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 4.5. *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 4.3, 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, \dots, 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 $P(f^-) = q_1 e_1 q_2 \dots e_{l-1} q_l$, 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 $C/f = C'$. \square

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 3. If the set F induces a connected sub-graph of T , then there is a unique contracted vertex in \mathcal{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 $\iota: T_1 \rightarrow T_2$ 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 4.6. *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} 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 .

For double contractions as considered in this lemma, however, 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 $\mathcal{T}/f_1/f_2 = \mathcal{T}/f_2/f_1$ now 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 by definition $P/f_1/f_2(t) = P(t) = P/f_2/f_1(t)$, 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 4.3 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 4.3, 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 4.3, 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 4.4, both $C/f_1/f_2$ and $C/f_2/f_1$ are bounded consistent configurations, thus by Remark 4.1 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 . \square

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 4.7 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,

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

Let $S \subseteq K_t$ consist of the vertex t and all vertices of K_t carrying non-boring configurations. Note that Lemma 4.2 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 $\mathcal{E}/F(t_F^\dagger, t_F) = \mathcal{E}(t^\dagger, t)$, 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 4.1. In Figure 4.3 we iteratively contract edges incident to the root vertex r until only a

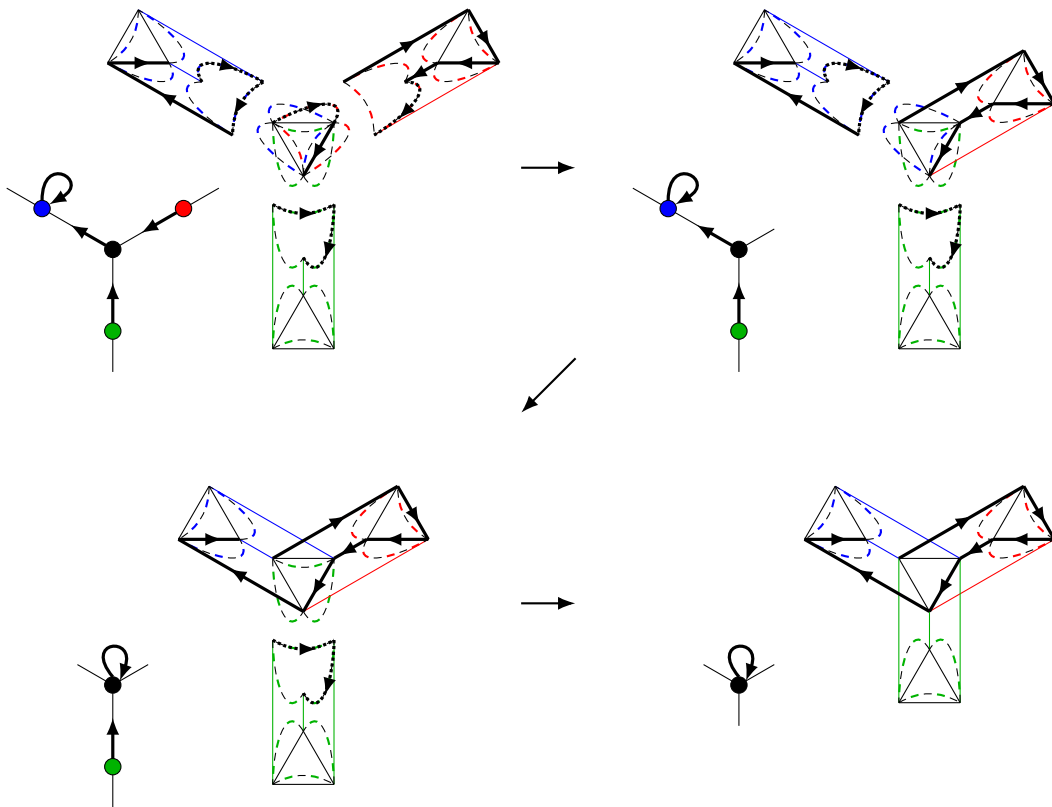


Figure 4.3: Contraction of the bounded consistent configuration C on T . The walk $\psi_r(t)$ is reached after 2 steps. It contains only non-virtual edges and does not change anymore during the third contraction.

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 4.7. *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 4.4, 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 4.2 implies that $C_1/F = C_2/F$. Inductive application of Lemma 4.5 yields $C_1 = C_2$, so ψ_r is injective.

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 4.2 provides a bounded consistent configuration $C \in \mathcal{C}_{\mathcal{T}/F}$ such that $C(t_F) = (p, t_F)$. Lemma 4.5 yields a configuration $C \in \mathcal{C}_{\mathcal{T}}$ with $\psi_r(C) = P$ and therefore ψ_r is surjective. \square

Remark 4.8. 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^\dagger .

5. A grammar for bounded consistent configurations

Throughout this section let (G, o) be a rooted, simple, locally finite, connected graph all of whose ends are thin, and let Γ be a group acting quasi-transitively on G . Thomassen and Woess [TW93] showed that any transitive graph without thick ends is accessible. Using the fact that any quasi-transitive graph is quasi-isometric to some transitive graph, and additionally that quasi-isometries preserve accessibility, we obtain that the graph G is accessible.

By Corollary 3.2 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 3 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 Σ 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.

5.1. Choosing representatives of configurations

We would like to define a function ρ which assigns 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 5.1. *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 $t^\dagger \in R$ fix some arbitrary automorphism $\delta_{t^\dagger} \in \Gamma$ mapping K_{t^\dagger} to $K_{\rho(t^\dagger)}$. Finally for all other vertices t of T inductively define

$$\delta_t = \delta_{\delta_{t^\dagger}(t)} \circ \delta_{t^\dagger}. \tag{5.1}$$

This is well defined because $\delta_{t^\dagger}(t)$ must be a child of a vertex of R for every t ; in particular, if $\delta_{t^\dagger}(t) \in R$, then $\delta_{\delta_{t^\dagger}(t)} = 1_\Gamma$. Note that equation (5.1) 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 $t_1^\downarrow, \dots, t_{k(t)}^\downarrow$ be the $k(t)$ children of t ordered in a way such that

$$\delta_t(t_i^\downarrow) = \delta_t(t)_i^\downarrow = \rho(t)_i^\downarrow. \tag{5.2}$$

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 (5.1) of δ_t it is immediate that

$$\delta_{t_i^\downarrow} = \delta_{\delta_t(t_i^\downarrow)} \circ \delta_t. \tag{5.3}$$

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

$$\rho(t_i^\downarrow) = \delta_{t_i^\downarrow}(t_i^\downarrow) = \delta_{\delta_t(t_i^\downarrow)}(\delta_t(t_i^\downarrow)) = \delta_{\rho(t)_i^\downarrow}(\rho(t)_i^\downarrow) = \rho(\rho(t)_i^\downarrow). \tag{5.4}$$

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 (5.3) and (5.2).

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 (5.4), 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 :

$$\rho(C(t_i^\downarrow)) = \delta_{t_i^\downarrow}(C(t_i^\downarrow)) = \delta_{\delta_t(t_i^\downarrow)}(\delta_t(C(t_i^\downarrow))) = \rho(\delta_t(C(t_i^\downarrow))) = \rho(\delta_t \circ C \circ \delta_t^{-1}(\rho(t)_i^\downarrow)). \tag{5.5}$$

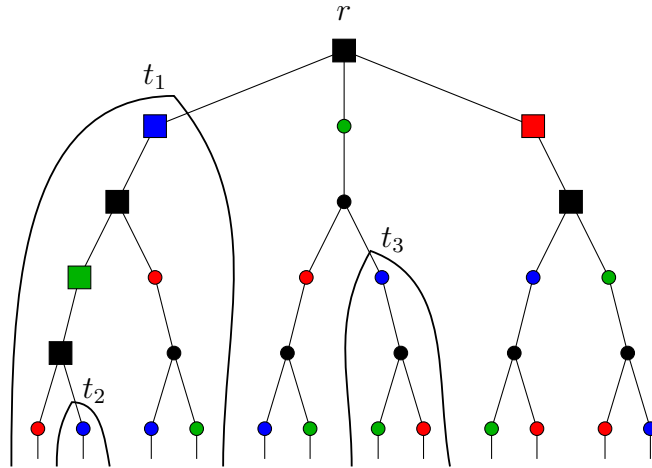


Figure 5.1: Cone types of the tree decomposition T . The square nodes give one possible valid choice for the set R . The marked cones rooted at t_1 , t_2 and t_3 have the same type.

Similarly to (5.4), the first and second equalities in (5.5) follow from the definition of ρ and equation (5.3), respectively. For the third equality, we use the definition of ρ and the fact that $\delta_t(C(t_i^\dagger))$ is a configuration on $\delta_t(t_i^\dagger)$. The last equality follows from (5.2).

Let us illustrate the above definitions using the tree decomposition (T, \mathcal{V}) shown in Figure 3.1. 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 3.7). Figure 5.1 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 5.1. 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 to its representative t_1 in R and set $\delta_{t_2} = \gamma$. Regarding t_3 , note that the representative of t_3^\dagger in R is t_2^\dagger , thus by definition $\delta_{t_3^\dagger}$ maps the cone $K_{t_3^\dagger}$ onto $K_{t_2^\dagger}$ and thus t_3 onto t_2 . The map δ_{t_3} is obtained by first applying $\delta_{t_3^\dagger}$ (which maps t_3 to t_2), and then applying $\delta_{\delta_{t_3^\dagger}} = \delta_{t_2}$ (which maps t_2 to t_1).

5.2. 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 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, 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 $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, A_r)$ generating our desired language is defined as follows.

The alphabet is

$$\Sigma = \{a_c \mid c \text{ configuration on some } t \in R\}.$$

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

$$A_{t,C(t)}(a_{C(t)}x_1 \dots x_{k(t)}) \leftarrow \left(A_{\rho(t_i^\downarrow), \rho(C(t_i^\downarrow))}(x_i) \right)_{i \in [k(t)]}$$

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 5.2. 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 $v_1 (= d), v_2, \dots, v_n$ be the vertices of D in DFS-order and let $\lambda(v_i) = A_{t_i, c_i}$ be their labels. Then

$$w(D) = x_1 \dots x_n, \quad \text{where } x_i = \begin{cases} a_{c_i} & \text{if } c_i \text{ is non-boring} \\ \epsilon & \text{otherwise} \end{cases}$$

Observations (i) and (ii) are direct consequences of the structure of \mathbf{G} . For (iii) we 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 .

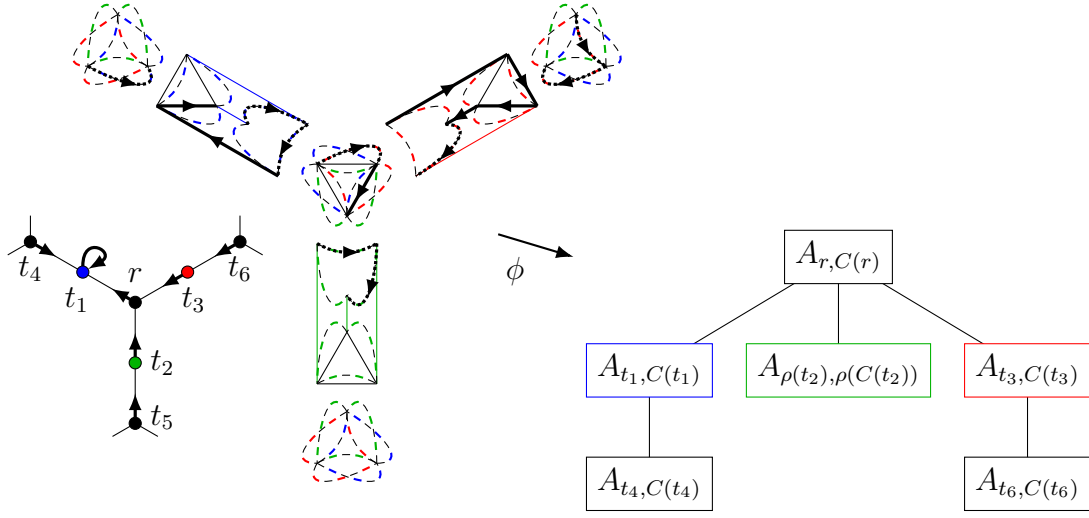


Figure 5.2: The map ϕ transforms the configuration C on the decomposition tree T into the derivation tree $\phi(C)$ over the grammar \mathbf{G} . The word generated by this derivation tree is $w(\phi(C)) = a_{C(r)}a_{C(t_1)}a_{C(t_4)}a_{\rho(C(t_2))}a_{C(t_3)}a_{C(t_6)}$.

Lemma 5.3. *The grammar \mathbf{G} is unambiguous 1-multiple context-free.*

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 5.2 (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 4.2 the configurations c and c' must be non-boring. Thus Remark 5.2 (iii) implies that $w(D) \neq w(D')$ and \mathbf{G} is unambiguous. \square

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 5.2 the derivation tree $\phi(C)$ of the configuration C given in Figure 4.1. Note that the vertex t_2 is not contained in the set R from Figure 5.1, 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 5.4. *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, \dots, t_k there is a rule in \mathbf{P} with head $\lambda(t)$ and tail $(\lambda(t_1), \dots, \lambda(t_k))$.

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 $s_1^\downarrow, \dots, s_{k(s)}^\downarrow$ of s in T , which are also the children of s in $T[S]$. Then the production

$$A_{\rho(s), \rho(C(s))}(a_{\rho(C(s))}x_1 \dots x_{k(s)}) \leftarrow \left(A_{\rho(s_i^\downarrow), \rho(C(s_i^\downarrow))}(x_i) \right)_{i \in [k(s)]}$$

is in \mathbf{P} because $\delta_s \circ C \circ \delta_s^{-1}$ is a consistent configuration on $\{\delta_s(s), \delta_s(s_i^\downarrow), \dots, \delta_s(s_{k(s)}^\downarrow)\}$ and (5.4) and (5.5) 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

$$\lambda(u) = A_{\rho(\iota(u)), \rho(C(\iota(u)))}. \tag{5.6}$$

We start the top-down construction by setting $\iota(d) = r$ and $C(r) = c$. Then clearly d satisfies (5.6). Suppose now $\iota(u) = t$ is already defined for a vertex u of D and that u satisfies (5.6). Let u_1, \dots, u_k be the $k > 0$ children of u in D and $\lambda(u_i) = A_{s_i, c_i}$ their labels. Then

$$A_{\rho(t), \rho(C(t))}(a_{\rho(C(t))}x_1 \dots x_k) \leftarrow (A_{s_i, c_i}(x_i))_{i \in [k]} \tag{5.7}$$

is a rule in \mathbf{P} . This implies that t has precisely k children $t_1^\downarrow, \dots, t_k^\downarrow$ 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 (5.7) 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 4.2 the configuration C can be (uniquely) extended to a bounded consistent configuration on T . Moreover it follows directly from (5.6) 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 4.2 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. \square

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 5.5. *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 5.5 with the connection between bounded consistent configurations and SAWs established in Theorem 4.7, we obtain the proof of our first main result, Theorem 1.1. Before we turn to the proof, let us recall the statement of the theorem.

Theorem 5.6. *Let G be a locally finite, connected, quasi-transitive graph having only thin ends and let $o \in V(G)$. Then $F_{\text{SAW},o}$ 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 4.7 the generating function of self-avoiding walks coincides with the generating function of the set $\mathcal{C}_{\mathcal{T}}$,

$$F_{\text{SAW},o}(z) = \sum_{C \in \mathcal{C}_{\mathcal{T}}} z^{\|C\|}.$$

Let $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, S)$ be the unambiguous 1-MCFG over the alphabet $\Sigma = \{a_{c_1}, \dots, a_{c_m}\}$ as defined at the start of Section 5.2, 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 $L(\mathbf{G})$. It is known (see for instance [KS86]) 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 5.5 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 $z^{\|c_i\|}$,

$$F_{\text{SAW},o}(z) = F_{L(\mathbf{G})}(z^{\|c_1\|}, \dots, z^{\|c_m\|})$$

In particular $F_{\text{SAW},o}(z)$ is algebraic over \mathbb{Q} ; it satisfies the equation

$$Q(F_{\text{SAW},o}(z), z) = 0,$$

where $Q(y, z) = P(y, z^{\|c_1\|}, \dots, z^{\|c_m\|})$ 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 $F_{\text{SAW},o}(z)$ and thus is an algebraic number. \square

6. 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 1.3. *Let G be a simple, locally finite, connected, deterministically edge-labelled quasi-transitive graph and let $o \in V(G)$. Then $L_{\text{SAW},o}$ 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 $L_{\text{SAW},o}$ 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 5. For the converse implication we essentially follow the approach of Lindorfer and Woess [LW20]; we note that this final part does not depend on the other results of this paper and can be read independently.

6.1. 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) which will be fixed throughout this section; we also fix a map ρ as constructed in Section 5. Our aim is proving the following theorem, a stronger version of one of the two implications of Theorem 1.3.

Theorem 6.1. *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 $L_{\text{SAW},o}$ is $\lceil k/2 \rceil$ -multiple context-free.*

If the edge-labelling is deterministic, then $L_{\text{SAW},o}$ is unambiguous $\lceil k/2 \rceil$ -multiple context-free.

To prove the above theorem, we give an MCFG $\mathbf{G} = (\mathbf{N}, \Sigma, \mathbf{P}, A_r)$ and show that it generates the language $L_{\text{SAW},o}$. As mentioned above, \mathbf{G} is a refinement of the 1-MCFG from Section 5.

Obviously, the alphabet Σ 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 5, 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 $p - \mathcal{E}(t^\dagger, t)$ containing at least two vertices of $\mathcal{V}(t^\dagger, t)$. Furthermore let $r(c) = \mu(c) + 1$ if $x \neq t^\dagger$ and the final walk component of $p - \mathcal{E}(t^\dagger, t)$ contains only a single vertex of $\mathcal{V}(t^\dagger, t)$ and let $r(c) = \mu(c)$ 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

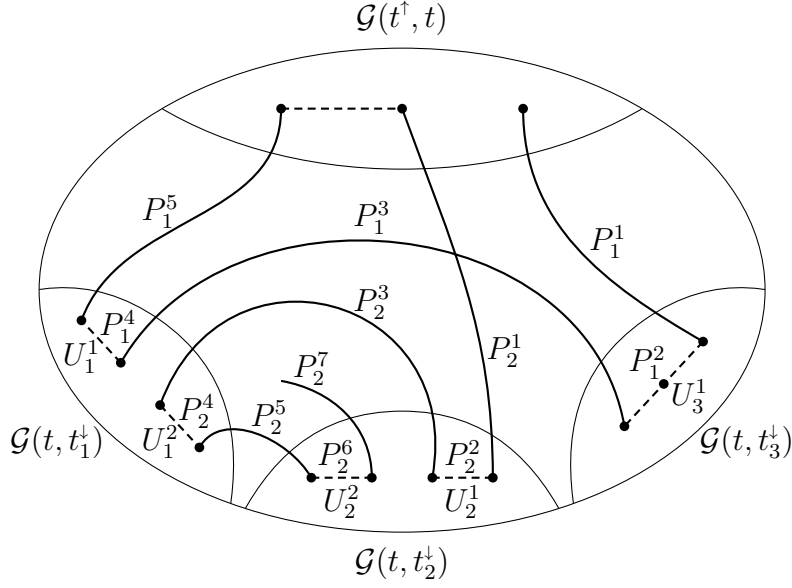


Figure 6.1: Decomposition of $P(t)$ into subwalks P_h^l . For even l , the walk P_h^l is equal to some U_i^j ; for instance $P_2^4 = U_1^2$ means that the fourth subwalk in the second walk component of $P(t) - \mathcal{E}(t^\dagger, t)$ coincides with the second walk component of $P(t) \cap \mathcal{G}(t, t_1^\dagger)$.

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 $\{t, t_1^\dagger, \dots, t_{k(t)}^\dagger\}$ such that $C(t)$ is non-boring. Let $P_1, \dots, P_{\mu(C(t))}$ be the walk components of $P(t) - \mathcal{E}(t^\dagger, t)$ containing at least 2 vertices of $\mathcal{V}(t^\dagger, t)$, and if $r(C(t)) > \mu(C(t))$ let $P_{r(C(t))}$ be the (possibly trivial) final walk component of $P(t) - \mathcal{E}(t^\dagger, t)$. Moreover, for each $i \in [k(t)]$ let $U_i^1, \dots, U_i^{\mu_i}$ be the non-trivial walk components of $P(t) \cap \mathcal{G}(t, t_i^\dagger)$, 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 $P(t) \cap \mathcal{G}(t, t_i^\dagger)$, that is, the notation P_h^l indicates which walk component of $P(t)$ the walk lies in, whereas the notation U_i^j tells us which adhesion set the virtual edges belong to, see Figure 6.1 for an example.

For $h < r(C(t))$, the string α_h corresponding to P_h is

$$\alpha_h = \ell(P_h^1)z(P_h^2)\ell(P_h^3)z(P_h^4)\dots\ell(P_h^{2m+1}),$$

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

$$\alpha_h = \ell(P_h^1)z(P_h^2)\ell(P_h^3)z(P_h^4)\dots\ell(P_h^{2m+1})\beta,$$

where $z(P_h^l) = z_{i,j}$ as above, and $\beta = \epsilon$ unless there is some $i \in [k(t)]$ such that $X(t) = t_i^\dagger$ and $P(t)$ does not end with an edge in $\mathcal{E}(t, t_i^\dagger)$, 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

$$A_{t,C(t)}(\alpha_1, \dots, \alpha_{r(C(t))}) \leftarrow \left(A_{\rho(t_i^\dagger), \rho(C(t_i^\dagger))}(z_{i,1}, \dots, z_{i,r(C(t_i^\dagger))}) \right)_{i \in [k(t)]} \tag{6.1}$$

where the strings $\alpha_1, \dots, \alpha_{r(C(t))}$ 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 $L_{\text{SAW},o}$, 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 $\ell(p) = (\ell(p_1), \dots, \ell(p_m))$ of the non-trivial walk components p_1, \dots, p_m of $p - \mathcal{E}(t^\dagger, t)$. 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 28. This is inductively taken care of by production rules of type (6.1): 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 4.6, 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 6.2. This replacement procedure is essentially captured by rules of type (6.1).

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 of $\psi_t(C) - \mathcal{E}(t^\dagger, t)$ come in two different flavours. Recall that the walk $\psi_t(C)$ starts in $\mathcal{V}(t^\dagger, t)$, and so does every walk component of $\psi_t(C) - \mathcal{E}(t^\dagger, t)$. Call a walk component a *U-walk* if it returns to $\mathcal{V}(t^\dagger, t)$, 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 $\psi_t(C) - \mathcal{E}(t^\dagger, t)$. It is not hard to see that the projection of any U-walk is a non-trivial walk

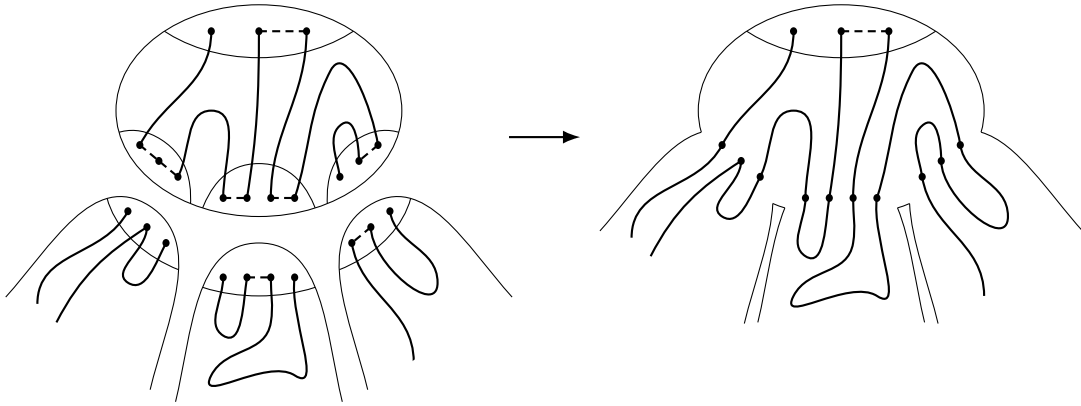


Figure 6.2: Iterated contraction shows that we can obtain $\psi_t(C)$ by combining walks $P(t)$ and $\psi_s(C)$ for all children s of t .

component of $P(t) - \mathcal{E}(t^\dagger, t)$ containing at least two vertices in $\mathcal{V}(t^\dagger, t)$. 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 $\psi_t(C) - \mathcal{E}(t^\dagger, t)$, and it is not hard to see that in this case $X(t) \neq t^\dagger$. 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^\dagger such that $\psi_{t_i^\dagger}(C) - \mathcal{E}(t, t_i^\dagger)$ contains an I-walk, then $\psi_t(C)$ ends in this I-walk. Note that this happens if and only if $X(t) = t_i^\dagger$ and the last edge of $P(t)$ is not contained in $\mathcal{E}(t, t_i^\dagger)$. 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 $P(t) \cap \mathcal{G}(t, t_i^\dagger)$. Adding $z_{i, \mu_i + 1}$ 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 6.2. *The grammar \mathbf{G} is $\lceil k/2 \rceil$ -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 (6.1) is a well-formed production rule over (\mathbf{N}, Σ) . Compatibility of $C(t)$ and $C(t_i^\dagger)$ implies that $\mu_i = \mu(C(t_i^\dagger))$ for every i . Additionally, if $X(t) = t_i^\dagger$ for some $i \in [k(t)]$ and $P(t)$ does not end with an edge in $\mathcal{E}(t, t_i^\dagger)$, then $X(t_i^\dagger) \neq t$ and the final walk component of $P(t_i^\dagger) - \mathcal{E}(t, t_i^\dagger)$ contains only a single vertex of $\mathcal{V}(t, t_i^\dagger)$, so $\mu_i + 1 = \mu(C(t_i^\dagger)) + 1 = r(C(t_i^\dagger))$. We conclude that every variable $z_{i,j}$ with $j \leq r(C(t_i^\dagger))$ 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 $P(t) - \mathcal{E}(t^\dagger, t)$. Then P_i contains at least two vertices of $\mathcal{V}(t^\dagger, t)$ for $i < r(C(t))$ and at least one vertex of $\mathcal{V}(t^\dagger, t)$ for $i = r(C(t))$.

The size of $\mathcal{V}(t^\dagger, t)$ 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. \square

While the grammar \mathbf{G} may appear more complicated than the 1-multiple context-free grammar of configurations introduced in Section 5, 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 will 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 5.4, the proof is exactly the same and is thus omitted.

Lemma 6.3. *The map ϕ is a bijection between the set \mathcal{C}_T of bounded consistent configurations on 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 $C \in \mathcal{C}_T$, 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. *Let $C \in \mathcal{C}_T$ be a bounded consistent configuration on T . Then*

$$\ell(\psi_r(C)) = w(\phi(C)).$$

Proof. Let $C \in \mathcal{C}_T$. 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 $C(t^\dagger)$ is non-boring it holds that

$$\ell(\psi_t(C)) = w(\phi_t(C)). \tag{6.2}$$

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 $\psi_t(C) - \mathcal{E}(t^\dagger, t)$. For $h \in [r]$, let Q_h denote the h -th walk component of $\psi_t(C) - \mathcal{E}(t^\dagger, t)$, and let $w_h = \ell(Q_h)$. By definition $\ell(\psi_t(C)) = (w_1, \dots, w_r)$. 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

$$w(\phi_{t_i^\downarrow}(C)) = \ell(\psi_{t_i^\downarrow}(C)) = (w_i^1, \dots, w_i^{r_i}),$$

in particular $r_i = r(C(t_i^\dagger))$. We examine the left-hand side and right-hand side of (6.2) 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 4.6 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^\dagger 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 $uv \in P(t) \cap \mathcal{E}(t, t_i^\dagger)$, the walk $\psi_{t_i^\dagger}(C)$ contains a sub-walk with the same endpoints consisting entirely of non-virtual edges. Replace every such virtual edge in $P(t)$ by the corresponding walk in $\psi_{t_i^\dagger}(C)$. If $X(t) = t_i^\dagger$, then append the sub-walk of $\psi_{t_i^\dagger}(C)$ after the last vertex in $\mathcal{V}(t, t_i^\dagger)$ to the resulting walk. Note that equivalently, we can let $U_i^1, \dots, U_i^{\mu_i}$ be the sequence of non-trivial walk components of $P(t) \cap \mathcal{G}(t, t_i^\dagger)$, replace every U_i^j by the respective Q_i^j , and append $Q_i^{r_i}$ in case $X(t) = t_i^\dagger$ and $P(t)$ does not end in a virtual edge in $\mathcal{E}(t, t_i^\dagger)$.

Let $P_1, \dots, P_{r(C(t))}$ be the walk components of $P(t) - \mathcal{E}(t^\dagger, t)$ and for every $h \in [r(C(t))]$ let $P_h = P_h^1 P_h^2 P_h^2 \dots P_h^{2m+1}$ 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

$$\ell(Q_h) = \ell(P_h^1) \tilde{\ell}(P_h^2) \ell(P_h^3) \tilde{\ell}(P_h^4) \dots \ell(P_h^{2m+1}),$$

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

$$\ell(Q_h) = \ell(P_h^1) \tilde{\ell}(P_h^2) \ell(P_h^3) \tilde{\ell}(P_h^4) \dots \ell(P_h^{2m+1}) \beta,$$

where $\tilde{\ell}(P_h^l) = w_i^j$ as above and $\beta = \epsilon$, unless $X(t) = t_i^\dagger$ and $P(t)$ does not end with an edge in $\mathcal{E}(t, t_i^\dagger)$, in which case $\beta = \ell(Q_i^{\mu_i+1}) = w_i^{\mu_i+1}$.

We now turn to the right-hand side of (6.2). For every $i \in [k(t)]$, the induction hypothesis implies that $\phi_{t_i^\dagger}(C)$ is a derivation tree of the term

$$\tau_i := A_{\rho(t_i^\dagger), \rho(C(t_i^\dagger))} (w_i^1, \dots, w_i^{r_i}).$$

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

$$A_{\rho(t), \rho(C(t))} (\alpha_1, \dots, \alpha_{r(C(t))}) \leftarrow \left(A_{\rho(t_i^\dagger), \rho(C(t_i^\dagger))} (z_{i,1}, \dots, z_{i,r(C(t_i^\dagger))}) \right)_{i \in [r(C(t))]}$$

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 6.1. Theorem 4.7 yields that the language of self-avoiding walks of the graph G satisfies

$$L_{\text{SAW},o} = \{\ell(\psi_r(C)) : C \in \mathcal{C}_{\mathcal{T}}\}.$$

Furthermore Lemma 6.3 implies that the language generated by the grammar \mathbf{G} is given by

$$L_{\mathbf{G}} = \{w(\phi(C)) : C \in \mathcal{C}_{\mathcal{T}}\}.$$

But by Lemma 6.4 these two sets are equal and \mathbf{G} is a $\lceil k/2 \rceil$ -multiple-context-free grammar generating $L_{\text{SAW},o}$.

Finally, if the edge-labelling of G is deterministic, then ℓ is a bijection between the set of self-avoiding walks on G and $L_{\text{SAW},o}$. Lemma 6.4 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

6.2. Multiple context-freeness implies bounded end size

In this section we prove the second part of our main result, namely

Theorem 6.5. *Let G be a simple, locally finite, connected, quasi-transitive deterministically edge-labelled graph such that $L_{\text{SAW},o}$ 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 of Lindorfer and Woess [LW20].

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 will 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 $L_{\text{SAW},o}(G)$ is multiple context-free, then there are always non-elliptic automorphisms. The following lemma extends [LW20, Lemma 4.3] to multiple context-free languages; the proof is essentially the same.

Lemma 6.6. *Let G be a connected, infinite, locally finite and deterministically edge-labelled graph and let $\Gamma \leq \text{AUT}(G, \ell)$ act quasi-transitively on G . If $L_{\text{SAW},o}$ 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 $L_{\text{SAW},o}$ is an infinite language. By Lemma 2.4 the k -multiple context-free language $L_{\text{SAW},o}$ 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 $x_1 y_1^n x_2 \dots y_{2k}^n x_{2k+1} \in L_{\text{SAW},o}$ for every $n \in \mathbb{N}_0$. Let $m = \min\{i \in [2k] : y_i \neq \epsilon\}$. 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 $L_{\text{SAW},o}$ and thus itself contained in $L_{\text{SAW},o}$. 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 [Hal73, Proposition 12] yields that τ is non-elliptic. \square

A locally finite, connected graph S is called a *strip* if it is quasi-transitive and has precisely two ends. Note that both ends of a strip have the same finite size k which we call the size of S . It is known that any strip S of size k has an automorphism τ such that the cyclic group $\langle \tau \rangle$ generated by τ acts quasi-transitively on S . By [Hal73, Theorem 9], there is $n \in \mathbb{N}$ such that S contains k disjoint τ^n -invariant double rays. Let us call S a τ -strip for $\tau \in \text{AUT}(S)$, if the last statement holds with $n = 1$. We use the same notation if S is a sub-graph of a graph G invariant under $\tau \in \text{AUT}(G)$.

The following lemma is a combination of [LW20, Lemmas 3.3 and 3.4] 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 6.7. *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 6.8. *Let G be a connected, infinite, deterministically edge-labelled graph on which $\Gamma = \text{AUT}(G, \ell)$ 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 $\tau \in \Gamma$, then $L_{\text{SAW},o}$ 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 $\ell(\mathcal{P}_{\text{SAW}})$ of the subset \mathcal{P}_{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 $L_{\text{SAW},o}$ is not k -multiple context-free.

The set \mathcal{P} essentially consists of spiral-shaped walks on the strip S , see Figure 6.3. For a concise definition, first recall that the strip S contains $2k + 1$ τ -invariant rays R_1, \dots, R_{2k+1} by the definition of a τ -strip, and that the subgroup $\langle \tau \rangle$ of Γ generated by τ acts quasi-transitively

on S . Let K be a set of orbit representatives of the action of $\langle \tau \rangle$ on S such that the induced sub-graphs $S[K]$ and $R_i[K]$ (for every $i \in [2k + 1]$) are connected.

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 for which 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 $\tau(W_{i+1,i})$. 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} . Note that X_i is self-avoiding because $W_{i-1,i+1}$ and $W_{i+1,i}$ are fully contained in $S[K]$. Furthermore observe that X_i is contained in $T_K(i-1)$ and 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 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

$$X_1 Y_1^{n_1} X_2 Y_2^{n_2} \dots X_{2k+1} Y_{2k+1}^{n_{2k+1}}, \tag{6.3}$$

where $n_1, \dots, n_{2k+1} \in \mathbb{N}$. See Figure 6.3 for an illustration of an element of \mathcal{P} .

The language $L(\mathcal{P})$ has the form

$$L(\mathcal{P}) = \{x_1 y_1^{n_1} x_2 y_2^{n_2} \dots x_{2k+1} y_{2k+1}^{n_{2k+1}} : n_1, \dots, n_{2k+1} \in \mathbb{N}\} \tag{6.4}$$

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 (6.3) 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 $Y_i^{n_i}$.

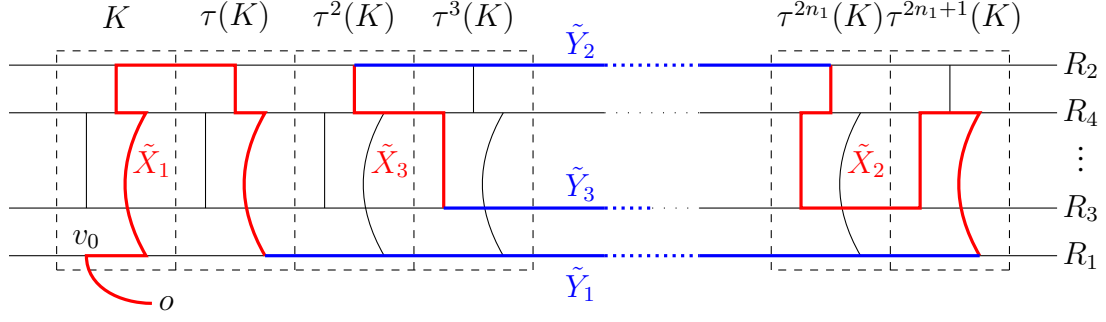


Figure 6.3: Spiral shaped walks in \mathcal{P} . The dashed rectangles contain the set K and its respective translates under τ^i .

In the example shown in Figure 6.3 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_j . 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

$$0 \geq l_j - l_i = \sum_{l=i}^{j-1} (-1)^{l-1} n_l = (n_i - n_{i+1}) + \cdots + (n_{j-2} - n_{j-1}),$$

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 $L_{\text{SAW},o}$ 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 2.5, we derive a contradiction. First, note that the language

$$L(\mathcal{P}_{\text{SAW}}) = \{x_1 y_1^{n_1} x_2 y_2^{n_2} \cdots x_{2k+1} y_{2k+1}^{n_{2k+1}} : n_1 > n_2 > \cdots > n_{2k+1} > 0\}$$

is the intersection of the regular language $L(\mathcal{P})$ and the k -multiple context-free language $L_{\text{SAW},o}$ and thus must be k -multiple context-free. We define a language homomorphism

$$\phi : \{a_1, b_1, \dots, a_{2k+1}, b_{2k+1}\}^* \rightarrow \Sigma^*$$

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

$$L_1 = \{a_1 b_1^{n_1} \dots a_{2k+1} b_{2k+1}^{n_{2k+1}} : n_1 \geq n_2 \geq \dots \geq n_{2k+1} \geq 0\}$$

is k -multiple context-free because

$$L_1 = \phi^{-1}(L(\mathcal{P}_{\text{SAW}})) \cap \{a_1 b_1^{n_1} \dots a_{2k+1} b_{2k+1}^{n_{2k+1}} : n_1, \dots, n_{2k+1} \in \mathbb{N}_0\}.$$

Note that this statement strongly relies on the fact that the edge-labelling ℓ is deterministic: the image $\phi(w)$ of any word $w \in \{a_1 b_1^{n_1} \dots a_{2k+1} b_{2k+1}^{n_{2k+1}} : n_1, \dots, n_{2k+1} \in \mathbb{N}_0\}$ is the label of a unique walk in \mathcal{P} and thus has a unique representation of the form (6.4), which lies in $L(\mathcal{P}_{\text{SAW}})$ if and only if $n_1 \geq n_2 \geq \dots \geq n_{2k+1} \geq 0$.

Finally, the language $L_2 = \{c_1^{n_1} \dots c_{2k+1}^{n_{2k+1}} : n_1 \geq n_2 \geq \dots \geq n_{2k+1} \geq 0\}$ 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 2.5. \square

The property of having a k -multiple context-free language of SAWs is closed under taking certain sub-graphs. The following lemma extends [LW20, Lemma 4.2] to MCFLs; the proof works exactly the same and is thus omitted.

Lemma 6.9. *Let H be a sub-graph of G which is invariant under a subgroup Γ of $\text{AUT}(G, \ell)$ acting quasi-transitively on H . If $L_{\text{SAW},o}(G)$ is k -multiple context-free, then there is a vertex $o' \in V(H)$ such that $L_{\text{SAW},o'}(H)$ 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 6.5. By Lemma 6.8 there is no $\tau \in \Gamma = \text{AUT}(G, \ell)$ such that the graph G contains a τ -strip of size $2k + 1$. Lemma 6.7 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 3.1 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 3 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 3.4 the set-wise stabiliser $\Gamma_{\mathcal{V}(t)} \leq \Gamma$ of $\mathcal{V}(t)$ acts quasi-transitively on this part.

Let H be the sub-graph of G obtained from the induced sub-graph $G[\mathcal{V}(t)]$ 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 6.9 there exists a vertex o' of H such that the language $L_{\text{SAW},o'}(H)$ is k -multiple context-free and by Lemma 6.6 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 6.7, 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 6.8. We conclude that all ends of G are thin. \square

Acknowledgements

The authors express their gratitude to Wolfgang Woess for introducing them to each other and to the research problems discussed in this paper.

References

- [AJ90] Sven Erick Alm and Svante Janson. Random self-avoiding walks on one-dimensional lattices. *Comm. Statist. Stochastic Models*, 6(2):169–212, 1990. doi:10.1080/15326349908807144.
- [Ani71] Anatoly V. Anisimov. The group languages. *Kibernetika (Kiev)*, 4:18–24, 1971.
- [Ant11] Yago Antolín. On Cayley graphs of virtually free groups. *Groups Complex. Cryptol.*, 3(2):301–327, 2011. doi:10.1515/gcc.2011.012.
- [BD15] Cyril Banderier and Michael Drmota. Formulae and asymptotics for coefficients of algebraic functions. *Combin. Probab. Comput.*, 24(1):1–53, 2015. doi:10.1017/S0963548314000728.
- [BDCGS12] Roland Bauerschmidt, Hugo Duminil-Copin, Jesse Goodman, and Gordon Slade. Lectures on self-avoiding walks. In *Probability and statistical physics in two and more dimensions*, volume 15 of *Clay Math. Proc.*, pages 395–467. Amer. Math. Soc., Providence, RI, 2012.
- [CS63] Noam Chomsky and Marcel-Paul Schützenberger. The algebraic theory of context-free languages. In *Computer programming and formal systems*, pages 118–161. North-Holland, Amsterdam, 1963. doi:10.1016/S0049-237X(08)72023-8.
- [CSW02] Tullio Ceccherini-Silberstein and Wolfgang Woess. Growth and ergodicity of context-free languages. *Trans. Amer. Math. Soc.*, 354(11):4597–4625, 2002. doi:10.1090/S0002-9947-02-03048-9.
- [DCS12] Hugo Duminil-Copin and Stanislav Smirnov. The connective constant of the honeycomb lattice equals $\sqrt{2 + \sqrt{2}}$. *Ann. of Math. (2)*, 175(3):1653–1665, 2012. doi:10.4007/annals.2012.175.3.14.
- [DK15] Martin J. Dunwoody and Bernhard Krön. Vertex cuts. *J. Graph Theory*, 80(2):136–171, 2015. doi:10.1002/jgt.21844.

- [Dun82] Martin J. Dunwoody. Cutting up graphs. *Combinatorica*, 2(1):15–23, 1982. doi:10.1007/BF02579278.
- [Fla87] Philippe Flajolet. Analytic models and ambiguity of context-free languages. *Theoret. Comput. Sci.*, 49(2-3):283–309, 1987. Twelfth international colloquium on automata, languages and programming (Nafplion, 1985). doi:10.1016/0304-3975(87)90011-9.
- [Flo53] Paul J. Flory. *Principles of Polymer Chemistry*. Cornell University Press, 1953.
- [FS09] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009. doi:10.1017/CB09780511801655.
- [GL19] Geoffrey R. Grimmett and Zhongyang Li. Self-avoiding walks and connective constants. In *Sojourns in probability theory and statistical physics. III. Interacting particle systems and random walks, a festschrift for Charles M. Newman*, pages 215–241. Singapore: Springer; Shanghai: NYU Shanghai, 2019. doi:10.1007/978-981-15-0302-3_8.
- [GM17] Lorenz A. Gilch and Sebastian Müller. Counting self-avoiding walks on free products of graphs. *Discrete Math.*, 340(3):325–332, 2017. doi:10.1016/j.disc.2016.08.018.
- [Hal65] Rudolf Halin. Über die Maximalzahl fremder unendlicher Wege in Graphen. *Math. Nachr.*, 30:63–85, 1965. doi:10.1002/mana.19650300106.
- [Hal73] Rudolf Halin. Automorphisms and endomorphisms of infinite locally finite graphs. *Abh. Math. Sem. Univ. Hamburg*, 39:251–283, 1973. doi:10.1007/BF02992834.
- [Hal76] Rudolf Halin. S -functions for graphs. *J. Geom.*, 8(1-2):171–186, 1976. doi:10.1007/BF01917434.
- [Ham57] John M. Hammersley. Percolation processes. II. The connective constant. *Proc. Cambridge Philos. Soc.*, 53:642–645, 1957. doi:10.1017/S0305004100032692.
- [HLMR22] Matthias Hamann, Florian Lehner, Babak Miraftab, and Tim Rühmann. A Stallings type theorem for quasi-transitive graphs. *J. Combin. Theory Ser. B*, 157:40–69, 2022. doi:10.1016/j.jctb.2022.05.008.
- [Ho18] Meng-Che Ho. The word problem of \mathbb{Z}^n is a multiple context-free language. *Groups Complex. Cryptol.*, 10(1):9–15, 2018. doi:10.1515/gcc-2018-0003.
- [KS86] Werner Kuich and Arto Salomaa. *Semirings, automata, languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1986. doi:10.1007/978-3-642-69959-7.
- [LL21] Florian Lehner and Christian Lindorfer. Comparing consecutive letter counts in multiple context-free languages. *Theoret. Comput. Sci.*, 868:1–5, 2021. doi:10.1016/j.tcs.2021.03.034.
- [LW20] Christian Lindorfer and Wolfgang Woess. The language of self-avoiding walks. *Combinatorica*, 40(5):691–720, 2020. doi:10.1007/s00493-020-4184-z.

- [MS83] David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983. doi:10.1016/0022-0000(83)90003-X.
- [MS85] David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoret. Comput. Sci.*, 37(1):51–75, 1985. doi:10.1016/0304-3975(85)90087-8.
- [MS96] Neal Madras and Gordon Slade. *The self-avoiding walk*. Modern Birkhäuser Classics. Birkhäuser/Springer, New York, 1996. Reprint 2013. doi:10.1007/978-1-4612-4132-4.
- [RS84] Neil Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *J. Combin. Theory Ser. B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.
- [Sal15] Sylvain Salvati. MIX is a 2-MCFL and the word problem in \mathbb{Z}^2 is captured by the IO and the OI hierarchies. *J. Comput. System Sci.*, 81(7):1252–1277, 2015. doi:10.1016/j.jcss.2015.03.004.
- [SMFK91] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoret. Comput. Sci.*, 88(2):191–229, 1991. doi:10.1016/0304-3975(91)90374-B.
- [SS78] Arto Salomaa and Matti Soittola. *Automata-theoretic aspects of formal power series*. Texts and Monographs in Computer Science. Springer-Verlag, New York-Heidelberg, 1978. doi:10.1007/978-1-4612-6264-0.
- [Tho92] Carsten Thomassen. The Hadwiger number of infinite vertex-transitive graphs. *Combinatorica*, 12(4):481–491, 1992. doi:10.1007/BF01305240.
- [TW93] Carsten Thomassen and Wolfgang Woess. Vertex-transitive graphs and accessibility. *J. Combin. Theory Ser. B*, 58(2):248–268, 1993. doi:10.1006/jctb.1993.1042.
- [Vie85] Gérard Viennot. Enumerative combinatorics and algebraic languages. In *Fundamentals of computation theory (Cottbus, 1985)*, volume 199 of *Lecture Notes in Comput. Sci.*, pages 450–464. Springer, Berlin, 1985. doi:10.1007/BFb0028829.
- [Woe00] Wolfgang Woess. *Random walks on infinite graphs and groups*, volume 138 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2000. doi:10.1017/CB09780511470967.
- [Zei96] Doron Zeilberger. Self-avoiding walks, the language of science, and Fibonacci numbers. *J. Statist. Plann. Inference*, 54(1):135–138, 1996. doi:10.1016/0378-3758(95)00162-X.