Ground Tree Rewriting Graphs of Bounded Tree Width

Christof Löding

RWTH Aachen

Abstract. We analyze structural properties of ground tree rewriting graphs, generated by rewriting systems that perform replacements at the front of finite, ranked trees. The main result is that the class of ground tree rewriting graphs of bounded tree width exactly corresponds to the class of pushdown graphs. Furthermore we show that ground tree rewriting graphs of bounded clique width also have bounded tree width.

1 Introduction

Graphs are an important tool to describe the behaviour of programs or processes. While finite graphs can be used to deal with finite-state processes, infinite-state processes require the use of infinite (transition-)graphs. To allow algorithmic approaches to tasks like verification of infinite-state systems, these systems have to be given effectively, i.e., by some finite object. Given a formalism for a finite representation of infinite graphs, there are two natural questions arising.

1. Which decision problems can be solved for the graphs generated by this formalism?
2. How expressive is the given formalism and how is it related to other formalisms for the specification of graphs?

In [MS85] Muller and Schupp analyze transition graphs of pushdown automata. They show that the monadic second-order (MSO) theory of these graphs is decidable. Furthermore they give a structural characterization of pushdown graphs. This characterization is in terms of the number of non-isomorphic connected components one obtains by deleting vertices within fixed diameters around the root vertex.

Pushdown graphs are generated by prefix rewriting systems on words. Starting from this idea, there are several ways of obtaining other classes of graphs. One possibility is to use more general forms of rewriting on words. The decidability result for the MSO theory on pushdown graphs was extended in [Cau96] to prefix recognizable graphs, which are generated by rewriting rules using regular languages instead of single words. Even more general classes of graphs can be obtained when using finite word transducers to define the edge relation of a graph. This leads to automatic graphs in case of synchronous transducers (see e.g. [BG00]) and to rational graphs ([Mor99]) in case of asynchronous transducers. Since both of these classes contain all transition graphs of Turing machines, reachability problems are undecidable on these graphs, but the first-order theory of automatic graphs is still decidable (see [BG00]) which is not the case for rational graphs ([Mor99]).

Another possibility for obtaining new classes of graphs is to pass from rewritings of words to rewritings on other structures. In [May00] classes of process rewriting graphs are analyzed. In process rewriting parallel composition of words is allowed in addition to the usual sequential composition. Depending on what operators (parallel and sequential composition) are allowed in the rewritings, one gets a hierarchy of classes of graphs. For these different classes decidable logics are exhibited in [May00] and their relation with respect to bisimulation is studied.

In the present paper we use another generalization of word rewriting, namely tree rewriting. In tree (or term) rewriting the basic objects in the rewriting systems are trees and the simplest form of rewriting is ground rewriting on these models. For ground tree rewrite systems confluence [DHLT90], the first-order theory [DT90], and several reachability problems [CG90] have been shown to be decidable. But so far there has been no analysis of the structure of graphs generated by ground tree rewriting systems. In this paper we try to fill this gap, i.e., we deal with Question 2 from above for the formalism of ground tree rewriting systems. The interest in the structure of these graphs comes from the fact that ground tree rewriting systems allow the specification of grid structures that are automatic but not prefix recognizable. On the other hand there are prefix recognizable graphs that can not be generated by ground tree rewriting systems. For our structural analysis we consider the properties tree width (see [Die00]) and clique width ([CO00]) of graphs, which amount to "complexity measures for graphs"; the former in terms of tree decompositions and the latter in terms of infinite graph expressions.

For the class of prefix recognizable graphs and subclasses these notions of width have been analyzed (see [Blu01] for an overview): Prefix recognizable graphs are of bounded clique width [Blu01], the class of HR-equational graphs (hyperedge replacement graphs), a class of graphs in between pushdown and prefix recognizable graphs, corresponds to the class of prefix recognizable graphs of bounded tree width ([Bar98]), and the class of HR-equational graphs of bounded degree exactly matches the class of pushdown graphs ([Cau92]).

Here we provide a similar analysis for ground tree rewriting graphs with the following result. The classes of pushdown graphs, ground tree rewriting graphs of bounded tree width, and ground tree rewriting graphs of bounded clique width coincide. First of all this result shows that the MSO theory of ground tree rewriting graphs of bounded tree width (or clique width) is decidable since it is decidable for pushdown graphs. Furthermore the structure of pushdown graphs is well understood ([MS85]). Thus, our result provides a method for showing that certain graphs of bounded tree or clique width can not be generated by ground tree rewriting systems.

The paper is organized as follows. In Sections 2 and 3 we introduce the basic terminology and definitions for graphs, trees, and rewriting systems. The rewriting systems under consideration are ground tree rewriting systems, pushdown systems, and infix pushdown systems (a novel extension of pushdown systems that we use as intermediate model for the proof of the main result). In Section 4 we show how to transform a ground tree rewriting system generating a graph of bounded tree width into an infix pushdown system. Finally, in Section 5, we prove that infix pushdown graphs of bounded tree width are pushdown graphs.

2 Graphs and Trees

In this section we introduce some general terminology on graphs and trees.

An edge labeled graph G is a tuple $G = (V, E, \Sigma)$, where V is the set of vertices, Σ is the set of edge labels, and $E \subseteq V \times \Sigma \times V$ is the set of edges. We only consider countable graphs, i.e., the set V is countable. If we are not interested in the edge labels or if we consider graphs without edge labels, we omit Σ and assume that $E \subseteq V \times V$.

A graph $G' = (V', E')$ is a subgraph of G iff $V' \subseteq V$ and $E' \subseteq E$. We call G' the subgraph induced by V' iff $E' = E \cap (V' \times \Sigma \times V')$.

A path π in G is a nonempty sequence of vertices $\pi = v_0 \cdots v_n \in V^*$ such that $(v_i, v_{i+1}) \in E$ for all $i \in \{0, \ldots, n-1\}$. An undirected path in G is a nonempty sequence of vertices $\pi = v_0 \cdots v_n \in V^*$ such that $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$ for all $i \in \{0, \ldots, n-1\}$. The length of a (undirected) path $\pi = v_0 \cdots v_n$ is n. For two vertices $u, v \in V$ we say that there is a path (undirected path) from u to v if there is a path (undirected path) $\pi = v_0 \cdots v_n$ in G with $v_0 = u$, and $v_n = v$.

Given a vertex $v \in V$, the connected component of v in G is the subgraph of G induced by the set $\{u \in V \mid \text{ there is an undirected path from } u \text{ to } v\}$.

Special graphs that we use in this article are the complete bipartite graphs $K_{m,m}$ with $2m$ vertices such that the edge relation is the set $V_1 \times V_2$ for a partition of vertex set into two sets V_1 and V_2 of size m.

The $(m \times m)$-grid has the vertex set $\{1, \ldots, m\} \times \{1, \ldots, m\}$ and from each vertex (i, j) there is an edge to $(i, j+1)$ if $j < m$ and to $(i+1, j)$ if $i < m$. We will use large grids (or grid-like graphs) as a tool for showing large tree width of graphs.

A tree-decomposition of $G = (V, E)$ is a pair $(T, (W_t)_{t \in V_T})$ where $T = (V_T, E_T)$ is a tree (in the graph-theoretic sense), $W_t \subseteq V$ for all $t \in V_T$, and

- $\forall (u, v) \in E \ \exists t \in V_T \ : \ u, v \in W_t$,
- $\forall v \in V$ the subgraph of T induced by $\{t \in V_T \mid v \in W_t\}$ is connected.

The width of a tree-decomposition is $\max\{|W_t| \mid t \in V_T\} - 1$ and ∞ if the maximum does not exist[1]. The tree width $tw(G)$ of G is the minimal width of a tree-decomposition of G.

For the following facts about tree width of graphs see e.g. [Die00].

Proposition 1. *The $(k \times k)$-grid has tree width k.*

[1] Since trees should have tree width 1, the "-1" is used in the definition

Proposition 2. *Let H be a subgraph of a graph G. Then $tw(G) \geq tw(H)$.*

In this paper we mainly need to show lower bounds on the tree width of graphs. For this purpose we introduce "brambles". A bramble of G is a finite family of sets of vertices $\mathcal{B} = (B_i)_{i \in I}$ such that

(1) each B_i induces a connected subgraph in G, and
(2) for each $i, j \in I$: $B_i \cap B_j \neq \emptyset$ or there are $u \in B_i$, $v \in B_j$ with $(u, v) \in E$.

A set $S \subseteq V$ covers $\mathcal{B}$ iff $S \cap B_i \neq \emptyset$ for each $i \in I$. The width of a bramble is $\mathrm{width}(\mathcal{B}) = \min\{|S| \mid S \text{ covers } \mathcal{B}\}$.

The following result can be found in [Die00] for finite graphs. An extension to infinite graphs is straightforward (whereas the other direction is nontrivial).

Proposition 3. *If a graph G contains a bramble of width k, then $tw(G) \geq k-1$.*

The notion of clique width of graphs was introduced in [CO00]. Although our graphs are directed, we consider undirected clique width. This means in this paper the clique width of a directed graph G is the clique width of the undirected graph obtained from G by replacing every directed edge with an undirected one. We will only give an informal description of clique width. For details the reader is referred to [CO00].

To define clique width of graphs one considers (infinite) graph expressions over colored graphs. These expressions are built up from the constant graph with one vertex, colored c, and the operators $\oplus, \rho, \eta$, where

- $G_1 \oplus G_2$ is the disjoint union of the graphs G_1 and G_2,
- $\rho_{c_1 \to c_2}(G)$ changes the color of all c_1 colored vertices in G into c_2, and
- $\eta_{c_1, c_2}(G)$ introduces edges between all pairs of vertices u, v, where u is colored c_1 and v is colored c_2.

The clique width of a graph G is the minimal number of colors that is needed in such an expression defining G. The following propositions relate the notions of clique width and tree width.

Proposition 4. ([CO00]) *Let G be a graph. If G has bounded tree width, then G has bounded clique width.*

Proposition 5. [GW00] *Let G be a graph. If G is of bounded clique width and there is an $n \in \mathbb{N}$ such that $K_{n,n}$ is not a subgraph of G, then G is of bounded tree width.*

We will generate graphs using ground tree rewriting systems. These are introduced in the next section. Here we define finite ranked trees, the objects the rewritings will be applied to.

A ranked alphabet A is a family of sets $(A_i)_{i \in [k]}$, where $[k] = \{0, \ldots, k\}$. For simplicity we identify A with the set $\bigcup_{i=0}^{k} A_i$. A ranked tree t over A is a mapping $t : D_t \to A$ with $D_t \subseteq \mathbb{N}^*$ such that D_t is prefix closed, for each $ui \in D_t$ we have $uj \in D_t$ for all $j \leq i$, and if $u0, \ldots, u(i-1) \in D_t$, $ui \notin D_t$, then $t(u) \in A_i$. D_t is called the domain of t. The set of all trees over A is called T_A. By $\sqsubseteq$ we denote the prefix ordering on $\mathbb{N}^*$. The height of a tree $t \in T_A$ is $\text{height}(t) = max\{|u| \mid u \in D_t\}$, i.e., the length of a longest path through t.

The rewriting rules used later will be replacements of subtrees. Here we formally define what it means to replace a subtree. For $u \in \mathbb{N}^*$ we define $uD_t = \{uv \in \mathbb{N}^* \mid v \in D_t\}$ and $u^{-1}D_t = \{v \in \mathbb{N}^* \mid uv \in D_t\}$. For $u \in D_t$ the subtree t^u of t at u is the tree with domain $D_{t^u} = u^{-1}D_t$ and $t^u(v) = t(uv)$. For trees $t, t' \in T_A$ and $u \in D_t$ we define $t(u \leftarrow t')$ to be the tree s with domain $D_s = uD_{t'} \cup (D_t \setminus u(u^{-1}D_t))$ and $s(v) = t'(u^{-1}v)$ for $v \in uD_{t'}$ and $s(v) = t(v)$ for $v \in D_t \setminus u(u^{-1}D_t)$. This means we replace the subtree t^u in t by t'.

3 Rewriting Systems on Words and Trees

In this section we introduce different rewriting systems we are using to generate infinite graphs. These are ground tree rewriting systems and pushdown automata. In the next sections we are going to translate ground tree rewriting systems that generate graphs of bounded tree width into pushdown automata such that the generated graphs are isomorphic. For this purpose we also need an intermediate model, which we call infix pushdown automata. All the graphs under consideration are rooted graphs of finite out-degree.

A ground tree rewriting system (GTRS) is a tuple $S = (A, \Sigma, P, t_I)$, where

- $A = (A_i)_{i \in [k]}$ is a ranked alphabet,
- Σ is an alphabet,
- P is a finite set of rules $s \to^\sigma s'$ with $s, s' \in T_A$, $\sigma \in \Sigma$, and
- $t_I \in T_A$ is the initial tree.

For two trees $t, t' \in T_A$ we write $t \to_S^\sigma t'$ iff there exists a rule $s \to^\sigma s'$ in P and $u \in D_t$ such that $t^u = s$ and $t' = t(u \leftarrow s')$. We write $t \to_S t'$ iff there is a $\sigma \in \Sigma$ with $t \to_S^\sigma t'$. By $\to_S^*$ we denote the transitive and reflexive closure of $\to_S$. The tree language that is generated by S is $T(S) = \{t \in T_A \mid t_I \to_S^* t\}$.

In [Bra69] it is shown that the set $T(S)$ is a regular set of trees. Here we do not consider the language generated by such a system but the graph G_S that is generated.

Let $S = (A, \Sigma, P, t_I)$ be a GTRS. The edge labeled graph $G_S = (V_S, E_S, \Sigma)$ generated by S is defined by $V_S = T(S)$, and $(t, \sigma, t') \in E_S$ iff $t \to_S^\sigma t'$.

Example 1. The GTRS $S = (A, \Sigma, P, t_I)$ given by

- $\Sigma = \{0, 1\}$, $A = (A_i)_{i \in [2]}$ with $A_0 = \{a, b\}$, $A_1 = \{c\}$, and $A_2 = \{d\}$,
- $P = \left\{ b \to^0 \begin{array}{c} c \\ | \\ b \end{array}, a \to^1 \begin{array}{c} c \\ | \\ a \end{array} \right\}$, and $t_I = \begin{array}{c} d \\ / \ \backslash \\ a \quad b \end{array}$

generates the infinite grid as shown in Figure 1. This demonstrates that GTRS can generate graphs which have an undecidable MSO theory and therefore are not pushdown graphs. □

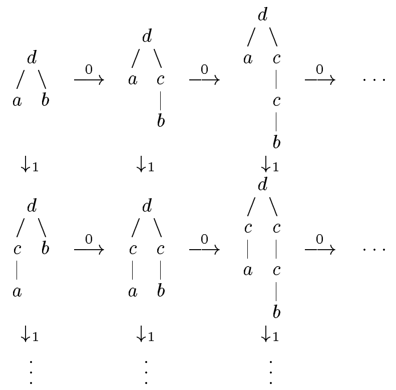


Fig. 1. The infinite grid generated by a GTRS.

To avoid complications we always assume that height$(t_I) \leq 1$. This can always be obtained by adding new symbols and rules to S.

A pushdown automaton (PDA) M is a tuple $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$, where Q is a finite set of states, Σ is the input alphabet, Γ is the stack alphabet (disjoint from Q), $\Delta \subseteq Q \times (\Gamma \cup \{\varepsilon\}) \times \Sigma \times Q \times \Gamma^*$ is the transition relation (ε denotes the empty word), $q_0 \in Q$ is the initial state, and $\gamma_0 \in \Gamma$ is the initial stack symbol. A configuration of M is a word qw with a state $q \in Q$ and a stack content $w \in \Gamma^*$. Given configurations q_1w_1, q_2w_2 and an input $\sigma \in \Sigma$ we write $q_1w_1 \vdash_M^\sigma q_2w_2$ if there is a transition $(q_1, \gamma, \sigma, q_2, v) \in \Delta$ such that $w_1 = \gamma w$ and $w_2 = vw$. We write $q_1w_1 \vdash_M q_2w_2$ if there is an input $\sigma \in \Sigma$ with $q_1w_1 \vdash_M^\sigma q_2w_2$ and by $\vdash_M^*$ we denote the transitive and reflexive closure of $\vdash_M$. The set of configurations $C(M)$ generated by M is the set of configurations that are reachable from the initial configuration of M: $C(M) = \{qw \mid q_0\gamma_0 \vdash_M^* qw\}$.

The language $C(M)$ is regular (a result due to Büchi) but as for GTRS we are interested in the graph generated by the PDA. The directed edge labeled graph $G_M = (V_M, E_M, \Sigma)$ generated by M has the set of vertices $V_M = C(M)$ and for two words $q_1w_1, q_2w_2 \in V_M$ there is an edge labeled with σ from q_1w_1 to q_2w_2 if $q_1w_1 \vdash_M^\sigma q_2w_2$. Given a graph G we say that G is a pushdown graph iff G is isomorphic to the graph G_M for some PDA M. The following proposition on pushdown graphs is not difficult to prove.

Proposition 6. *Pushdown graphs are of bounded tree width.*

In usual pushdown automata we only have rewritings on top of the stack. We also need an extended type of pushdown automata, where we allow a restricted type of infix rewriting in the stack.

An infix PDA is a system $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$, where $Q, \Sigma, \Gamma, q_0, \gamma_0$ are as in usual pushdown automata and in Δ there are transitions of the usual form but also transitions from $\Gamma \times \Sigma \times \Gamma$. These are the infix rules. The other rules are called prefix rules. Now we write $q_1 w_1 \vdash_{M_{\text{pre}}} q_2 w_2$ if $q_1 w_1 \vdash_M q_2 w_2$ in the usual sense, and $q w_1 \vdash_{M_{\text{in}}} q w_2$ if there is an infix rule $(\gamma_1, \sigma, \gamma_2) \in \Delta$ such that $w_1 = w\gamma_1 w'$ and $w_2 = w\gamma_2 w'$ with $w, w' \in \Gamma^*$. This means we are allowed to change single letters inside the stack.

We write $q_1 w_1 \vdash_M q_2 w_2$ if $q_1 w_1 \vdash_{M_{\text{pre}}} q_2 w_2$ or $q_1 w_1 \vdash_{M_{\text{in}}} q_2 w_2$. We adopt the notations $\vdash_M^*$, $\vdash_{M_{\text{pre}}}^*$, $\vdash_{M_{\text{in}}}^*$, $C(M)$, and G_M for infix pushdown automata in the obvious way.

In Section 4 we will transform certain ground tree rewriting systems into infix pushdown automata. The transitions of the obtained automata will have a certain form.

Definition 1. Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA. Then M is in normal form iff

1. all elements in Δ are of the form $(q, \gamma, \sigma, q', \varepsilon)$, $(q, \varepsilon, \sigma, q', \gamma)$, $(q, \varepsilon, \sigma, q', \varepsilon)$, $(q, \gamma_0, \sigma, q', \gamma_0)$, or $(\gamma, \sigma, \gamma')$ with $q, q' \in Q$, $\gamma, \gamma' \in \Gamma \setminus \{\gamma_0\}$, and $\sigma \in \Sigma$, and
2. whenever $qw \vdash_{M_{\text{pre}}} q'\gamma w \vdash_{M_{\text{in}}} q'\gamma' w$, then there is a $q'' \in Q$ such that $qw \vdash_{M_{\text{pre}}} q'' w \vdash_{M_{\text{pre}}} q'\gamma' w$.

The essential property of infix pushdown automata in normal form is that every reachable configuration can also be reached only by applying prefix rules. For later use we formulate the more general claim that, given a reachable configuration quw, there is a reachable configuration $q'w$ such that one can reach quw from $q'w$ only by applying prefix rules and without "touching" the w.

Lemma 1. *Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form, $q \in Q$, and $u, w \in \Gamma^*$. If $quw \in C(M)$, then there exists $q' \in Q$ such that $q' \vdash_{M_{\text{pre}}}^* qu$, and $q'w \in C(M)$.*

Proof. Since $quw \in C(M)$, there is a directed path from $q_0\gamma_0$ to quw. We chose this path such that every infix rule is applied when the corresponding symbol is on top of the stack (this is possible because by point 1 of Definition 1 the prefix rules increasing the stack length do not depend on the stack content). Then, by point 2 of Definition 1, we can replace all the infix rules on this path and get $q_0\gamma_0 \vdash_{M_{\text{pre}}}^* quw$. On this path π from $q_0\gamma_0$ to quw, there must be a vertex $q'w$ such that every successor of $q'w$ on π contains the suffix w. But this means $q' \vdash_{M_{\text{pre}}}^* qu$. $\square$

4 GTRS Graphs of Bounded Tree and Clique Width

In this section we show that GTRS graphs of bounded tree width are infix pushdown graphs. We do this by analyzing the trees that are generated by a rewriting system S producing a graph G_S of bounded tree width. These trees have a special form that allows us to code them as words.

Let $S = (A, \Sigma, P, t_I)$ be a GTRS. If we are given $u \in \mathbb{N}^*$ and $t' \in T_A$, then we call $\alpha = (u, t')$ a substitution. A substitution $\alpha = (u, t')$ is S-applicable to

$t \in T_A$ iff $u \in D_t$ and there is a rule $t^u \to^\sigma t'$ in P. The result of the substitution is $\alpha(t) = t(u \leftarrow t')$. A sequence of substitutions $(\alpha_1 \cdots \alpha_k)$ is S-applicable to t if α_1 is S-applicable to t and $(\alpha_2 \cdots \alpha_k)$ is S-applicable to $\alpha_1(t)$. We define $(\alpha_1 \cdots \alpha_k)(t) = (\alpha_2 \cdots \alpha_k)(\alpha_1(t))$.

Two substitutions $\alpha = (u_1, t_1)$ and $\beta = (u_2, t_2)$ are called independent iff u_1 and u_2 are not comparable with respect to $\sqsubseteq$. Two sequences $(\alpha_1 \cdots \alpha_k)$ and $(\beta_1 \cdots \beta_l)$ of substitutions are called independent iff α_i and β_j are independent for each $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, l\}$. Since independent sequences of substitutions can be interleaved arbitrarily, it is not difficult to see that they generate a grid with size corresponding to the length of the sequences as subgraph[2].

Lemma 2. *Let $S = (A, \Sigma, P, t_I)$ be a GTRS. If there is a $t \in T(S)$ and two independent sequences $(\alpha_1 \cdots \alpha_k)$ and $(\beta_1 \cdots \beta_k)$ of substitutions that are S-applicable to t such that $(\alpha_1 \cdots \alpha_j)(t) \neq (\alpha_1 \cdots \alpha_i)(t)$ and $(\beta_1 \cdots \beta_j)(t) \neq (\beta_1 \cdots \beta_i)(t)$ for all $i, j \in \{1, \ldots, k\}$ with $i \neq j$, then G_S has a $(k+1) \times (k+1)$-grid as subgraph.*

We consider a consequence of this lemma for the trees generated by a GTRS S where G_S has bounded tree width. These trees consist of a single long path with subtrees of bounded height branching from this path.

Lemma 3. *Let $S = (A, \Sigma, P, t_I)$ be a GTRS such that G_S has bounded tree width. Then there is a k such that for each $t \in T(S)$ and for each vertex x of t there is at most one successor y of x with* $\mathrm{height}(t^y) \geq k$.

Proof. Let $c = \max\{\mathrm{height}(t') - \mathrm{height}(t) \mid t \to^\sigma t' \in P\}$ (the maximal increase of height by an application of one rule), $m = tw(G_S)$, and define $k = c \cdot m + 1$. (Here we assume that $c \geq 1$. Otherwise G_S is finite and the claim obviously holds.)

Suppose there is a $t \in T(S)$ with a vertex x such that there are two successors y, z of x and the subtrees t^y and t^z of t have height $\geq k$. Since $t \in T(S)$, there is a sequence $(\alpha_1 \cdots \alpha_l)$ of substitutions with $(\alpha_1 \cdots \alpha_l)(t_I) = t$. Let $\alpha_i = (u_i, t'_i)$ and let $j \in \{0, \ldots, l\}$ be the smallest number with $u_i \not\sqsubseteq x$ for all $i > j$. By the choice of c, the subtrees t^y and t^z have height $\leq c$ in the tree $(\alpha_1 \cdots \alpha_j)(t_I)$ and we need at least m substitutions, respectively, to increase their height to be $\geq k$ (by the choice of k). Therefore we can extract from $(\alpha_{j+1} \cdots \alpha_l)$ two independent sequences of substitutions of length at least m that are S-applicable to $(\alpha_1 \cdots \alpha_j)(t_I)$ such that Lemma 2 can be applied, i.e., G_S has the $(m+1) \times (m+1)$-grid as subgraph and therefore by Propositions 1 and 2 tree width $\geq m + 1$, a contradiction. □

This property of the trees enables us to code them as words and to construct an infix PDA generating a graph isomorphic to the graph generated by the GTRS.

Theorem 1. *Let $S = (A, \Sigma, P, t_I)$ be a GTRS such that G_S has bounded tree width. Then we can construct an infix PDA $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ in normal form such that G_S is isomorphic to G_M.*

[2] Note that this subgraph in general is not an induced subgraph.

Proof. Let $\circ$ be a symbol of rank 0 not occurring in A. From Lemma 3 we can see that the trees in $T(S)$ may have one long path only. From this path subtrees of bounded height are branching. The idea is to cut such a tree along this path. The new symbol $\circ$ signals where the tree was cut as indicated in Figure 2. We can extend S to a GTRS S' for trees over $A \cup \{\circ\}$ by defining $S' = (A \cup \{\circ\}, \Sigma, P, t_I)$. We define a set of special trees over $A \cup \{\circ\}$ by $T_A^\circ = \{t \in T_{A \cup \{\circ\}} \mid t^{-1}(\circ) = \{x\}$ with $|x| = 1\}$. These are all the trees over $A \cup \{\circ\}$ containing exactly one $\circ$ at a successor of the root. Using these trees we can define concatenation of trees. For $t \in T_A^\circ$ and $t' \in T_A^\circ \cup T_A$ let $t \cdot t' = t(x \leftarrow t')$ where $x \in D_t$ with $t(x) = \circ$.

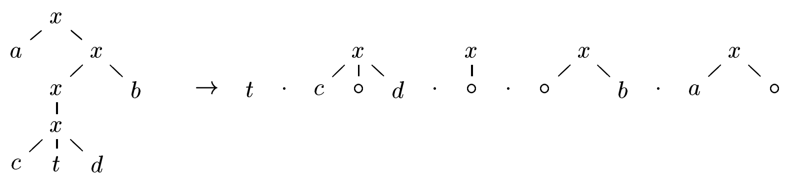


Fig. 2. Decomposition of a tree to pass from trees to words.

Let k be as in Lemma 3 and define the components of M by $Q = \{t \in T_A \mid \text{height}(t) \leq k\}$, $q_0 = t_I$ (by our convention $\text{height}(t_I) \leq 1 \leq k$), $\Gamma = \{t \in T_A^\circ \mid \text{height}(t) \leq k\} \dot\cup \{\gamma_0\}$ (all the special trees of height $\leq k$ over $A \cup \{\circ\}$ and a new initial stack symbol γ_0). By Γ' we denote $\Gamma \setminus \{\gamma_0\}$. For $\sigma \in \Sigma$, $r, r' \in \Gamma'$, and $t, t' \in Q$ we have the following rules in Δ.

1. $(r, \sigma, r') \in \Delta$ iff $r \rightarrow_{S'}^{\sigma} r'$.
2. $(t, \gamma_0, \sigma, t', \gamma_0) \in \Delta$ iff $t \rightarrow_S^\sigma t'$.
3. $(t, \varepsilon, \sigma, t', \varepsilon) \in \Delta$ iff $t \rightarrow_S^\sigma t'$ and $\text{height}(t') = k$.
4. $(t, r, \sigma, t', \varepsilon) \in \Delta$ iff $r \cdot t \rightarrow_S^\sigma t'$ and $\text{height}(t') = k$.
5. $(t, \varepsilon, \sigma, t', r) \in \Delta$ iff $t \rightarrow_S^\sigma r \cdot t'$ and $\text{height}(t') = k$.

Note that the S' in the first rule refers to the GTRS over the extended alphabet. Furthermore, the state of the infix PDA will always be a tree of height exactly k when the decomposed tree that is represented by stack and state has height $\geq k$ (Rules 3,4, and 5). This is to keep the decomposition of a tree unique.

Now one can verify that G_M is isomorphic to G_S via the mapping that maps each tree to its decomposition as indicated in Figure 2. $\square$

As mentioned before we also want to show that bounded clique width and bounded tree width imply each other for GTRS graphs. Using Proposition 5 this question can be reduced to the following lemma which can be shown by a detailed analysis of how complete bipartite graphs can be generated by a GTRS.

Lemma 4. *Let $S = (A, \Sigma, P, t_I)$ be a GTRS. Then $K_{m,m}$ is not a subgraph of G_S for all $m > 2 \cdot |P| + 1$.* $\square$

Now, using Propositions 4 and 5, we can conclude the following.

Corollary 1. *Let $S = (A, \Sigma, P, t_I)$ be a GTRS. Then G_S has bounded clique width iff G_S has bounded tree width.*

5 From Infix Pushdown Graphs to Pushdown Graphs

In this section we prove that infix pushdown graphs of bounded tree width are pushdown graphs. The idea is to find a structural characterization of infix PDA graphs of bounded tree width similar to the characterization of PDA graphs given by Muller and Schupp ([MS85]). For each vertex qw in the infix PDA graph we define the connected component of qw in the graph restricted to vertices of length greater than or equal to the length of qw. The goal is to show that, if the graph has bounded tree width, then there are only finitely many isomorphism classes of these connected components. The difference to the property of pushdown graphs given by Muller and Schupp is that we refer to the length of the vertices and not the distance to a designated root vertex.

Once we have this structural characterization, this finite number of isomorphism classes can be used to obtain a usual PDA for generating the graph.

Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form. For $w, v \in \Gamma^*$ we define

$$w \preceq_M v \text{ iff } \forall q \in Q \ : \ (qw \in C(M) \Rightarrow qv \in C(M)),$$
$$w \approx_M v \text{ iff } w \preceq_M v \text{ and } v \preceq_M w.$$

The following lemma is immediate from the definition of $\approx_M$.

Lemma 5. *The relation $\approx_M$ is an equivalence relation of finite index.*

For $n \in \mathbb{N}$ the graph $G(M)|_n$ is the subgraph of G_M induced by the vertices $\{qw \in C(M) \mid |qw| \geq n\}$. For $qw \in C(M)$ the graph $G_M(qw)$ is the connected component of qw in $G_M|_{|qw|}$. The front $\Omega(qw)$ of $G_M(qw)$ is the set of all vertices in $G_M(qw)$ with the same length as qw, i.e.,

$$\Omega(qw) = \{pv \in G_M(qw) \mid |pv| = |qw|\}.$$

Let $pv \in C(M)$, qw a vertex in $G_M(pv)$, and $\pi = (p_1 x_1 u_1, \ldots, p_n x_n u_n)$ be an undirected path in $G_M(pv)$ from pv to qw with $|u_i| = |v|$ for all $i \in \{1, \ldots, n\}$. The depth of π is $d(\pi) = max\{|x_i| \mid i \in \{1, \ldots, n\}\}$ and the number of suffix changes on π is $sc(\pi) = |\{i \in \{1, \ldots, n-1\} \mid u_i \neq u_{i+1}\}|$.

Let $\Pi(pv, qw)$ be the set of undirected paths in $G_M(pv)$ from pv to qw. We fix a path $\pi(pv, qw) \in \Pi(pv, qw)$ with

1. $\forall \pi \in \Pi(pv, qw) : sc(\pi(pv, qw)) \leq sc(\pi)$ and
2. $\forall \pi \in \Pi(pv, qw) : sc(\pi(pv, qw)) = sc(\pi) \Rightarrow d(\pi(pv, qw)) \leq d(\pi)$.

By induction on $sc(\pi(pv, qw))$ one can show that vertices that belong to the same front have a connection of small depth.

Lemma 6. *Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form. There exists $d_M \in \mathbb{N}$ such that $d(\pi(pv, qw)) < d_M$ for each $pv \in C(M)$ and $qw \in \Omega(pv)$.* □

After this general observation on the structure of infix PDA graphs we turn to infix PDA graphs of bounded tree width. Using Lemmas 1 and 6 one can construct large brambles from large fronts $\Omega(qw)$. Therefore these fronts have to be small in graphs of bounded tree width. This fact is expressed in the following lemma.

Lemma 7. *Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form such that G_M is of bounded tree width. Then there exists $c \in \mathbb{N}$ such that $|\Omega(qw)| \leq c$ for each $qw \in C(M)$.* □

This enables us to prove the desired structural characterization.

Definition 2. Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form, and let $pv, qw \in C(M)$. The graphs $G_M(pv)$ and $G_M(qw)$ are end isomorphic, denoted by $G_M(pv) \sim G_M(qw)$, iff there is a graph isomorphism between $G_M(pv)$ and $G_M(qw)$ such that vertices from $\Omega(pv)$ are mapped to vertices from $\Omega(qw)$.

Lemma 8. *Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form. If G_M is of bounded tree width, then $\sim$ has finite index.*

Proof. We show that $G_M(pv) \sim G_M(qw)$ if there is a bijection φ between $\Omega(pv)$ and $\Omega(qw)$ such that

(1) $p_1v_1 \vdash_M^\sigma p_2v_2$ iff $\varphi(p_1v_1) \vdash_M^\sigma \varphi(p_2v_2)$ for each $p_1v_1, p_2v_2 \in \Omega(pv)$, $\sigma \in \Sigma$,
(2) $\varphi(p_1v_1) \in p_1\Gamma^*$ for each $p_1v_1 \in \Omega(pv)$,
(3) if $\varphi(p_1v_1) = q_1w_1$ and $\varphi(p_2v_1) = q_2w_2$, then $w_1 = w_2$.

We extend this bijection to a graph isomorphism between $G_M(pv)$ and $G_M(qw)$. Let $p_1xv_1 \in G_M(pv)$ with $|v_1| = |v|$. Then there is $p' \in Q$ such that $p'v_1 \in \Omega(pv)$ and $p' \vdash_M^* p_1x$ (by Lemma 1). Let $p'w_1 = \varphi(p'v_1)$. Then we define $\varphi(p_1xv_1) = p_1xw_1$. It is easy to verify that φ is a graph isomorphism. Since the size of the sets $\Omega(qw)$ is bounded (Lemma 7) we can conclude that $\sim$ has finite index. □

Using the previous lemma, the proof of the following theorem is similar to the proof of Muller and Schupp [MS85] that finitely generated context free graphs are pushdown graphs.

Theorem 2. *Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form. If G_M is of bounded tree width, then G_M is a pushdown graph.* □

This leads us to our main theorem.

Theorem 3. *Let $S = (A, \Sigma, P, t_I)$ be a GTRS. Then the following statements are equivalent.*

(i) G_S is of bounded clique width.
(ii) G_S is of bounded tree width.
(iii) G_S is a pushdown graph.

Proof. The equivalence of (i) and (ii) was already stated in Corollary 1. From Proposition 6 we know that (iii) implies (ii) and, by Theorems 1 and 2, (ii) implies (iii). □

A consequence of this theorem is that graphs that are GTRS graphs and are also prefix recognizable have to be pushdown graphs since prefix recognizable graphs are of bounded clique width ([Blu01]).

6 Conclusion

The class of GTRS graphs transcends the PDA graphs by examples like the infinite grid. We have clarified the relation of this class of graphs to the class of PDA graphs and prefix recognizable graphs in terms of tree width. Future work could address the decidability of the question "Given a GTRS S, is G_S of bounded tree width?". Furthermore the relation to other classes of graphs like automatic or rational graphs should by analyzed.

I would like to thank Wolfgang Thomas for the idea to analyze GTRS graphs and the anonymous referees for several helpful comments.

References

[Bar98] Klaus Barthelmann. When can an equational simple graph be generated by hyperedge replacement? In *Proceedings of MFCS '98*, volume 1450 of *LNCS*, pages 543–552. Springer-Verlag, 1998.

[BG00] Achim Blumensath and Erich Grädel. Automatic structures. In *Proceedings of LICS '00*, pages 51–62. IEEE Computer Society Press, 2000.

[Blu01] Achim Blumensath. Prefix-recognizable graphs and monadic second order logic. Technical Report AIB-2001-06, RWTH Aachen, May 2001.

[Bra69] Walter S. Brainerd. Tree generating regular systems. *Information and Control*, 14:217–231, 1969.

[Cau92] D. Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 106(1):61–86, 1992.

[Cau96] Didier Caucal. On infinite transition graphs having a decidable monadic theory. In *Proceedings of ICALP '96*, volume 1099 of *LNCS*. Springer-Verlag, 1996.

[CG90] J.L. Coquidé and R. Gilleron. Proofs and reachability problem for ground rewrite systems. In *Aspects and Prospects of Theoretical Computer Science*, volume 464 of *LNCS*, pages 120–129. Springer, 1990.

[CO00] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.

[DHLT90] Max Dauchet, Thierry Heuillard, Pierre Lescanne, and Sophie Tison. Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems. *Information and Computation*, 88(2):187–201, October 1990.

[Die00] Reinhard Diestel. *Graph Theory*. Springer, second edition, 2000.

[DT90] Max Dauchet and Sophie Tison. The theory of ground rewrite systems is decidable. In *Proceedings of LICS '90*, pages 242–248. IEEE Computer Society Press, 1990.

[GW00] Frank Gurski and Egon Wanke. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *Proceedings of WG 2000*, volume 1928 of LNCS, pages 196–205. Springer-Verlag, 2000.

[May00] Richard Mayr. Process rewrite systems. *Information and Computation*, 156(1–2):264–286, 2000.

[Mor99] Christophe Morvan. On rational graphs. In *Proceedings of FoSSaCS '99*, volume 1784 of *LNCS*, pages 252–266. Springer, 1999.

[MS85] David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.

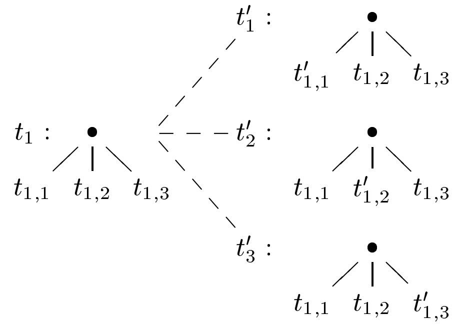
Appendix

Proof of Lemma 4

Let $m > 2 \cdot |P| + 1$ and suppose that $K_{m,m}$ is a subgraph of G_S. Then there are different trees $t_1, \dots t_m, t'_1, \dots, t'_m \in V_S$ such that $(t_i, t'_j) \in E$ or $(t'_j, t_i) \in E$ for all $i, j \in \{1, \dots, m\}$. For all $i, j \in \{1, \dots, m\}$ let $t_{i,j}, t'_{i,j} \in T_A$ and $u_{i,j} \in D_{t_i}$ such that

- $(t_i)^{u_{i,j}} = t_{i,j}$,
- $(t'_j)^{u_{i,j}} = t'_{i,j}$,
- $t_{i,j} \to t'_{i,j} \in P$ or $t'_{i,j} \to t_{i,j} \in P$.

Let $i \in \{1, \dots, m\}$. By the choice of m there must be $j_1, j_2, j_3 \in \{1, \dots, m\}$ such that u_{i,j_1}, u_{i,j_2}, and u_{i,j_3} are pairwise incomparable. For simplicity we assume that $i = 1$, $j_1 = 1$, $j_2 = 2$, and $j_3 = 3$. Then the situation is as indicated in the picture below (the dashed lines are edges in the graph G_S).



Now we want to show that it is not possible to connect another $2|P| + 1$ trees to t'_1, t'_2, and t'_3. This is done by an analysis of which of the subtrees at $u_{1,1}$, $u_{1,2}$, and $u_{1,3}$ are affected by the rewriting that generates the edge between the trees t_l and t'_1, t'_2, and t'_3.

First suppose that for each $l \in \{2, \dots, m\}$ there is a $j \in \{1, 2, 3\}$ such that in the rewriting that connects t_l and t'_j all the subtrees at $u_{1,1}$, $u_{1,2}$, and $u_{1,3}$ are affected. Since $m > 2|P| + 1$ there must be at least two of these rewritings that use the same rule. This is a contradiction because the trees t'_1, t'_2, and t'_3 only differ in the subtrees at $u_{1,1}$, $u_{1,2}$, and $u_{1,3}$ (so if the same rule is used for say t_r, t_s and all these subtrees are affected, then $t_r = t_s$).

Therefore there is an $l \in \{2, \dots, m\}$ such that for each $j \in \{1, 2, 3\}$ in the rewriting generating an edge between t_l to t'_j one of the subtrees at $u_{l,1}$, $u_{l,2}$, and $u_{l,3}$ is not affected. By a tedious distinction of cases one can show that this implies $t_{l,j} = t_{1,j}$ for each $j \in \{1, 2, 3\}$. Once again, since t'_1, t'_2, and t'_3 only differ in the subtrees at $u_{1,1}$, $u_{1,2}$, and $u_{1,3}$ we get $t_l = t_1$. $\square$

Proof of Lemma 6

For the proof of Lemma 6 we need the following lemma.

Lemma 9. *Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, \gamma_0)$ be an infix PDA in normal form, and let $w, v \in \Gamma^*$ with $w \preceq_M v$. If $pw, qxw \in C(M)$ such that there is an undirected path π without suffix changes in $G_M(pw)$ from pw to qxw, then $pv, qxv \in C(M)$, and there is a path π' without suffix changes in $G_M(pv)$ from pv to qxv that has the same depth as π.*

Proof. We show by induction on the length of π that for each vertex $p'yw$ in π the vertex $p'yv$ is in $C(M)$. Then we obtain the path π' by replacing the suffix w in the π vertices by v.

Let k be the length of π. If $k = 0$, then $pw = qxw$, and the claim obviously holds by definition of $\preceq_M$.

So assume $k \geq 1$. It is sufficient to show that $qxv \in C(M)$. For the other vertices on π the claim holds by induction. From Lemma 1 we know that there exists a $q' \in Q$ such that $q'w \in C(M)$ and $q' \vdash^*_{M_{\text{pre}}} qx$. Then $w \preceq_M v$ implies $q'v \in C(M)$, and from $q' \vdash^*_{M_{\text{pre}}} qx$, we can then conclude $qxv \in C(M)$. □

The proof of Lemma 6 goes by induction on $sc(\pi(pv, qw))$. Before we start the induction we define d_M. Let

$$D = \{d(\pi(pv, qv)) \mid pv \in C(M), qv \in \Omega(pv), \text{ and } sc(\pi(pv, qv)) = 0\}.$$

Because $sc(\pi(pv, pv)) = 0$ for each $pv \in C(M)$, the set D is not empty. We show that D is finite. If $v \approx_M w$ for $v, w \in \Gamma^*$, then, by Lemma 9, $d(\pi(pv, qv)) = d(\pi(pw, qw))$ for all $p, q \in Q$. Since Q is finite, and $\approx_M$ has finite index, the set D is finite. Thus, we can define $d_M = (max\ D) + 1$.

Now let $pv \in C(M)$, $qw \in \Omega(pv)$, and $k = sc(\pi(pv, qw))$. We show the claim by induction on k.

If $k = 0$, then $v = w$, and therefore $d(\pi(pv, qw)) \in D$. But then clearly $d(\pi(pv, qw)) < d_M$ by definition of d_M.

So consider the case $k \geq 1$. Let $p'xv'$ (with $|v'| = |v|$) be the first vertex on the path $\pi(pv, qw)$ without the suffix v. Then v' is of the form $y\gamma'z$ with $y\gamma z = v$ and $(\gamma, \sigma, \gamma') \in \Delta$ or $(\gamma', \sigma, \gamma) \in \Delta$ for some $\sigma \in \Sigma$ (remember that v can only be changed by infix rules since M is in normal form).

Case 1: $(\gamma, \sigma, \gamma') \in \Delta$. Then $v \preceq_M v'$, and therefore $pv' \in C(M)$. The predecessor of $p'xv'$ on $\pi(pv, qw)$ is $p'xv$ and there is a path from pv to $p'xv$ without suffix changes. Then there is also a path from pv' to $p'xv'$ without suffix changes (by Lemma 9). But this means that $sc(\pi(pv', qw)) < sc(\pi(pv, qw))$ and therefore $d(\pi(pv', qw)) \leq d_M$. Combining the path $\pi(pv', qw)$ and the transition $pv \vdash_{M_{\text{in}}} pv'$, we get a path from pv to qw with the same number of suffix changes as in $\pi(pv, qw)$, and this path has a depth $< d_M$. Therefore also $d(\pi(pv, qw)) < d_M$. This construction is illustrated in the picture below.

$$\begin{tikzcd}
pv \arrow[r, "\gamma\to\gamma'"] \arrow[d, dash] & pv' \arrow[d, dash] & & qw \\
p'xv \arrow[r, "\gamma\to\gamma'"] & p'xv' \arrow[urr, dash] & &
\end{tikzcd}$$

Case 2: $(\gamma', \sigma, \gamma) \in \Delta$. Then $v' \preceq_M v$. Since $p'xv' \in C(M)$, there is a $p'' \in Q$ such that $p''v' \in C(M)$, and there is a path without suffix changes from $p''v'$ to $p'xv'$ (by Lemma 1). Therefore we get $sc(\pi(p''v', qw)) < sc(\pi(pv, qw))$. By induction we get $d(\pi(p''v', qw)) < d_M$. Because $v' \preceq_M v$, we get $p''v \in C(M)$, and there is a path without suffix changes from $p''v$ to $p'xv$ (Lemma 9). But this means $sc(\pi(pv, p''v)) = 0$, and therefore $d(\pi(pv, p''v)) < d_M$. Now we combine the path $\pi(pv, p''v)$, the transition $p''v' \vdash_{M_{\text{in}}} p''v$, and the path $\pi(p''v', qw)$ and obtain a path from pv to qw with the same number of suffix changes as $\pi(pv, qw)$, and a depth $< d_M$. Therefore also $d(\pi(pv, qw)) < d_M$. This construction is illustrated in the picture below.

$$\begin{tikzcd}
pv \arrow[dr, dash] & p''v \arrow[d, dash] & p''v' \arrow[l, "\gamma\leftarrow\gamma'"'] \arrow[d, dash] & qw \arrow[dl, dash] \\
& p'xv & p'xv' \arrow[l, "\gamma\leftarrow\gamma'"'] &
\end{tikzcd}$$

□

Proof of Lemma 7

Let k be the tree width of G_S and let $l \in \mathbb{N}$. Assume that there is $qw \in C(M)$ such that $|\Omega(qw)| > |Q| \cdot l \cdot |\Gamma|^{l \cdot d_M}$ (with d_M from Lemma 6). We want to construct a bramble of width l. Then we are done because for $l \geq k+2$ we get a contradiction and therefore we know that the size of the $\Omega(qw)$ is bounded.

There are $q_1 x_1 w_1, \ldots, q_l x_l w_l \in \Omega(qw)$ with $|x_i| = l \cdot d_M$ and $w_i \neq w_j$ for $i, j \in \{1, \ldots, l\}$ with $i \neq j$. From Lemma 1 follows that there are $q_1^1, \ldots q_l^1 \in Q$ with $q_i^1 w_i \in C(M)$ and $q_i^1 \vdash^*_{M_{\text{pre}}} q_i x_i$. Let π_i be the path from $q_i^1 w_i$ to $q_i x_i w_i$ we obtain from $q_i^1 \vdash^*_{M_{\text{pre}}} q_i x_i$. For $i \in \{2, \ldots, l\}$ and $j \in \{1, \ldots, l\}$ let $q_j^i \in Q$ and $x_j^i \in \Gamma^*$ such that $q_j^i x_j^i w_j$ is on the path π_j, $q_j^l = q_j$, $x_j^l = x_j$, and $|x_j^i| = i \cdot d_M$.

Furthermore let $x_j^1 = \varepsilon$. We get the following picture:

$$\begin{array}{ccc}
q_1^1 x_1^1 w_1 & \cdots & q_l^1 x_l^1 w_l \\
\downarrow & & \downarrow \\
q_1^2 x_1^2 w_1 & \cdots & q_l^2 x_l^2 w_l \\
| & & | \\
\vdots & & \vdots \\
| & & | \\
q_1^{l-1} x_1^{l-1} w_1 & \cdots & q_l^{l-1} x_l^{l-1} w_l \\
\downarrow & & \downarrow \\
q_1^l x_1^l w_1 & \cdots & q_l^l x_l^l w_l
\end{array}$$

In this grid-like structure we define the ith row R_i and the ith column C_i as

$$\begin{aligned}
R_i &= \{pv \mid pv \in \pi(q_m^i x_m^i w_m, q_n^i x_n^i w_n) \text{ for some } m, n \in \{1, \ldots, l\}\}, \\
C_i &= \{pv \mid pv \in \pi_i\}.
\end{aligned}$$

Note that $C_i \cap C_j = \emptyset$ and $R_i \cap R_j = \emptyset$ for $i \neq j$. For the columns this follows because all the vertices in C_i have the suffix w_i, all the vertices in C_j have the suffix w_j, $|w_i| = |w_j|$, and $w_i \neq w_j$. For the rows this follows because the length of a vertex in R_i is $\geq |w_i| + i \cdot d_M + 1$ (by definition of R_i) and $< |w_i| + (i+1) \cdot d_M + 1$ (by Lemma 6).

Now we define the family $\mathcal{B} = (B_{i,j})_{i,j \in \{1,\ldots,l\}}$ of sets as

$$B_{i,j} = R_i \cup C_j.$$

From the definition of R_i and C_j it is clear that the sets $B_{i,j}$ are connected in G_M. Furthermore $B_{i_1,j_1} \cap B_{i_2,j_2} \neq \emptyset$, for $i_1, j_1, i_2, j_2 \in \{1, \ldots, l\}$, because the vertex $q_{j_2}^{i_1} x_{j_2}^{i_1} w_{j_2}$ is in R_{i_1} and in C_{j_2}. Therefore $\mathcal{B}$ is a bramble. Suppose the set S of vertices covers $\mathcal{B}$. If $|S| < l$, then there is at least one i with $R_i \cap S = \emptyset$, and at least one j with $C_j \cap S = \emptyset$. But then also $B_{i,j} \cap S = \emptyset$. Therefore $\mathcal{B}$ has width at least l. $\square$

Proof of Theorem 2

By Lemma 8 we know that $\sim$ has finite index. Let $\mathcal{G} = \{G_0, \ldots, G_n\}$ be a (minimal) set of representatives of the $\sim$-classes with $G_i = (V_i, E_i, \Sigma)$. Let Ω_i be the front of G_i.

The rough idea of the transformation is the following: The PDA will remember on the stack through which of the graphs G_i he went to reach a certain node. So, if the infix PDA is in a vertex pv, then the top stack symbol will somehow indicate to which end isomorphism class the graph $G_M(pv)$ belongs, i.e., in which

G_i we are currently. The state of the new PDA will represent the vertex on the front that *pv* corresponds to (under some end isomorphism).

There is one problem when we just use the G_i as stack symbols. In some G_i there might be a vertex on the front that has two successors u_1, u_2. These successors themselves are on the front of graphs that are end isomorphic to G_{i_1} and G_{i_2}. Now, if $i_1 = i_2 = j$, then the PDA would only remember that it went to some graph that is end isomorphic to G_j but it can not distinguish the two different graphs. For this reason we have to work with a larger set of stack symbols that arises from the "second level subgraphs" (see [MS85]).

For $i \in \{0, \ldots, n\}$ let G'_i be the subgraph of G_i that is induced by the set $V_i \setminus \Omega_i$. Then G'_i consists of connected components $\{H_{i,0}, \ldots, H_{i,m_i}\}$ (the second level subgraphs). Each of these $H_{i,j}$ is end isomorphic to a G_k. To avoid the problem sketched above, for each $i \in \{0, \ldots, n\}$, $j \in \{0, \ldots, m_i\}$ we will introduce a stack symbol $z^k_{i,j}$ if $H_{i,j} \sim G_k$. Furthermore we fix an end isomorphism $\phi^k_{i,j} : H_{i,j} \to G_k$. The initial configuration is not on the front of a second level subgraph. Therefore we will also introduce an extra initial stack symbol z_0.

As mentioned above, the states of the PDA will represent the position on the front. Thus, for each $i \in \{0, \ldots, n\}$ we fix a bijection $\psi_i : \{x_0, \ldots, x_{|\Omega_i|-1}\} \to \Omega_i$ and we let $l = (max|\Omega_i|) - 1$.

To define the initial state, we assume that $G_M(q_0\gamma_0) \sim G_0$ under an end isomorphism ϕ and that $\psi_0(x_0) = \phi(q_0\gamma_0)$, i.e., the initial vertex is mapped to the vertex of G_0 that corresponds to x_0.

Now we are ready to define a PDA $M' = (Q', \Sigma, \Gamma', \Delta', x_0, z_0)$ such that G_M and $G_{M'}$ are isomorphic.

- $Q' = \{x_0, \ldots, x_l\}$.
- $\Gamma' = \{z^k_{i,j} \mid i \in \{0, \ldots, n\}, j \in \{0, \ldots, m_i\}, H_{i,j} \sim G_k\} \cup \{z_0\}$.
- Δ' is defined as follows. The first two items treat the cases when the top stack symbol is z_0. For the general case there are three items because the stack length can remain the same, increase, or decrease.
  - $(x_r, \sigma, z_0, x_s, z_0) \in \Delta'$ iff $(\psi_0(x_r), \sigma, \psi_0(x_s)) \in E_0$.
  - $(x_r, \sigma, z_0, x_s, z^k_{0,j} z_0) \in \Delta'$ iff $((\psi_0(x_r), \sigma, u) \in E_0$ such that u is a vertex of $H_{0,j} \sim G_k$ and $\psi_k(x_s) = \phi^k_{0,j}(u)$.
  - $(x_r, \sigma, z^k_{i,j}, x_s, z^k_{i,j}) \in \Delta'$ iff $(\psi_k(x_r), \sigma, \psi_k(x_s)) \in E_k$.
  - $(x_r, \sigma, z^k_{i,j}, x_s, z^{k'}_{i',j'} z^k_{i,j}) \in \Delta'$ iff $i' = k$, $(\psi_k(x_r), \sigma, u) \in E_k$ such that u is a vertex of $H_{i',j'} \sim G_{k'}$ and $\psi_{k'}(x_s) = \phi^{k'}_{i',j'}(u)$.
  - $(x_r, \sigma, z^k_{i,j}, x_s, \varepsilon) \in \Delta'$ iff $(u, \sigma, \psi_i(x_s)) \in E_i$ such that u is a vertex of $H_{i,j} \sim G_k$ and $\psi_k(x_r) = \phi^k_{i,j}(u)$. □