Theoretical Computer Science 106 (1992) 61–86
Elsevier

On the regular structure of prefix rewriting

Didier Caucal
IRISA, Campus de Beaulieu, F35042 Rennes Cedex, France

Abstract

Caucal, D., On the regular structure of prefix rewriting, Theoretical Computer Science 106 (1992) 61–86.

We consider a pushdown automaton as a word-rewriting system with labelled rules applied only in a prefix way. The notion of pushdown transition graph is then extended to the notion of prefix transition graph generated by a word-rewriting system and accessible from a given axiom. Such accessible prefix transition graphs are context-free graphs in the sense of Muller and Schupp (1985), and we show that they are also the rooted pattern graphs of finite degree, where a pattern graph is a graph produced from a finite graph by iterating the addition of a finite family of finite graphs (the patterns). Furthermore, this characterization is effective in the following sense: any finite family of patterns generating a rooted graph G of finite degree, is mapped effectively into a word-rewriting system R such that the accessible prefix transition graph of R is isomorphic to G, and the reverse transformation is effective.

0. Introduction

A labelled rewriting system on an alphabet X and a set L of labels, is a finite subset of $X^* \times L \times X^*$. Every element (u,f,v) of $X^* \times L \times X^*$ corresponds to a labelled transition $u \xrightarrow{f} v$. One step of prefix rewriting generated by a rewriting system R is a labelled transition $uw \xrightarrow{f} vw$, where $u \xrightarrow{f} v$ is a rule of R. Prefix rewriting steps may be viewed as the arcs of a graph, called a prefix transition graph; an accessible prefix transition graph is the graph generated in this way from a given axiom. If for a rule $u \xrightarrow{f} v$, u is a letter, we say that R is alphabetic and that the corresponding accessible prefix transition graphs are alphabetic.

As an example of prefix transitions, let us briefly introduce the transitions between the configurations (without input string) of a pushdown automaton (PDA). Such a configuration may be represented as a word $qA_1...A_n$, where q is a state of the automaton and A_1 is the top of the stack $A_1...A_n$. Then the transition relation of the

0304-3975/92/$05.00 © 1992—Elsevier Science Publishers B.V. All rights reserved

PDA can be seen as a rewriting system; a transition between configurations is mapped in this way into a step of prefix rewriting. The corresponding accessible prefix transition graph is called a pushdown transition graph. In Section 1, we show that accessible prefix transition graphs coincide with pushdown transition graphs, but there exist accessible prefix transition graphs that are not alphabetic.

In a seminal paper, Muller and Schupp [14] have proved that every pushdown transition graph has a regular structure: it is a rooted graph of finite degree with a finite number of nonisomorphic connected components obtained after removing all vertices within arbitrary distances of a given vertex. Such a regular structure can be generated by a deterministic graph grammar. In Section 2, we give a procedure which produces a graph grammar generating the transition graph of a given PDA. Conversely, we will also show that any rooted graph of finite degree, generated by a deterministic graph grammar, is isomorphic to the transition graph of some PDA, and we, moreover, give a procedure which produces the corresponding PDA from the grammar. Then, and in an effective way, we show that the connected components of prefix transition graphs coincide with connected and finite-degree graphs generated by deterministic graph grammars. Furthermore, we establish that every prefix transition graph can effectively be generated by a deterministic graph grammar. Finally, we show how an arbitrary deterministic graph grammar (generating a connected graph of finite degree) can be put into a particular normal form, corresponding to the decomposition of Muller and Schupp, and we effectively obtain in this way their correspondence. As a corollary, we can decide whether two accessible prefix transition graphs (or two connected components of prefix transition graphs) are isomorphic with respect to some given vertices.

1. Prefix rewriting and pushdown automaton

In this section, we recall basic facts about rewriting systems, and introduce prefix rewriting as a special case of rewriting, constrained to operate on left factors of words. We then illustrate prefix rewriting with the help of pushdown automata (PDA) and their transitions. The transitions of a PDA are a particular case of prefix rewritings but their transition graphs are shown to be the same.

Let us first introduce notations and terminology for rewriting systems.

Definition 1.1. Given an alphabet X and a set L of *labels*, a (labelled) *rewriting system* R is a finite subset of $X^* \times L \times X^*$.

Every element (u,f,v) of $X^* \times L \times X^*$ is denoted by $u \xrightarrow{f} v$. Note that rules $\varepsilon \xrightarrow{f} v$ are allowed. A rewriting system is said to be *alphabetic* if $u \in X$ for all rules $u \xrightarrow{f} v$, and *normal* (ε-free) if both u and v have length smaller than 3 (u and v are nonempty).

Rewritings in a rewriting system are generally defined as applications of rewriting rules in every context. On the contrary, we are concerned in this paper with prefix rewriting defined as follows.

Definition 1.2. Given a rewriting system R, we define a *prefix rewriting step* $\overset{f}{\mapsto}$ for each label f as follows:

$$\overset{f}{\mapsto} := \{(uw, vw) \mid (u \xrightarrow{f} v) \in R \wedge w \in X^*\}.$$

We represent by $u \mapsto v$ ($u \overset{*}{\mapsto} v$) an elementary (unlabelled) prefix rewriting step (an arbitrary sequence of such steps).

A well known property [5] is that the set $\{w \mid r \overset{*}{\mapsto} w\}$ of words in X^* reachable by prefix rewriting from a given axiom $r \in X^*$ is a regular language, and a corresponding finite automaton is effectively constructible from R. Such an automaton is polynomially constructible in time and space [6]. From [4], we can deduce a stronger result.

Theorem 1.3. *For any rewriting system R, the prefix rewriting $\underset{R}{\overset{*}{\mapsto}}$ is a rational transduction, and a corresponding transducer is effectively constructible from R.*

The proof is given in [6]: we establish that the prefix rewriting is the componentwise concatenation of a recognizable relation with the identity relation, so is a right synchronized rational relation [10, 11].

Prefix rewriting may be seen as a way to generate labelled transition graphs: the prefix transition graph $P(R)$ is the set of prefix-rewriting steps, i.e.

$$P(R) = \{u \xrightarrow{f} v \mid u \overset{f}{\mapsto} v\} = \{xw \xrightarrow{f} yw \mid (x \xrightarrow{f} y) \in R \wedge w \in X^*\}.$$

The prefix transition graph $P(R,r)$ accessible from an axiom $r \in X^*$ is the set of prefix-rewriting steps reachable from r, i.e.

$$P(R,r) = \{u \xrightarrow{f} v \mid u \overset{f}{\mapsto} v \wedge r \overset{*}{\mapsto} u\}.$$

Figure 1 gives an example of an accessible prefix transition graph.

In the remainder of the section, we establish a strong connection between prefix rewritings and pushdown automata. To begin with, let us recast pushdown automata and their transitions in the framework of prefix rewriting.

Definition 1.4. A *pushdown automaton* (without initial and final states) is a rewriting system R satisfying the following conditions:

(i) the alphabet is partitioned into $Q_R \cup P_R$

(ii) for any rule $u \xrightarrow{f} v$ in R, we have $u \in Q_R.P_R$ and $v \in Q_R.P_R^*$.

Let R be the rewriting system on $(\{A,p,q\},\{a,b,c,d\})$ defined as follows:

$$R=\{p \xrightarrow{a} q, p \xrightarrow{b} pA, pA \xrightarrow{c} p, qA \xrightarrow{d} q\}.$$

The accessible prefix transition graph $P(R,p)$ is represented by

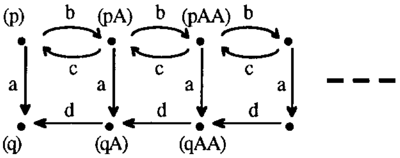


Fig. 1. Accessible prefix transition graph.

This definition corresponds to the usual definition of a pushdown automaton [15] on an input alphabet Σ when the label's set is $\Sigma\cup\{\varepsilon\}$. The language recognized by a pushdown automaton R starting at $r\in Q_R.P_R$ with acceptance on a set $F\subseteq Q_R$ of final states, is the set of label sequences $f_1 \ldots f_n$ of the paths $u_1 \overset{f_1}{\longmapsto} u_2 \cdots \overset{f_n}{\longmapsto} u_{n+1}$ such that $u_1=r$ and $u_{n+1}\in F.P_R^*$.

Of course, a pushdown automaton (PDA) works under prefix rewriting. Thus, pushdown transition graphs are certainly accessible prefix transition graphs in the following sense.

Definition 1.5. An *accessible prefix transition graph* (a *pushdown transition graph*, an *alphabetic graph*) is a graph isomorphic to $P(R,r)$ for some rewriting system R and some word r (some pushdown automaton R with r in $Q_R.P_R^*$, some alphabetic rewriting system R).

Here, a graph isomorphism is simply a vertex renaming, but the labels of the arcs are preserved. To establish the converse, i.e. every accessible prefix transition graph is a pushdown transition graph, we show that every accessible prefix transition graph is generated by a normal ε-free transition system.

Lemma 1.6. *Any pair (R,r) consisting of a rewriting system R and a word r, can effectively be normalized into another pair (S,s), where S is a normal ε-free rewriting system and s is a letter, such that $P(S,s)$ is isomorphic to $P(R,r)$.*

Proof. Let R be a rewriting system on X, and $r\in X^*$. We may suppose R ε-free and $r\neq\varepsilon$. Otherwise, we could take a letter a in X appearing neither in R nor in r, and replace (R,r) by (aR,ar) with $aR=\{au \xrightarrow{f} av \mid (u \xrightarrow{f} v)\in R\}$; so, aR is ε-free, $ar\neq\varepsilon$ and $P(aR,ar)=aP(R,r)$.

Let m be the greatest length of r and the words of X^* in R, i.e.

$$m=\max\{|u|\,|\,(u=r) \vee \exists f \exists v((u \xrightarrow{f} v)\in R \vee (v \xrightarrow{f} u)\in R)\}.$$

Let us extend an injection i from $\{u\in X^+ \mid 1\leqslant |u|\leqslant m\}$ to some given alphabet Y to an injection j from X^* to Y^* by induction

$$j(\varepsilon)=\varepsilon,$$

$$j(u)=j(v)i(w), \quad \text{where } u=vw\neq\varepsilon \wedge |w|=\min(m,|u|).$$

The rewriting system S on Y is defined as follows:

$$S=\{j(uw) \xrightarrow{f} j(vw)\,|\,(u \xrightarrow{f} v)\in R \wedge w\in X^* \wedge |w|<m\}$$

is normal and ε-free. Moreover, $s=j(r)$ is a letter. If R is alphabetic, observe that S is alphabetic. We show that

$$P(S,s)=\{j(u) \xrightarrow{f} j(v)\,|\,(u \xrightarrow{f} v)\in P(R,r)\}.$$

The proof is an easy but bothering check. □

Such a transformation is not usual and corresponds to Lemma 2.4 of [14]. Since accessible prefix transition graphs of normal and ε-free systems are pushdown transition graphs, we can transform every prefix rewriting system into a PDA without duplication, nor reduction in the accessible prefix transition graph.

Proposition 1.7. *Accessible prefix transition graphs coincide effectively with pushdown transition graphs.*

Proof. Let R be a rewriting system on X and $r\in X^*$. To show that $P(R,r)$ is a pushdown transition graph, we may assume, by Lemma 1.6, that R is a normal and ε-free system, and r is a letter. Consider the following alphabets:

$$Q=\{u(1)\,|\,(u=r) \vee \exists f \exists v((u \xrightarrow{f} v)\in R \vee (v \xrightarrow{f} u)\in R)\}$$

of the first letters of R and r, and

$$\Gamma=\{u(i)\,|\,2\leqslant i\leqslant |u| \wedge \exists f \exists v((u \xrightarrow{f} v)\in R \vee (v \xrightarrow{f} u)\in R)\}$$

of the ith letters of R with $i>1$.

Given an injection i from Γ to an alphabet P disjoint of Q, we extend i to a total injection from $Q\Gamma^*$ to QP^* as follows:

$$i(au)=ai(u(1))\ldots i(u(|u|)), \quad \text{with } a\in Q \text{ and } u\in\Gamma^*.$$

The rewriting system S on $P\cup Q$ defined by

$$S=\{i(u)\xrightarrow{f} i(v)\mid (u\xrightarrow{f} v)\in R\}$$

is a normal system. Furthermore, we show that

$$P(S,s)=\{i(u)\xrightarrow{f} i(v)\mid (u\xrightarrow{f} v)\in P(R,r)\},$$

where $s=i(r)\in Q$.

The system S is not yet a PDA because the domain $\mathrm{Dom}(S)=\{u\mid \exists v, uSv\}$ of S, must be included into $Q.P$. So, we take a new element p. The system

$$T=\{(u\xrightarrow{f} v)\in S\mid |u|=2\}\cup\{ua\xrightarrow{f} va\mid (u\xrightarrow{f} v)\in S \wedge |u|=1 \wedge a\in P_T\}$$

is a pushdown automaton with $P_T=P\cup\{p\}$ and $Q_T=Q$.
Furthermore, $P(T,sp)$ is isomorphic to $P(S,s)$, hence to $P(R,r)$. □

After Proposition 1.7, we may ask whether alphabetic rewriting systems are also representatives of arbitrary rewriting systems as far as the generated graphs are concerned. The next proposition gives a negative answer.

Proposition 1.8. *The class of alphabetic graphs is a proper subset of the class of accessible prefix transition graphs.*

Proof. Any alphabetic graph is an accessible prefix rewriting graph. But let us show that the accessible prefix transition graph of Fig. 1 is not alphabetic. Consider the following system:

$$R=\{p\xrightarrow{a} q, p\xrightarrow{b} pA, pA\xrightarrow{c} p, qA\xrightarrow{d} q\},$$

and suppose that there exist an alphabet X, an alphabetic system S on $(X,\{a,b,c,d\})$ and a word s in X^* such that $P(R,p)$ is isomorphic to $P(S,s)$ according to a bijection f. Let i be an integer. As f is injective, $f(pA^i)\neq f(pA^j)$ for every $j\neq i$. In particular $\{n\mid |f(pA^n)|=i\}$ is finite. So, there exists j such that $|f(pA^n)|\geqslant|f(q)|$ for every $n\geqslant j$. As the set $\{|f(pA^n)|\mid n\geqslant j\}$ is infinite, there is an integer m such that $|f(pA^{m+1})|>|f(pA^m)|\geqslant|f(q)|$. Set $u=f(pA^m)$, $v=f(pA^{m+1})$ and $w=f(qA^{m+1})$.

Because S is alphabetic, $|v|>|u|$ and $(v\xrightarrow{c} u)\in P(S,s)$; there is $B\in X$ with $v=Bu$. The system S being alphabetic and $(v\xrightarrow{a} w)\in P(S,s)$, there exists $x\in X^*$ such that

$w=xu$. As there exists a unique path in $P(S,s)$ from w to $f(q)$, and $|u|\geqslant|f(q)|$ in the alphabetic system S, there is an n such that $0\leqslant n\leqslant m+1$ and $f(qA^n)=u$. It follows that $f(pA^m)=f(qA^n)$. Then $pA^m=qA^n$, hence $p=q$, which is a contradiction. □

Nevertheless, in the restricted case where $P(R,r)$ has at least one coroot state (reachable from every other state), we have the following result.

Theorem 1.9. *From any pair (R,r) consisting of a rewriting system R and a word r such that the accessible prefix transition graph $P(R,r)$ has a coroot, we can decide whether $P(R,r)$ is an alphabetic graph. In this case, the pair (R,r) may be effectively transformed into a pair (S,s), where S is an alphabetic rewriting system and s is a letter, such that $P(S,s)$ is isomorphic to $P(R,r)$.*

The construction obtained with Monfort, is given in [7].

2. Prefix rewriting and pattern graph

Since, for any finite relation R on X^*, the prefix-rewriting relation $\underset{R}{\overset{*}{\mapsto}}$ generated by R is a rational transduction, prefix rewriting has a regular behaviour. In particular, the set of vertices of any accessible prefix transition graph is a regular language (over X^*). A natural question is whether the regular structure of accessible prefix transition graph is preserved when transitions are labelled, as in Section 1. The answer is positive, since those graphs are pushdown transition graphs (by Proposition 1.7), and since Muller and Schupp [14] show that pushdown transition graphs coincide with context-free graphs: a context-free graph is a rooted and finite-degree graph which has a finite number of nonisomorphic connected components obtained after removing all vertices within arbitrary distances of a given vertex. Thus, context-free graphs can be cut into slices of a finite number of "patterns".

Building up over the ideas of Muller and Schupp, we devise an effective construction of patterns for accessible prefix transition graphs. We also relax the constraint of splitting up the graph "by slices" and allow the removal of patterns of arbitrary shapes and sizes, to ease the construction of patterns. Furthermore, we establish the converse result: we give a procedure which, given any finite family of patterns (of arbitrary shapes and sizes), produces a PDA whose transition graph is obtained by pasting these patterns together (along a regular tree of formal patterns).

To begin with, let us introduce patterns and their gluing. In order to ease the presentation, we use graph grammars, and first recall their definition (for a good list of references, see [12]).

Definition 2.1. Let V be a set of *vertices* and $F=\bigcup\{F_n\mid n\geqslant 1\}$ be a graded alphabet. Every word $fv_1\ldots v_n$ of $F_n.V^n$ is a *hyperarc* labelled by f and connecting in order the vertices $v_1,\ldots,v_n$. A *hypergraph* is a set of hyperarcs.

A *graph grammar* on (F, V) is a finite set of hyperarc replacement rules $fv_1 \dots v_n \to H$, where $fv_1 \dots v_n$ is a hyperarc labelled by the *nonterminal* f, the v_i are distinct vertices and H is a finite hypergraph. Every label of a hyperarc of H which is not a nonterminal, is a *terminal* and is of arity 2.

A graph grammar is *deterministic* if there is only one rule for each nonterminal f. Figure 2 is an example of a deterministic graph grammar (a hyperarc $fv_1 \dots v_n$ is represented as a label of v_1 if $n = 1$, otherwise is represented as an arc labelled by f from v_1 to v_n with intermediate vertices $v_2, \dots, v_{n-1}$; another representation can be found in [12]).

Let us give some remarks and notations. A hypergraph has no isolated vertex. The first letter $X(1)$ of a hyperarc X is the label of X, and $V_X = \{X(2), \dots, X(|X|)\}$ is the set of vertices of X; we say that X is a *nonterminal hyperarc* if $X(1)$ is a nonterminal. There is identity between a hyperarc X and the hypergraph $\{X\}$ reduced to X. So, a graph grammar G is a binary relation on the set of (finite) hypergraphs, its *domain* $\mathrm{Dom}(G) = \{X \mid (X, H) \in G\}$ is the set of left-hand sides of its rules, and its *image* (or range) $\mathrm{Im}(G)$ is the set of right-hand sides of G. We extend by union the set of vertices of a hyperarc to the set V_H of vertices of a hypergraph H, i.e. $V_H = \bigcup \{V_X \mid X \in H\}$.

Each deterministic graph grammar defines a graph, resulting from a given start graph by iterating the graph rewriting [12, 13]. Intuitively, a rewriting step consists in choosing a nonterminal hyperarc $ft_1 \dots t_n$ whose label f indicates the rule $fs_1 \dots s_n \to H$ to be applied, and the vertices s_i in H indicate how to replace $ft_1 \dots t_n$ by H.

Definition 2.2. Given a graph grammar G on (F, V) and a hypergraph M on (F, V), M gives a hypergraph N in one rewriting step, and we denote $M \to_G N$, if there exists a nonterminal hyperarc $ft_1 \dots t_n$ of M such that

$$N = (M - \{ft_1 \dots t_n\}) \cup \{hg(x_1) \dots g(x_m) \mid hx_1 \dots x_m \in H\}$$

Let A, a, b be in F of respective arity 3, 2, 2.
Let $G = \{(A123, \{a12, a14, a25, b63, A564\})\}$ be a deterministic graph grammar.
A is the unique nonterminal of G, and G is represented as follows:

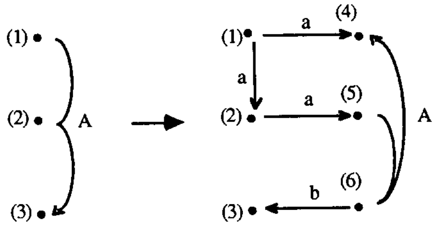


Fig. 2. Deterministic graph grammar.

for some rule $fs_1 \ldots s_n \to H$ in G and for some matching function g mapping s_i to t_i, and the other vertices of H injectively to vertices outside of M.

Note that $\to_G$ is not, in general, a functional relation, even when G is deterministic. Nevertheless, if we let $M \to_{G,X} N$ denote the rewriting of a nonterminal hyperarc X, then

$$M \to_{G,X_1} \circ \cdots \circ \to_{G,X_n} N \text{ if and only if } M \to_{G,X_{\pi(1)}} \circ \cdots \circ \to_{G,X_{\pi(n)}} N$$

for any $X_i \in M$, and for any permutation π on $\{1, \ldots, n\}$. Thus, it makes sense to define steps of complete parallel rewriting $M \Rightarrow_G N$ as follows:

$$M \Rightarrow_G N \text{ if } M \to_{G,X_1} \circ \cdots \circ \to_{G,X_n} N,$$

and M has exactly n nonterminal hyperarcs $X_1, \ldots, X_n$. One step of complete parallel rewriting corresponds to the Kleene substitution.

Henceforth, the grammar G will be *deterministic*. The infinite graph $G^\omega(M)$ generated by G starting from M is defined below, where $[M] = \{fst \in M \mid f \text{ is a terminal}\}$ is the set of terminal arcs of M.

Definition 2.3. $G^\omega(M) = \bigcup_n [G^n(M)]$, where $G^0(M) = M$ and $G^n(M) \Rightarrow_G G^{n+1}(M)$ for all n.

Since G is deterministic, $G^\omega(M)$ is unique up to hypergraph isomorphism. When M is finite, this element is called the *pattern graph* generated by G from M. Pattern graphs are the equational graphs of Bauderon [2] and Courcelle [8]. The grammar of Fig. 2 generates from $A123$ the pattern graph of Fig. 3.

Let us recall that a graph G is of *finite degree* if for every vertex s in G, the number of arcs to which s belongs is finite, and is of *bounded degree* if this number is uniformly bounded. It turns out that every finite-degree pattern graph is a bounded degree graph. A vertex r is a *root* of a graph G if each vertex of G is reachable from r. In particular, every accessible prefix transition graph $P(R, r)$ has a finite degree and root r.

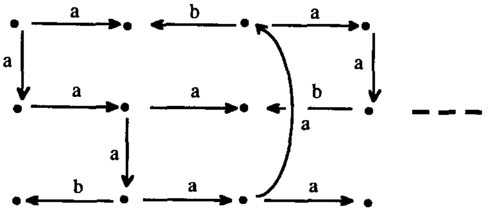


Fig. 3. Pattern graph.

To characterize prefix transition graphs as pattern graphs, we show that every pattern graph of finite degree can be generated by a normalized graph grammar.

Definition 2.4. A deterministic graph grammar G is in *standard form* if G satisfies the three following conditions:

(i) G is *proper*: for all rule (X, H) of G, every vertex of X is a vertex of a terminal arc of H,

(ii) G is in *normal form*: for all rule (X, H) of G, the vertices of every nonterminal hyperarc of H are disjoint from the X's ones,

(iii) G is *separated*: for all rule (X, H) of G, two nonterminal hyperarcs of H have no common vertex, and every nonterminal hyperarc of H has distinct vertices.

The grammar of Fig. 2 is in standard form. Furthermore, we say that G is *connected* (of *finite degree*) if for all hyperarc X in $\mathrm{Dom}(G)$, $G^{\omega}(X)$ is connected (of finite degree). Finally, G is *reduced according* to a nonterminal f if every nonterminal g is "accessible" from f, that is the hyperarc $X \in \mathrm{Dom}(G)$ such that $X(1) = f$, rewrites into a hypergraph H (i.e. $X \rightarrow^{*} H$) having a hyperarc labelled by g (i.e. there exists $Y \in H$ such that $Y(1) = g$).

Given a (deterministic) grammar generating a connected and finite-degree graph $H \neq \emptyset$, we can deduce a standard and connected grammar generating H.

Lemma 2.5. *Any pair (G, M) of a deterministic graph grammar G and a finite hypergraph M such that $G^{\omega}(M)$ is a connected and nonempty graph of finite degree, may be effectively transformed into another pair (H, N), where H is in standard form and connected, N is a hyperarc in* $\mathrm{Dom}(H)$, *and such that $H^{\omega}(N)$ is equal to $G^{\omega}(M)$.*

Proof. (i) We may assume that the set $[M]$ of terminal arcs of M is nonempty: since $G^{\omega}(M)$ is a nonempty graph, it suffices to replace M by a hypergraph H such that $M \Rightarrow^{*} H$ and $[H] \neq \emptyset$. Let $N = Ss$, where S is a new symbol in F_1 and s is a vertex of a terminal arc of M. So, $A = G \cup \{(N, M)\}$ is a grammar such that $A^{\omega}(N) = G^{\omega}(M)$.

(ii) We reduce the grammar A according to h. Let us consider the accessibility relation

$$R = \{(X(1), Y(1)) \in F' \times F' \mid \exists H, (X, H) \in A \wedge Y \in H\}$$

on the set $F' = \{X(1) \mid X \in \mathrm{Dom}(A)\}$ of the nonterminals of A. We construct the set $F'' = R^{*}(h)$ of the accessible nonterminals from $h = N(1)$, and the restricted grammar

$$B = \{(X, H) \in A \mid X(1) \in F''\}.$$

This grammar satisfies $N \in \mathrm{Dom}(B)$, $B^{\omega}(N) = A^{\omega}(N)$ and B is reduced according to $N(1)$.

(iii) Given a graph H, we consider the associated symmetric unlabelled graph:

$$s \leftrightarrow_{H} t \text{ if } \exists f, (s \xrightarrow{f} t) \in H \vee (t \xrightarrow{f} s) \in H.$$

For each hyperarc X in $\mathrm{Dom}(B)$, we want to construct the relation

$$R_X = \{(s,t) \mid s,t \in V_X \wedge s(\leftrightarrow_{B^{\omega}(X)})^+ \, t\}$$

on the vertices of X which are connected in any pattern graph generated by B from X. Let a be a new label of arity 2, and let us consider the sequence $(G_n)_{n \geqslant 0}$ of graph grammars, defined inductively as follows:

$$G_0 = \{(X, \emptyset) \mid X \in \mathrm{Dom}(B)\};$$

$$G_{n+1} = \{(X, \{ast \mid s,t \in V_X \wedge s(\leftrightarrow_H)^+ \, t\}) \mid \exists K, (X,K) \in B \wedge K \Rightarrow_{G_n} H\}.$$

For all $n \geqslant 0$, G_n is a deterministic graph grammar and

$$G_n \subseteq G_{n+1} \subseteq \{(X, \{ast \mid s,t \in R_X\}) \mid X \in \mathrm{Dom}(B)\}.$$

As the set of vertices $V_{\mathrm{Dom}(B)}$ of $\mathrm{Dom}(B)$ is finite, $h = \min\{n \mid G_n = G_{n+1}\}$ exists and G_h can be constructed polynomially in time and space. Furthermore, and for all X in $\mathrm{Dom}(B)$, we have $R_X = \leftrightarrow_{H_X}$, where $(X, H_X) \in G_h$.

(iv) We transform B into a proper grammar preserving $B^{\omega}(N)$. For instance, the following grammar $\{(S1, \{a12, P23\}), (P12, \{a13, P32\})\}$ will be transformed into the grammar $\{(S1, \{a12, Q2\}), (Q1, \{a13, Q3\})\}$ by removing the useless vertex 2 of the hyperarc $P12$. For every hyperarc X of $\mathrm{Dom}(B)$, a useless vertex of X is a vertex which does not belong to $\mathrm{Dom}(R_X)$. To each X such that $R_X \neq \emptyset$, we associate a hyperarc $\underline{X}$ labelled by a new symbol $\underline{X}(1)$ of arity equal to the cardinality $\#\mathrm{Dom}(R_X)$ of $\mathrm{Dom}(R_X)$, whose set $V_{\underline{X}} = \{\underline{X}(2), \ldots, \underline{X}(|\underline{X}|)\}$ of vertices is equal to $\mathrm{Dom}(R_X)$ and such that $\underline{X}(1) \neq \underline{Y}(1)$ if $X \neq Y$. As the vertex $N(2)$ of the axiom N is a vertex of a terminal arc, $N(2) \in \mathrm{Dom}(R_N)$ and we can identify $\underline{N}$ with N (i.e. $\underline{N}(1)$ with $N(1)$). The grammar

$$I = \{(X, \underline{X}) \mid X \in \mathrm{Dom}(B) \wedge R_X \neq \emptyset\}$$

removes the useless vertices of hyperarcs in $\mathrm{Dom}(B)$. Then, the grammar

$$C' = \{(\underline{X}, K) \mid \exists (X,H) \in B, \; R_X \neq \emptyset \wedge H \Rightarrow_I K\}$$

is deterministic and generates $C'^{\omega}(N) = B^{\omega}(N)$ because $B^{\omega}(N)$ is connected and B is reduced according to $N(1)$. Let us remark that $R_X = R_{\underline{X}}$ for all $X \in \mathrm{Dom}(B)$ such that $R_X \neq \emptyset$.

By the construction of C', for every hyperarc $X \in \mathrm{Dom}(C')$, there exists a hypergraph H_X such that $X(\Rightarrow_{C'})^+ \, H_X$ and each vertex of X is a vertex of a terminal arc of H_X. The new grammar

$$C = \{(X, H_X) \mid X \in \mathrm{Dom}(C')\}$$

is proper. Furthermore, $C^{\omega}(N) = C'^{\omega}(N) = B^{\omega}(N)$ and C is reduced according to $N(1)$.

(v) We transform C into a proper grammar in normal form. First, we replace the right-hand side H of every rule (X,H) of C by $C^{\omega}(X)$ when the set $V_{C^{\omega}(X)}$ of vertices of $C^{\omega}(X)$ is restricted to V_X. For instance, the grammar $\{(S1, \{a12, A12\}), (A12, \{a12, A21\})\}$ is replaced by the grammar $\{(S1, \{a12, A12\}), (A12, \{a12, a21\})\}$. As C is reduced according to $N(1)$ and $C^{\omega}(N)$ is of finite degree, C is of finite degree. It follows that there exists H_X such that $X(\Rightarrow_C)^+ H_X$ and every vertex of X is not a vertex of a nonterminal hyperarc of H_X. Then the grammar

$$D=\{(X,H_X)\mid X\in\mathrm{Dom}(C)\}$$

is in normal form. Furthermore, $D^\omega(N)=C^\omega(N)$ and the grammar D is proper and reduced according to $N(1)$ because C is. Note that R_X is unchanged for every X in $\mathrm{Dom}(D)=\mathrm{Dom}(C)$.

(vi) We transform D into a connected proper grammar in normal form, preserving $D^\omega(N)$. After a possible renaming (and adding new rules), we may suppose that every hypergraph of $\mathrm{Im}(D)$ does not have two nonterminal hyperarcs with the same label. For instance, the following proper grammar in normal form $\{(S1,\{a12,a13,P23\}),(P12,\{a13,a24,a25,P34,P35\})\}$ is replaced by the grammar $D_1=\{(S1,\{a12,a13,P23\}),$ $(P12,\{a13,a24,a25,P34,Q35\}),$ $(Q12,\{a13,a24,a25,P34,Q35\})\}$. With all $X\in\mathrm{Dom}(D)$ and each class P of $U_X=\{R_X(s)\mid s\in\mathrm{Dom}(R_X)\}$, we associate, as in (iv), a hyperarc X_P labelled by a new symbol $X_P(1)$ of arity $\# P$, whose set V_{X_P} of vertices is equal to P, and such that $X_P(1)\neq Y_Q(1)$ if $(X,P)\neq(Y,Q)$. As the vertex $N(2)$ of the axiom N is a vertex of a terminal arc of M, $U_N=\{\{N(2)\}\}$ and we can identify $N_{\{N(2)\}}(1)$ with $N(1)$.

Consider the grammar

$$J=\{(X,\{X_P\mid P\in U_X\})\mid X\in\mathrm{Dom}(D)\},$$

which splits each $X\in\mathrm{Dom}(D)$ into hyperarcs according to U_X. This grammar allows one to split the nonterminal hyperarcs into a disjoint union of rules in D, that is to say for each rule (X,H) of D, we associate a hypergraph H_X such that $H\Rightarrow_J H_X$. In fact, each part X_P of the splitting of X will generate only the part of H_X accessible from P. The restriction of a hypergraph H to a set V of vertices is denoted by $H_{|V}=\{fs_1\ldots s_n\in H\mid s_1,\ldots,s_n\in V\}$. Then the grammar

$$E=\{(X_P,H_{X|V})\mid X\in\mathrm{Dom}(D)\wedge P\in U_X\wedge V=(\leftrightarrow_{H_X})^+(P)\}$$

satisfies the following property

$$J\circ\Rightarrow_E=D\circ\Rightarrow_J.$$

For instance, from grammar D_1, we obtain the connected grammar $E_1=\{(S1,\{a12,a13,P'2,P''3\}),$ $(P'1,\{a13,P'3,Q'3\}),$ $(Q'1,\{a13,P'3,Q'3\}),$ $(P''2,\{a24,a25,P''4,Q''5\}),$ $(Q''2,\{a24,a25,P''4,Q''5\})\}$. Let T be the set of finite hypergraphs H such that two nonterminal (according to D) hyperarcs of H with the same label, have no common vertex. As every hypergraph of $\mathrm{Im}(D)$ does not have two nonterminal hyperarcs with the same label, $\mathrm{Im}(D)$ is in particular included into T. So, we have

$$\mathrm{id}_T\circ\Rightarrow_J\circ\Rightarrow_E=\mathrm{id}_T\circ\Rightarrow_D\circ\Rightarrow_J,$$

where $\mathrm{id}_T=\{(H,H)\mid H\in T\}$ is the identity on T.

As D is a proper grammar in normal form such that $\mathrm{Im}(D) \subseteq T$, the relation $\mathrm{id}_T \circ (\Rightarrow_D)^*$ is included into $T \times T$. So, by induction on $n \geqslant 0$, we have

$$\mathrm{id}_T \circ \Rightarrow_J \circ (\Rightarrow_E)^n = \mathrm{id}_T \circ (\Rightarrow_D)^n \circ \Rightarrow_J .$$

In particular, $E^\omega(N) = D^\omega(N)$.

Furthermore, E is a proper grammar in normal form, and reduced according to $N(1)$ because D is. As $E^\omega(N) = G^\omega(M)$ is connected and E is reduced according to $N(1)$, by construction, E is a connected grammar.

(vii) We transform E into a separated grammar. For each rule (X, H) of E, we extract the set $W_X = \{K \mid \exists H, (X, H) \in E \wedge K$ is a connected component of $H - [H]\}$ of connected components of the nonterminal hyperarcs of H, where $X \rightarrow H$ is a rule of E.

For all $K \in W_X$, we associate a hyperarc X_K labelled by a new symbol $X_K(1)$ of arity $\# V_K$, $V_{X_K} = V_K$, and such that $X_K(1) \neq X_{K'}(1)$ if $K \neq K'$. Let us consider the grammar

$$L' = \{(X, [H] \cup \{X_K \mid K \in W_X\}) \mid (X, H) \in E\}$$

obtained from E by replacing in all rules (X, H) of E, each hypergraph K of W_X in H by X_K. For all $K \in W_X$, we associate a hypergraph $H_{X,K}$ such that $K \Rightarrow_{L'} H_{X,K}$. The grammar

$$L = L' \cup \{(X_K, H_{X,K}) \mid X \in \mathrm{Dom}(E) \wedge K \in W_X\}$$

is separated, and satisfies $L^\omega(N) = E^\omega(N)$. Like grammar E, this grammar L is connected, proper and in normal form. Finally, L is a connected grammar in standard form which generates from $N \in \mathrm{Dom}(L)$ the pattern graph $G^\omega(M)$. □

The next step is to translate a grammar in standard form generating a rooted graph G of finite degree, into a rewriting system generating G by prefix rewritings.

Proposition 2.6. *Any triple (G, M, v) of a deterministic graph grammar G, a finite hypergraph M and a vertex v of M, such that $G^\omega(M)$ has finite degree and root v, may be effectively transformed into a pair (R, r) of a word-rewriting system R and a word r, such that the corresponding graphs $G^\omega(M)$ and $P(R, r)$ are isomorphic.*

Proof. From Lemma 2.5, we can assume that G is a connected grammar in standard form, and M is a hyperarc. After a possible renaming of labels (and provided, we add new rules), we further suppose that every hypergraph of $\mathrm{Im}(G)$ does not have two nonterminal hyperarcs with the same label.

Let N be the set of nonterminals of G, and V be the set of vertices of G. For convenience, a vertex of a nonterminal hyperarc is called an output. With each rule (X, H) of G, we associate a total function p_X from V_H to $V \cup V.N$, which is the identity on the set of nonoutput vertices of H. For any output vertex s of H, we have $p_X(s) = T(i)T(1)$, where T is the nonterminal hyperarc in the domain of G with the same label as the nonterminal hyperarc Y whose vertex is s, and i is the place of s in Y, i.e.

$$p_X(s)=s \qquad \text{for } s\in V_H \text{ such that } s\notin V_J \text{ for all } J\in H \text{ such that } J(1)\in N,$$

$$p_X(s)=T(i)T(1) \quad \text{if there exist } Y\in H \text{ and } T\in \mathrm{Dom}(G) \text{ such that } Y(i)=s \text{ and } T(1)=Y(1).$$

Since G is separated, p_X is well-defined.

Let $R(G)$ be the rewriting system on $N\cup V$ and labelled in F, defined by

$$R(G)=\{p_X(s).X(1)\xrightarrow{a}p_X(t).X(1)\mid \exists H,\ (X,H)\in G\wedge(s\xrightarrow{a}t)\in H\wedge a\notin N\}.$$

The vertices w of $P(R(G),v.M(1))$ are words $w_0n_p...n_1$, where w_0 is the vertex of the "pattern" P which first introduced w as nonoutput vertex, and $n_p...n_1$ is the label sequence of the nonterminal hyperarcs whose rewritings have given P. Then $P(R(G),v.M(1))$ is isomorphic to $G^{\omega}(M)$. □

Applied to the grammar of Fig. 2, the construction of Proposition 2.6 gives the following rewriting system

$$R=\{1A\xrightarrow{a}2A,1A\xrightarrow{a}3AA,2A\xrightarrow{a}1AA,2AA\xrightarrow{b}3A\}.$$

Hence, $P(R,1A)$ is the pattern graph of Fig. 3.

The converse of Proposition 2.6 is true: in an effective way, every accessible prefix transition graph is a rooted pattern graph of finite degree.

Proposition 2.7. *Any pair (R,r) of a word-rewriting system R and a word r, may be effectively transformed into a pair (G,M) of a deterministic graph grammar G and a hyperarc M, such that the corresponding graphs $P(R,r)$ and $G^{\omega}(M)$ are isomorphic.*

Proof. From Lemma 1.6, we may suppose that R is normal and ε-free on X, and $r\in X$. The grammar G to be constructed generates $P(R,r)$ by vertices of increasing length.

We consider the connected component $P(R,r)_{|u}$ of $P(R,r)$ restricted to the vertices of length at least $|u|$, and containing u.

We can determine the set $V(u)$ of vertices $P(R,r)_{|u}$ of length $|u|$. From [5] or [6], we can construct an automaton recognizing the set $\underset{R}{\overset{*}{\mapsto}}(r)=\{v\mid r\underset{R}{\overset{*}{\mapsto}}v\}$ of vertices of $P(R,r)$. So, we can determine the finite set $D(u)=\underset{R}{\overset{*}{\mapsto}}(r)\cap X^{|u|}$ of vertices of $P(R,r)$ of the same length as u. To decide if two elements in $D(u)$ are connected in the restriction of $P(R,r)$ to the vertices having length $\geqslant|u|$, we construct the unlabelled rewriting system S, defined by

$$S=\{(xz,yz)\mid |z|=\max(0,|u|-|x|)\wedge\exists f(x\xrightarrow{f}y)\in R\},$$

and we determine the following relation E on $D(u)$:

$$E=\{(x,y)\mid x,y\in D(u)\wedge(\overset{*}{\underset{s}{\mapsto}}(x)\cap\overset{*}{\underset{s}{\mapsto}}(y))-\{z\mid |z|<|u|\}\neq\emptyset\}.$$

So, $V(u)$ is the class in the partition of $D(u)$ by the equivalence E^*, containing u, i.e.

$$V(u)=E^*(u)=\{v\mid uE^*v\}.$$

Furthermore, consider the symmetric system T of unlabelled word rewritings on $X^{|u|}X^*$, defined by

$$T=\{(xz,yz)\mid |z|=\max(0,|u|-|x|,|u|-|y|)$$
$$\wedge\ \exists f((x\xrightarrow{f}y)\in R\vee(y\xrightarrow{f}x)\in R)\}.$$

So, the set of vertices of $P(R,r)_{|u}$ is included in $\{v\mid u\overset{*}{\underset{T}{\mapsto}}v\}$, i.e. $V_{P(R,r)_{|u}}\subseteq\overset{*}{\underset{T}{\mapsto}}(u)$.

As R is normal, the set of vertices of $P(R,r)_{|u}$ have a common suffix s_u of length $\max(0,|u|-2)$. We denote by $V.u^{-1}=\{v\mid vu\in V\}$ the right quotient of a language V by a word u.

Two vertices u and v of $P(R,r)$ are equivalent, denoted as $u\equiv v$, if $V(u).s_u^{-1}=V(v).s_v^{-1}$. If $u\equiv v$ then $P(R,r)_{|u}$ is isomorphic to $P(R,r)_{|v}$. Moreover, the equivalence $\equiv$ is of finite index and a set U of representatives is constructible from (R,r) with $r\in U$. For any $u\in U$, we associate the graph H_u of arcs of $P(R,r)_{|u}$ with a vertex of length $|u|$. To construct the grammar G, we only add to each H_u a set K_u of nonterminal hyperarcs which generates, according to G, the graph $P(R,r)_{|u}$ restricted to vertices of length strictly greater than $|u|$.

To this end, we take a graded alphabet F disjoint from the label set of R, and an injection j from U to the set of hyperarcs labelled by F with vertices in X^*, such that for every u in U, we have

$$j(u)=fs_1\ldots s_n\quad\text{with }\{s_1,\ldots,s_n\}=V(u),\ s_i\neq s_j\text{ if }i\neq j,\ f\neq j(v)(1)\text{ if }v\in U-\{u\}.$$

For any $u\in U$, we define

$$K_u=\{g(t_1s_w)\ldots(t_ns_w)\mid w\in V_{P(R,r)_u}\wedge|w|=|u|+1\wedge\exists v\in U,\ v\equiv w$$
$$\wedge\ j(v)=g(t_1s_v)\ldots(t_ns_v)\},$$

and we define the following deterministic graph grammar:

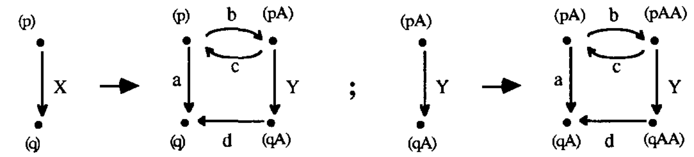
$$G=\{(j(u),H_u\cup K_u)\mid u\in U\}.$$

Note that G is a connected and finite-degree grammar in standard form.

For any $u\in U$, $P(R,r)_{|u}$ is isomorphic to $G^\omega(j(u))$ and, in particular, for $M=j(r)$ $P(R,r)$ is isomorphic to $G^\omega(M)$. □

The construction of Proposition 2.7 is illustrated in Fig. 4.

Applied to the pair (R,p) of Fig. 1, the construction of Proposition 2.7 gives the following grammar G:



The accessible prefix transition graph $P(R,p)$ is isomorphic to the pattern graph $G^{\omega}(Xpq)$.

Fig. 4. Extraction of a graph grammar from an accessible prefix transition graph.

A restricted version of Proposition 2.7 was established in [1] for alphabetic graphs with a coroot of out-degree zero. Propositions 2.6 and 2.7 establish *constructively* the following statement.

Theorem 2.8. *Accessible prefix transition graphs coincide effectively with rooted pattern graphs of finite degree.*

After Propositions 2.6 and 2.7, we can determine a word-rewriting system for the inverse of any accessible prefix transition graph with a coroot.

Proposition 2.9. *Any triple (R,r,c) consisting of a rewriting system R, a word r and a coroot c of $P(R,r)$, may be effectively transformed into another triple (S,s,d) such that there exists an isomorphism f from $P(S,s)$ to the inverse of $P(R,r)$ satisfying $f(s)=c$ and $f(d)=r$.*

Proof. After renaming, we can suppose that any two rules in R have not the same label, so the coroot c is unambiguously determined by a path from r to c. From Proposition 2.7, we can transform (R,r) into (G,M), where G is a deterministic graph grammar, and M is a finite hypergraph such that $P(R,r)$ is a pattern graph $G^{\omega}(M)$ generated by G from M. Provided we rewrite M a suitable number of times, we can assume that it contains r and c. Let h be a new label of arity 2 and let $G'=G\cup\{(hrc,M)\}$. We construct a grammar H by inverting the arcs of the right members of the rules in G', i.e.

$$H=\{(X,K^{-1})\mid(X,K)\in G'\}.$$

So, $H^{\omega}(hrc)$ is the inverse graph of $G^{\omega}(M)$, where c is a root and r is a coroot. Consequently, by restituting the old labels, the system S constructed in Proposition 2.6 satisfies the requirements with $s=ch$ and $d=rh$. □

Propositions 2.6 and 2.7 allow the study of other effective transformations of prefix-rewriting systems, not only the computation of the inverse as in Proposition 2.9. One of these transformations allows an effective extension of Theorem 2.8 to the connected components of prefix transition graphs. The connected component of a prefix transition graph $P(R)$ containing a vertex r, is denoted by $P(R)_r$, i.e.

$$P(R)_r=\{u \xrightarrow{f} v \mid r(\leftrightarrow_{P(R)})^* u \wedge u \overset{f}{\mapsto} v\}.$$

Such a class of graphs is extended up to isomorphism.

Definition 2.10. A *prefix transition graph* (a *connected prefix transition graph*, a *rooted prefix transition graph*) is a graph isomorphic to $P(R)$ ($P(R)_r$, $P(R)_r$ having a root) for some rewriting system R (and some word r).

Figure 5 gives an example of a connected prefix transition graph. Propositions 2.6 and 2.7 help to prove the following statement.

Theorem 2.11. *Connected prefix transition graphs coincide effectively with connected pattern graphs of finite degree.*

Proof. (i) Let us consider a word-rewriting system R on X labelled in L, and a word $r \in X^*$. With each symbol f of L, we associate injectively a symbol f' in a new set L', and we define a word-rewriting system S on X labelled in $L \cup L'$, as follows:

$$S=R \cup \{v \xrightarrow{f'} u \mid (u \xrightarrow{f} v) \in R\}.$$

Then $P(R)_r$ is the graph obtained from $P(S,r)$ by reversing the arcs labelled in L', i.e.

$$P(R)_r=\{(u \xrightarrow{f} v) \in P(S,r) \mid f \in L\} \cup \{u \xrightarrow{f} v \mid (v \xrightarrow{f'} u) \in P(S,r) \wedge f' \in L'\}.$$

Let R be the rewriting system on $(\{A,B,C\},\{a,b,c,d\})$ defined as follows:

$$R=\{A \xrightarrow{a} AA, A \xrightarrow{b} B, C \xrightarrow{c} B, C \xrightarrow{d} CA\}.$$

The connected component $P(R)_A$ of $P(R)$ containing A, is represented by

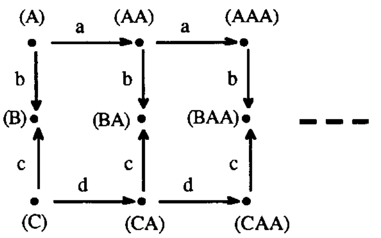


Fig. 5. Connected prefix transition graph.

From Proposition 2.7, the pair (S,r) may be effectively transformed into a deterministic graph grammar G and a hyperarc M in $\mathrm{Dom}(G)$ such that $P(S,r)$ is a pattern graph generated by G from M. In the patterns of G, we reverse the terminal arcs whose labels are in L', that is to say, we consider the following deterministic graph grammar H:

$$H=\{(X,\{Y\in K \mid Y(1)\notin L'\}\cup\{u\xrightarrow{f} v \mid (v\xrightarrow{f'} u)\in K \wedge f'\in L'\})\mid (X,K)\in G\}.$$

Then $P(R)_r$ is isomorphic to $H^{\omega}(M)$.

(ii) Let us consider a deterministic graph grammar G and a finite hypergraph M such that $G^{\omega}(M)$ is connected and of finite degree. The empty graph is a connected prefix transition graph; otherwise, we can assume that $G^{\omega}(M)$ is not the empty graph. From Lemma 2.5, we can assume that G is a connected grammar in standard form and M is a hyperarc in $\mathrm{Dom}(G)$. Then, the rewriting system $R(G)$ in the proof of Proposition 2.6 is suitable because $P(R(G))_{v.M(1)}$ is isomorphic to $G^{\omega}(M)$ for any vertex v of M. So, every connected pattern graph of finite degree is effectively a connected prefix transition graph. □

As a consequence, the rooted components of prefix transition graphs are the accessible prefix transition graphs.

Corollary 2.12. *Rooted prefix transition graphs coincide effectively with accessible prefix transition graphs.*

Proof. (i) Let $P(R)_u$ be a connected prefix transition graph with a root r. Then $P(R)_u=P(R,r)$ is an accessible prefix transition graph.

(ii) Let $P(R,r)$ be an accessible prefix transition graph. Note that the root r of $P(R,r)$ is not necessarily a root of $P(R)_r$, i.e. we can have $P(R,r)\neq P(R)_r$. By Theorem 2.8, $P(R,r)$ is effectively a rooted pattern graph of finite degree. By Theorem 2.11, $P(R,r)$ is effectively a connected prefix transition graph. □

Finally, we consider prefix transition graphs. Every finite graph is a pattern graph of finite degree which is not a prefix transition graph because (nonempty) prefix transition graphs are infinite. Nevertheless, and from Theorem 2.11, the converse is true.

Theorem 2.13. *Prefix transition graphs are effectively pattern graphs of finite degree.*

Proof. We first give a construction, and in the following, steps prove the assertion of the theorem.

(i) Let R be a word-rewriting system on X with labels in L. Let W be the set of words in R, i.e.

$$W=\{u \mid \exists v, \exists f, (u\xrightarrow{f} v)\in R \vee (v\xrightarrow{f} u)\in R\},$$

and we denote by $m=\max\{|u| \mid u\in W\}$ the maximum length of the words in R.

Recall that the symmetric closure of the (unlabelled) prefix-rewriting step is denoted by $\leftrightarrow$, i.e.

$$\leftrightarrow = \{(u,v) \mid u \mapsto v \vee v \mapsto u\},$$

and the equivalence $(\leftrightarrow)^*$ is preserved by right concatenation. We have $V_{P(R)} = \mathrm{Dom}(\leftrightarrow) = W.X^*$. We consider the set $E = \{uw \mid u \in W \wedge |uw| \leqslant m\}$ of vertices of $P(R)$ having length at most m, and we construct $E/(\leftrightarrow)^* = \{C_1, \ldots, C_p\}$. We choose an element r_i of C_i of minimal length. From Theorem 2.11 and Lemma 2.5, we can construct a grammar G_i in standard form (and connected) and a hyperarc N_i in $\mathrm{Dom}(G_i)$ such that $P(R)_{r_i}$ is isomorphic to $G_i^\omega(N_i)$. After a possible renaming, we may assume that the G_i have distinct nonterminals and vertices. Take two new unary symbols A and B, and a new vertex v. We construct the following "repetitive" grammar H:

$$H = \{(Bv, \{Bv\} \cup \{N_i \mid 1 \leqslant i \leqslant p \wedge |r_i| = m\})\}$$

and the grammar G as follows:

$$G = \{(Av, \{Bv\} \cup \{N_i \mid 1 \leqslant i \leqslant p\})\} \cup H \cup \bigcup\{G_i \mid 1 \leqslant i \leqslant p\}.$$

To prove this theorem, it is enough to show that $P(R)$ is isomorphic to $G^\omega(Av)$.

With each $1 \leqslant i \leqslant p$, we associate the following family

$$V_i = \{C \in V_{P(R)}/(\leftrightarrow)^* \mid \exists c \in \mathrm{Min}(C), p_c = r_i\},$$

where $\mathrm{Min}(C)$ is the set of words in C of minimal length and p_c is the prefix of the word c of length $\min(m, |c|)$.

To prove that $P(R)$ is generated by G from Av, it suffices to prove successively:

(1) $\bigcup\{V_i \mid 1 \leqslant i \leqslant p\}$ is a partition of $V_{P(R)}$,

(2) $P(R)_d$ is isomorphic to $P(R)_{r_i}$ for any $d \in C \in V_i$,

(3) if $|r_i| < m$ then $\# V_i = 1$ else $\# V_i = \infty$.

(ii) We shall need the following three claims.

Claim 1: $|r_i| = m \wedge u \leftrightarrow^* r_i v \Rightarrow \exists w, w \leftrightarrow^* r_i \wedge u = wv$.

As $|r_i| = m$ and by minimality of $|r_i|$, $|c| \geqslant m$ for every c in $[r_i] = \{u \mid r_i \leftrightarrow^* u\}$. By definition of m, if $r_i v \leftrightarrow^* u$ then there exists w such that $r_i \leftrightarrow^* w$ and $wv = u$.

Claim 2: $u \leftrightarrow^* r_i \wedge \exists v, uv \in \mathrm{Min}([uv]) \Rightarrow |u| = |r_i|$.

As $u \leftrightarrow^* r_i$ and by minimality of $|r_i|$, we have $|r_i| \leqslant |u|$. As $\leftrightarrow^*$ is closed by right concatenation, $uv \leftrightarrow^* r_i v$. So, $|uv| \leqslant |r_i v|$; hence, $|u| \leqslant |r_i|$. In consequence, $|u| = |r_i|$.

Claim 3: $C \in V_i \Leftrightarrow \exists c \in \mathrm{Min}(C)$, $p_c \leftrightarrow^* r_i$.

The necessary condition follows immediately from the definition of V_i. Conversely, let $c \in \mathrm{Min}(C)$ such that $p_c \leftrightarrow^* r_i$. Let c' be the suffix of c such that $c = p_c.c'$. As $\leftrightarrow^*$ is preserved by right concatenation, $c \leftrightarrow^* r_i.c'$. By Claim 2, $|p_c| = |r_i|$; hence, $r_i.c' \in \mathrm{Min}(C)$. Furthermore, $c' = \varepsilon$ if $|r_i| < m$; so, $p_{r_i.c'} = r_i$. Finally, $C \in V_i$.

(iii) **Proof of** (1). Let C be an equivalence class of $V_{P(R)}$ according to $\leftrightarrow^*$, and we want to show that there exists a unique i such that $C \in V_i$.

Existence: Let $c \in \mathrm{Min}(C)$. As $c \in V_{P(R)}$, $p_c \in V_{P(R)}$. Furthermore, $|p_c| \leqslant m$; hence $p_c \in E$. So, there exists (a unique) $1 \leqslant i \leqslant p$ such that $p_c \leftrightarrow^* r_i$; and, by Claim 3, $C \in V_i$.

Unicity: Let $c, d \in \mathrm{Min}(C)$ such that $p_c = r_i$ and $p_d = r_j$. We have $c \leftrightarrow^* d$ and $|c| = |d|$.

If $|c| \leqslant m$ then $p_c = c$ and $p_d = d$; then $p_c \leftrightarrow^* p_d$ and, hence, $r_i \leftrightarrow^* r_j$.

If $|c| > m$ then $|r_i| = |p_c| = m = |r_j| = |p_d|$. Let c' be such that $c = p_c.c'$. So, $r_i.c' \leftrightarrow^* d$ and, by Claim 1, there exists w such that $w \leftrightarrow^* r_i$ and $d = w.c'$. By Claim 2, $|w| = m$; then $w = p_d$ and, so, $r_i \leftrightarrow^* r_j$.

In all cases $r_i \leftrightarrow^* r_j$; hence $i = j$.

(iv) **Proof of** (2). Let $C \in V_i$: there exists $c \in \mathrm{Min}(C)$ such that $p_c = r_i$.

If $|c| \leqslant m$ then $p_c = c$. So, $c = r_i$; then $P(R)_c = P(R)_{r_i}$.

If $|c| > m$ then $|r_i| = |p_c| = m$. Let c' be such that $c = p_c.c'$. By Claim 1, we obtain

$$P(R)_c = \{u.c' \xrightarrow{f} v.c' \mid (u \xrightarrow{f} v) \in P(R)_{r_i}\}$$

which is isomorphic to $P(R)_{r_i}$.

(v) **Proof of** (3). Suppose that $\# V_i \neq 1$ and we show that $|r_i| = m$. As $[r_i] \in V_i$, there exists $C \in V_i$ such that $C \neq [r_i]$. So, there exists $c \in \mathrm{Min}(C)$ such that $p_c = r_i$. Since $C \neq [r_i]$, $p_c \neq c$; hence, $|c| > m$. So, $|r_i| = m$. Therefore, if $|r_i| < m$ then $\# V_i = 1$.

Suppose that $|r_i| = m$ and we show that $\# V_i = \infty$. Let $v \in X^*$ and $c \in \mathrm{Min}([r_i v])$. As $c \leftrightarrow^* r_i v$, $|r_i| = m$ and, by Claim 1, there exists w such that $w \leftrightarrow^* r_i$ and $c = wv$. By Claim 2, $|w| = |r_i| = m$. Therefore, $w = p_c$ and, hence, $p_c \leftrightarrow^* r_i$. By Claim 3, $[r_i v] \in V_i$ for any $v \in X^*$. Furthermore, if $[r_i u] = [r_i v]$ then $r_i u \leftrightarrow^* r_i v$ and, by Claim 1, we obtain $u = v$. Eventually $\# V_i = \infty$ if $|r_i| = m$. □

The construction of Theorem 2.13 is illustrated in Fig. 6.

From Theorem 2.13 and [8], every closed monadic second-order formula of prefix-rewriting step is decidable.

Corollary 2.14. *The monadic second-order theory of the prefix-rewriting step on words is decidable.*

For instance, the termination and the confluence of the prefix-rewriting step is decidable (short proofs can be found in [5]).

We shall now reconsider the result of Muller and Schupp [14]. The next definition translates their notion of finite decomposition into the framework of generating grammars.

Definition 2.15. A *uniform* grammar is a connected graph grammar in standard form in which all rule $fs_1 \ldots s_n \to H$ of G satisfies the additional conditions:

(i) every terminal arc of H goes through at least one s_i,

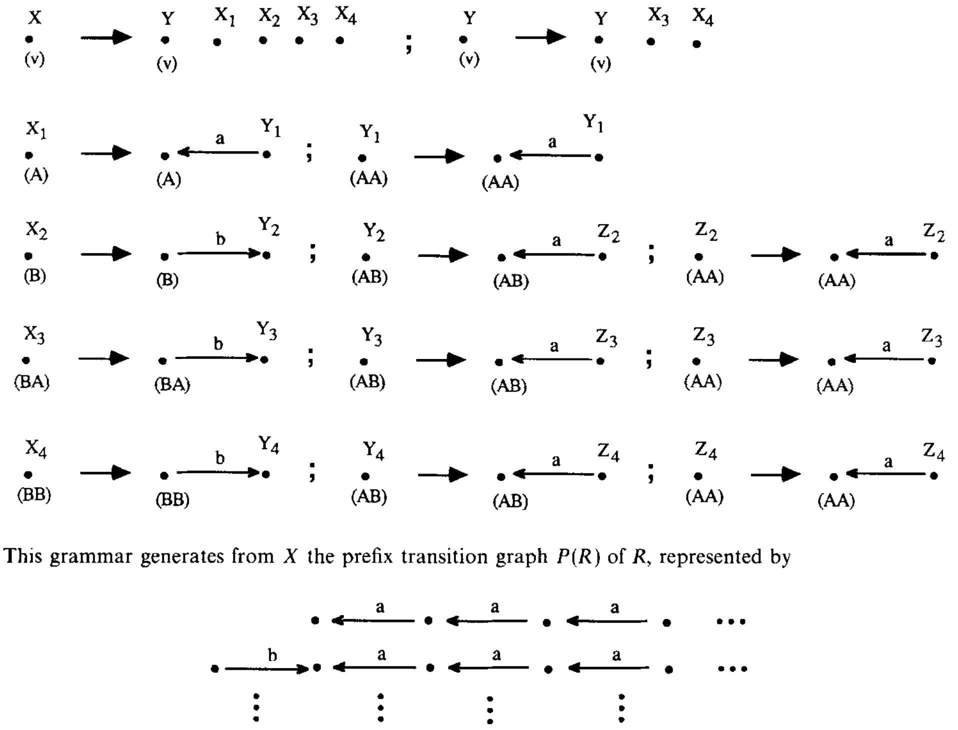
(ii) every vertex of a nonterminal hyperarc of H also belongs to a terminal arc of H.

For instance, the grammar of Fig. 2 is uniform. It is obvious to see that *context-free* graphs, defined in [14], are the rooted pattern graphs generated by uniform grammars. So, a context-free graph is a finite-degree pattern graph. Our goal is to prove constructively Theorem 2.6 of [14].

Let R be the rewriting system on $(\{A,B\},\{a,b\})$ defined as follows:

$$R=\{AA \xrightarrow{a} A, B \xrightarrow{b} AB\}.$$

The construction of Theorem 2.13 gives the set $E/\overset{*}{\leftrightarrow} = \{\{A,AA\},\{B,AB\},\{BA\},\{BB\}\}$ and the grammar



This grammar generates from X the prefix transition graph $P(R)$ of R, represented by

Fig. 6. Extraction of a graph grammar from a prefix transition graph.

Theorem 2.16. *Context-free graphs coincide effectively with pushdown transition graphs.*

From Propositions 2.6 and 1.7, to every uniform grammar corresponds effectively a pushdown automaton whose transition graph is generated by the grammar. The converse follows from Proposition 2.7 and from the proposition below, which states effectively that every finite-degree and connected pattern graph can be generated by a uniform grammar.

Proposition 2.17. *Any pair (G,M) of a deterministic graph grammar G and of a finite hypergraph M such that $G^{\omega}(M)$ is a connected and nonempty graph of finite degree, may be effectively transformed into a uniform grammar H such that $H^{\omega}(M)$ is equal to $G^{\omega}(M)$.*

Proof. From Lemma 2.5, we may assume that G is in standard form and connected, and that M is a hyperarc in $\mathrm{Dom}(G)$. After a possible renaming of labels (and provided, we add new rules), we further suppose that every hypergraph of $\mathrm{Im}(G)$ does not have two nonterminal hyperarcs with the same label. Finally, and after a possible renaming of vertices, we may assume that hypergraphs of $\mathrm{Im}(G)$ have no common vertex.

Let V be the vertex set of the right members of G and N be the set of nonterminals. We consider a total order $\leqslant$ on $V.N^*$ preserved by right concatenation, and for all graph C with vertices in $V.N^*$ and every nonterminal word u, we write

$$C.u=\{f(s_1u)\dots(s_nu)\mid fs_1\dots s_n\in C\}$$

the suffixing of the vertices of C by u,

$$C.u^{-1}=\{fs_1\dots s_n\mid f(s_1u)\dots(s_nu)\in C\}$$

the right quotient of C by u, and s_C the greater common suffix in N^* of vertices in C.

We need a representative $R(G,M)$ of the pattern graphs $G^{\omega}(M)$ where vertices have canonical names: $R(G,M)$ is the connected prefix transition graph $P(R)_{M(2)}$ containing a vertex of M (for instance, the first vertex $M(2)$ of M), where R is the following rewriting system:

$$R(G)=\{p_X(s)\xrightarrow{a}p_X(t)\mid \exists H,(X,H)\in G\wedge(s\xrightarrow{a}t)\in H\wedge a\notin N\},$$

where p_X is defined as in Proposition 2.6. We check that $R(G,M)$ is $G^{\omega}(M)$.

We give another definition of $R(G,M)$. Consider a sequence $(N_n,f_n)_{n\geqslant 0}$, where N_n is a hypergraph with vertices in $V.N^*$ and f_n is an injection of nonterminal hyperarcs of N_n into N^*. We shall set $R(G,M)=\bigcup_n[N_n]$. This sequence is defined as follows:

$$N_0=\{M\}\quad\text{and}\quad f_0(M)=\varepsilon,$$

$$N_{n+1}=[N_n]\cup\bigcup\{G_X\mid X\in N_n\wedge X(1)\in N\},$$

where for every nonterminal hyperarc X of N_n and for the rule (Y,K) of G such that $Y(1)=X(1)$, we have

$$G_X=\{fg_X(s_1)\dots g_X(s_n)\mid fs_1\dots s_n\in K\},$$

where for every vertex s in K, $g_X(s)$ is defined by

$g_X(s)=s.f_n(X)$ if s is not a vertex of a nonterminal hyperarc of K

$g_X(s)=U(i).U(1).f_n(X)$ if s is the ith vertex of a nonterminal hyperarc V of K and U is the left member hyperarc of the rule in G with the same label $U(1)=V(1)$ as V (such a V is unique because G is separated),

and for every nonterminal hyperarc Z of G_X,

$$f_{n+1}(Z(1)g_X(Z(2))\dots g_X(Z(|Z|)))=Z(1).f_n(X);$$

that is, we push (on the left side) the label of a nonterminal hyperarc before deriving it. So, each vertex in N_n is a word giving its place in its pattern (first letter) and the way it was obtained.

Figure 7 gives an example of representative. For every $n \geqslant 0$, we determine the restriction

$$G_n = \{ fst \in R(G,M) \mid d(s,M) < n \vee d(t,M) < n \}$$

of $R(G,M)$ to the vertices s whose distance $d(s,M)$ at M is at most $n-1$. Then $G_0 = \emptyset$ and as G is in Greibach form, $G_n \subseteq [N_n]$. The grammar H, to be constructed satisfying the proposition, must be able to generate from M in n steps of parallel rewritings, a graph having G_n as set of terminal arcs. With the exception of $M(1)$, a nonterminal of H will be a couple (P,Q), where P is a finite set of terminal arcs with vertices in $V.N^*$, and Q is a subset of vertices of P.

Let $n \geqslant 1$. We will determine a set M_n of nonterminal hyperarcs allowing the generation of the graph $R(G,M) - G_n$ according to H. To this aim, we determine the connected components $D_1, \dots, D_p$ of $N_{n+1} - G_n$. For $1 \leqslant i \leqslant p$, we consider the set

$$C_i = \{ fst \in D_i \mid f \notin N \wedge (s \in V_{G_n} \vee t \in V_{G_n}) \}$$

of terminal arcs of D_i having a vertex in G_n. The hypergraph M_n is defined by

$$\begin{aligned} M_n = \{ (C_i.(s_{C_i})^{-1}, (V_{C_i} \cap V_{G_n}).(s_{C_i})^{-1}).(u_{i,1}) \dots (u_{i,q_i}) \mid 1 \leqslant i \leqslant p \\ \wedge \{u_{i,1}, \dots, u_{i,q_i}\} = V_{C_i} \cap V_{G_n} \wedge \forall j,\ 1 \leqslant j < q_i,\ u_{i,j} < u_{i,j+1} \}. \end{aligned}$$

The grammar H we look for, is defined as the union of a sequence of grammars $(H_n)_{n \geqslant 1}$. This sequence is inductively constructed as follows:

$$H_1 = \{ (M, G_1 \cup M_1) \}$$

and

$$\begin{aligned} H_{n+1} = \{ (X.(s_C)^{-1}, C.(s_C)^{-1}) \mid X \in M_n \wedge X(1) \notin P_n \wedge C \text{ is the connected} \\ \text{component of } (G_{n+1} - G_n) \cup M_{n+1} \text{ having the vertices of } X \}, \end{aligned}$$

where P_n is the set of the nonterminals of $H_1, \dots, H_n$.

To prove the finiteness of H, it suffices to show that there exists only a finite number of possible nonterminals for H. Again, it suffices to find a bound on the distance in $R(G,M)$ of vertices common to C and G_n, for every n and every connected component C of $R(G,M) - G_n$.

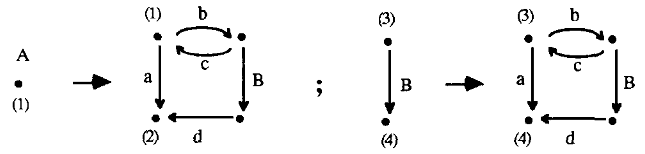
Because G is connected, the integer $b = \max\{ d_{R(G,Y)}(s,t) \mid \exists (Y,K) \in G,\ s,t \in V_K \}$ is well-defined. Let us consider a connected component C of $R(G,M) - G_n$ for any n, and vertices s and t common to C and G_n. We want to find an upper bound depending on b of the distance $d_{R(G,M)}(s,t)$. Let us take a vertex u of C with minimal length. As $R(G,M)$ is connected, there exists a path of minimal length $d_{R(G,M)}(s,M)$. The grammar G being in Greibach form, this path goes through the "pattern" of u, that is through a vertex v with the same suffix in N^* as u, i.e. $v(2)...v(|v|)=u(2)...u(|u|)=s_C$. Also $d_{R(G,M)}(u,v)\leqslant d_{R(G,Y)}(u(1),v(1))$ for the rule (Y,K) of G such that $u(1)$ is a vertex in K; hence, $d_{R(G,M)}(u,v)\leqslant b$. Since s is a "border" vertex of C and G_n, and u is a vertex of C, we have $d_{R(G,M)}(s,M)\leqslant d_{R(G,M)}(u,M)$; then

$$d_{R(G,M)}(s,v)+d_{R(G,M)}(v,M)=d_{R(G,M)}(s,M)\leqslant d_{R(G,M)}(u,M)$$
$$\leqslant d_{R(G,M)}(u,v)+d_{R(G,M)}(v,M).$$

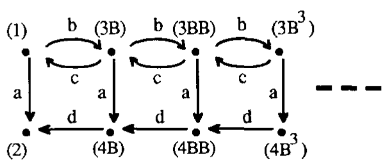
So, $d_{R(G,M)}(s,v)\leqslant d_{R(G,M)}(u,v)\leqslant b$; hence, $d_{R(G,M)}(u,s)\leqslant d_{R(G,M)}(u,v)+d_{R(G,M)}(v,s)\leqslant 2b$ and the same holds for $d_{R(G,M)}(u,t)$. Consequently, $d_{R(G,M)}(s,t)\leqslant 4b$, which is enough to prove the finiteness of H. So, there exists an m such that $H_{m+1}=\emptyset$. Then $H=\bigcup\{H_n \mid 1\leqslant n\leqslant m\}$ is effectively computable because the H_n are. By construction $M\in\mathrm{Dom}(H)$, H is uniform, and $R(G,M)$ is isomorphic to $H^\omega(M)$; therefore, $H^\omega(M)=G^\omega(M)$. □

The construction in the above proof is illustrated in Fig. 7.

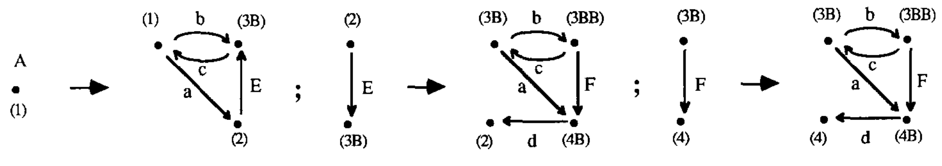
Let us consider the following (nonuniform) grammar G:



The representative $R(G,A1)$ of $G^\omega(A1)$ is the following graph:



The transformation in the proof of Proposition 2.17 gives the following uniform grammar H:



where

$$E=(\{3B\xrightarrow{b}3BB, 3BB\xrightarrow{c}3B, 3B\xrightarrow{a}4B, 4B\xrightarrow{d}2\},\{2,3B\})$$

$$F=(\{3B\xrightarrow{b}3BB, 3BB\xrightarrow{c}3B, 3B\xrightarrow{a}4B, 4B\xrightarrow{d}4\},\{4,3B\})$$

with the natural order on $\{1,2,3,4\}$ extended lexicographically on $\{1,2,3,4\}.B^*$.
Hence, $H^\omega(A1)=G^\omega(A1)$.

Fig. 7. Transformation of a grammar into a uniform grammar.

A noneffective version of Proposition 2.17 has been given by Bauderon [3]. After Propositions 2.17 and 2.7, we can decide that two accessible (connected) prefix transition graphs of word-rewriting systems are isomorphic with respect to some given vertices (i.e. the isomorphism is given on a pair of vertices, say on the roots).

Proposition 2.18. *From all triples (R, r, r') and (S, s, s') consisting of a rewriting system, a word and a vertex of the accessible (connected) prefix transition graph, we can decide whether there exists an isomorphism f from $P(R, r)$ to $P(S, s)$ (from $P(R)_r$ to $P(S)_s$) such that $f(r') = s'$.*

Proof. From Proposition 2.7 (Theorem 2.11) and Proposition 2.17, we transform (R, r, r') into a uniform grammar G and an axiom a_0r' such that the prefix transition graph of R accessible from r is the graph generated by G from a_0r', i.e. $P(R, r) = G^{\omega}(a_0r')$ $(P(R)_r = G^{\omega}(a_0r'))$. In the same way, we transform (S, s, s') into (H, b_0s'). Let us denote by N_G (N_H) the set of nonterminals in G (H). We will now compare the right members of the uniform grammar rules G and H, starting from the right members associated to a_0 and b_0: two such hypergraphs are comparable if there exists an isomorphism identifying their terminal arcs, and associating to every nonterminal hyperarc of the first one, a nonterminal of the other, up to a permutation of vertices. To compare the right members of G and H, consider the set E of words $e = (a, b)\pi(1)\ldots\pi(n)$; where a and b are nonterminals of arity n, from G and H, respectively, and π is a permutation of $\{1, \ldots, n\}$. With such a word e of E and given the rules $(as_1\ldots s_n \to P)$ in G and $(bt_1\ldots t_n \to Q)$ in H, we associate the finite set B_e of the bijections h of the vertices of P onto the vertices of Q, such that the following conditions hold:

$$h(s_i) = t_{\pi(i)} \quad \text{for } 1 \leqslant i \leqslant n,$$

$$cx_1\ldots x_m \in P \wedge c \notin N_G \Leftrightarrow ch(x_1)\ldots h(x_m) \in Q \wedge c \notin N_H,$$

$$cx_1\ldots x_m \in P \wedge c \in N_G \Rightarrow \exists dy_1\ldots y_m \in Q,\ d \in N_H$$
$$\wedge \{y_1, \ldots, y_m\} = \{h(x_1), \ldots, h(x_m)\},$$

$$dy_1\ldots y_m \in Q \wedge d \in N_H \Rightarrow \exists cx_1\ldots x_m \in P,\ c \in N_G$$
$$\wedge \{y_1, \ldots, y_m\} = \{h(x_1), \ldots, h(x_m)\};$$

we denote by $E_{e,h}$ the set of such words $(c, d)\sigma(1)\ldots\sigma(m)$ where $c \in N_G$ and $h(x_i) = y_{\sigma(i)}$ for $1 \leqslant i \leqslant m$.

Hence, there exists an isomorphism f of $P(R, r)$ onto $P(S, s)$ such that $f(r') = s'$ if and only if there exists a directed unlabelled graph C, with vertices in E, such that $(a_0, b_0)1$ is a vertex of C and if $e = (a, b)\pi(1)\ldots\pi(n)$ is a vertex of C then there exists a bijection h of B_e for which $E_{e,h}$ is the set of targets of arcs in C starting at e. Since the set C of such graphs is finite and constructible, we can decide on the isomorphism of $P(R, r)$ and $P(S, s)$ $(P(R)_r$ and $P(S)_s)$ associating r' with s'. □

Let us note that Proposition 2.18 is also a consequence of Proposition 2.7 and of Corollary 4.5 of [9].

Acknowledgment

Let me thank P. Darondeau, R. Monfort and J.-C. Raoult for their help in the drafting of this paper. I also thank some anonymous referees for their numerous remarks.

References

[1] J.C.M. Baeten, J.A. Bergstra and J.W. Klop, Decidability of bisimulation equivalence for processes generating context-free languages, in: Lecture Notes in Computer Science, Vol. 259 (Springer, Berlin, 1987) 94–111.

[2] M. Bauderon, On systems of equations defining infinite graphs, in: Lecture Notes in Computer Science, Vol. 344 (Springer, Berlin, 1989) 54–73.

[3] M. Bauderon, Infinite hypergraphs: basic properties and systems of equations, Internal Report I-8920, 1989.

[4] L. Boasson and M. Nivat, Centers of context-free languages, Internal Report LITP 84–44.

[5] R. Büchi, Regular canonical systems, *Arch. Math. Logik Grundlag.* **6** (1964) 91–111.

[6] D. Caucal, Récritures suffixes de mots, Report INRIA 871, 1988.

[7] D. Caucal and R. Monfort, On the transition graphs of automata and grammars, in: *WG 90*, Lecture Notes in Computer Science, Vol. 484 (Springer, Berlin, 1990) 311–337.

[8] B. Courcelle, The monadic second-order logic of graphs, II: Infinite graphs of bounded width, *Math. Systems Theory* **21** (1989) 187–222.

[9] B. Courcelle, The definability of equational graphs in monadic second order logic, in: *ICALP 89*, Lecture Notes in Computer Science, Vol. 372 (Springer, Berlin, 1989) 207–221.

[10] M. Dauchet and S. Tison, The theory of ground rewrite systems is decidable, in: *Proc. 5th IEEE Symp. LICS 90*, 242–248.

[11] C. Frougny and J. Sakarovitch, Rational relations with bounded delay, in: *Proc. 8th STACS 90*, Lecture Notes in Computer Science, Vol. 480 (Springer, Berlin, 1990) 50–63.

[12] A. Habel, Hyperedge replacement: grammars and languages, Dissertation, University of Bremen, 1989.

[13] A. Habel and H.J. Kreowski, Some structural aspects of hypergraph languages generated by hyperedge replacement, in: Lecture Notes in Computer Science, Vol. 247 (Springer, Berlin, 1987) 207–219.

[14] D. Muller and P. Schupp, The theory of ends, pushdown automata, and second order logic, *Theoret. Comput. Sci.* **37** (1985) 51–75.

[15] A. Salomaa, *Formal Languages*, ACM monograph series (Academic Press, New York, 1973).