On the regular structure of prefix rewritings

Didier Caucal

UNITE DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1196

Programme 1
Programmation, Calcul Symbolique et Intelligence Artificielle

ON THE REGULAR STRUCTURE OF PREFIX REWRITINGS

Didier CAUCAL

Mars 1990

RR . 1196

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

On the regular structure of prefix rewritings

Didier CAUCAL *

Publication interne n° 507 - Janvier 1990 - 32 pages

Abstract . We can consider a pushdown automaton as a word rewriting system with labelled rules applied only in a prefix way. The notion of context-free graph, defined by Muller and Schupp is then extended to the notion of prefix transition graph of a word rewriting system. Prefix transition graphs are context-free graphs, and we show 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 characterisation is effective in the following sense : any finite family of patterns generating a graph G having a finite degree and a root, is mapped effectively into a rewriting system R on words such that the prefix transition graph of R is isomorphic to G , and the reverse transformation is effective.

Sur la structure régulière des récritures préfixes

Résumé . On peut considérer un automate à pile comme un système de récritures de mots, à règles étiquetées, et où les transitions sont préfixes. La notion de graphe d'automate, définie par Muller et Schupp, est alors étendue en celle de graphe de transition préfixe de systèmes de récritures de mots. Les graphes de transition préfixe sont les graphes d'automates, et sont aussi les graphes ayant une racine, dont chaque sommet est de degré fini, et à motifs, c'est-à-dire produit à partir d'un graphe fini en itérant l'adjonction parallèle et déterministe d'une famille finie de graphes finis (les motifs). Cette correspondance est effective dans le sens où, à toute famille finie de motifs engendrant un graphe G ayant une racine et dont chaque sommet est de degré fini, on associe de façon effective un système de récriture de mots dont le graphe de transition préfixe est isomorphe à G . De même, le passage inverse est effectif.

Ce travail a été effectué à l'IRISA de Rennes. Programme 1 (Projet NICAS)

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (L. A. 227)
UNIVERSITÉ DE RENNES 1 — I. N. S. A. DE RENNES
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (LABORATOIRE DE RENNES)

On the regular structure of prefix rewritings (1)

Didier CAUCAL

IRISA , Campus de Beaulieu , F35042 Rennes-Cedex , France

Introduction

A rewriting system on a set Σ of terminal symbols and a set X of non-terminal symbols, is a finite set of rules $u \xrightarrow{f} v$ between words $u , v \in X^*$, labelled by some word $f \in \Sigma^*$. One step of prefix rewriting in a rewriting system R is a labelled transition $uw \xrightarrow{f} vw$ between words in X^* , where $u \xrightarrow{f} v$ is a rule of R. Prefix rewriting steps may be viewed as the arcs of a graph ; a prefix transition graph is the graph generated in this way from any axiom in X^* .

As an example of prefix transitions, let us briefly introduce the transitions between the configurations of a pushdown automaton, pda for short. 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 contents $A_1...A_n$. Then the transition relation of the pda may be seen as a rewriting system ; any transition between configurations is mapped in this way to a step of prefix rewriting. The corresponding prefix transition graph is called a pushdown transition graph. We show (in section 1) that every prefix transition graph is isomorphic to a pushdown transition graph. Secondly, we will show (also in section 1) that the correspondence may be lifted to the level of languages over Σ : the context-free languages are recognized by the prefix transition graphs working like (infinite) automata, with finite, or context-free, set of final words over X .

(1) This work will be presented at CAAP 90 .

A well-known property of accessible configurations of pda's is that they form regular languages. A similar property holds in the case of prefix rewriting [Bü 64] : the set of words in X^* reachable from a given axiom is again a regular language. We get in fact a stronger result. Consider the binary relation over X^* induced from prefix derivations by forgetting labels in Σ^* . This relation is in fact a rational transduction. We give (in section 2) a procedure which, given a rewriting system , produces the corresponding transducer.

In a seminal paper, Muller and Schupp have proved that every pushdown transition graph has a regular structure, which can be generated via a deterministic graph grammar. We will give an effective proof of this result, by writing a procedure which produces the graph grammar (section 3). Conversely, we will also show that any rooted graph with finite degree, generated by a deterministic graph grammar, is isomorphic to a prefix transition graph, and we moreover give a procedure which produces the corresponding rewriting system (also in section 3). At last, we establish effectively the characterisation of Muller and Schupp [Mu-Sc 85] . As a corollary, we can decide that two prefix transition graphs are isomorphic with respect to some given vertices.

The proofs are given in appendix.

1. Pushdown automata and prefix rewriting

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 for short, and their transitions. The transitions of a pda are a particular case of prefix rewritings. They are equally powerful : the transition graphs are the same, and the same holds for their trace languages, that is to say the language of labels along the paths from an axiom to a vertex in a given finite set (which are respectively the context-free graphs and the context-free languages).

Let us first introduce notations and terminology for rewriting systems. In the sequel, X and Σ are fixed alphabets, of non-terminals, and terminals respectively. A *rewriting system* R on (X,Σ) is a finite set of rules of the form $u \xrightarrow{f} v$, where $f \in \Sigma^*$ and $u, v \in X^*$. A rewriting system is said to be *alphabetic* if $u \in X$ for any rule $u \xrightarrow{f} v$, and *ε-free* if both u and v are non empty in any such rule.

Rewritings in a rewriting system are generally defined as applications of rewriting rules in any context. On the contrary, we are exclusively concerned in this paper with *prefix rewriting* defined as follows : given a rewriting system R, a *prefix rewriting step* (according to R) is a labelled transition $uw \overset{f}{\longmapsto} vw$ where $u \xrightarrow{f} v$ is a rule in R and $w \in X^*$. Let $u \longmapsto v$, resp. $u \overset{*}{\longmapsto} v$, represent an elementary (unlabelled) prefix rewriting step, resp. an arbitrary sequence of such steps. For example, let us consider the unlabelled rewriting system $S = \{ abb \to ab , abc \to ca , c \to cb \}$ on $X = \{a,b,c\}$; the language $\{ u \mid ab^nc \overset{*}{\longmapsto} u \}$ of words obtained by prefix rewriting from ab^nc is equal to $\{ ab^ic \mid 1 \leq i \leq n \} \cup cb^*a$.

Prefix rewriting may be seen as a way to generate terminal languages. Given a rewriting system R on (X,Σ), an axiom $r \in X^*$, and a set F of final states in X^*, the language L(R,r,F) recognized by the sequential machine (R,F) starting at r, is the set of labels $f_1 \ldots f_n$ of paths $w_1 \overset{f_1}{\longmapsto} w_2 \ldots \overset{f_n}{\longmapsto} w_{n+1}$ such that $w_1 = r$ and $w_{n+1} \in F$. In the case where F is a finite [resp. context-free] subset of X^*, L(R,r,F) is said to be *finitely accepted* [resp. *context-free accepted*].

The following proposition about context-free accepted languages is easily proven on the basis of the forthcoming lemmas 1.3 and 1.4, and from theorem 5.5 of [Sa 79].

Proposition 1.1 . *Finitely accepted languages (respectively context-free accepted languages) coincide with context-free languages.*

Prefix rewriting may also be seen as a way to generate labelled transition graphs. The prefix transition graph G(R,r) generated from an axiom $r \in X^*$ according to R is the set of arcs $w \overset{f}{\longmapsto} w'$ induced by the corresponding prefix rewriting steps from words w such that $r \overset{*}{\longmapsto} w$. Figure A gives an example of a prefix transition graph.

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 prefix transition graph G(R,p) is represented by

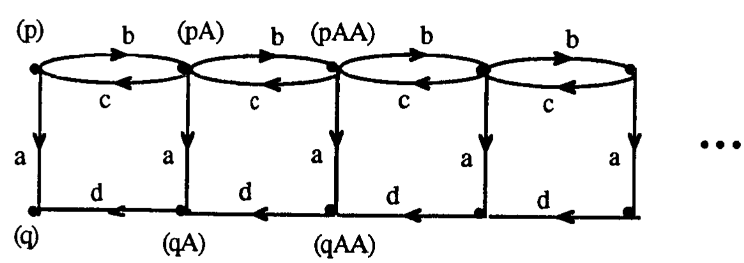


Fig. A . Prefix transition graph .

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

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

(i) X is partitioned into $Q_R \cup P_R$

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

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

Definition. A *prefix transition graph* (resp. a *pushdown transition graph*, an *alphabetic graph*) is a graph isomorphic to G(R,r) for some rewriting system R (resp. 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 arcs are preserved. The main result of the section is the following.

Theorem 1.2 . *Prefix transition graphs coincide with pushdown transition graphs.*

The problem in establishing theorem 1.2 lies in the transformation of a prefix rewriting system into a pda without introducing ε transition, nor duplication in the prefix transition graph. The proof of the theorem 1.2 will be cut into two lemmas. The first lemma (lemma 1.3) shows that any prefix transition graph is generated by a normal (see next definition) ε-free transition system. The second lemma (lemma 1.4) shows that normal ε-free transition systems are equivalent to pushdown automata as far as generated graphs are regarded.

Definition. A rewriting system R is *normal* if both u and v have length (strictly) smaller than 3 for any rule $u \xrightarrow{f} v$ in R .

Lemma 1.3 . *Any pair* (R,r) *consisting of a rewriting system* R *on* (X,Σ) *and an axiom* $r \in X^*$ *normalizes effectively to another pair* (S,s) *, where* S *is a normal ε-free rewriting system on* (Y,Σ) *and* $s \in Y$ *, such that* G(S,s) *is isomorphic to* G(R,r) .

Such a transformation is not usual. For instance and by identifying AA with B, the alphabetic system $R = \{ A \xrightarrow{a} \varepsilon , A \xrightarrow{b} A^3 \}$ can be transformed into the normal alphabetic system $S = \{ A \xrightarrow{a} \varepsilon , A \xrightarrow{b} BA , B \xrightarrow{a} A , B \xrightarrow{b} BB \}$ recognizing the same language, on empty stack from A; nevertheless G(R,A) is not isomorphic to G(S,A).

Lemma 1.4 . *Any pair* (R,r) *consisting of a normal* ε*-free rewriting system* R *on* (X,Σ) *and an axiom* $r \in X^*$ *may be effectively transformed into a pair* (S,s), *where* S *is a normal pushdown automaton on* (Y,Σ) *and* $s \in Q_S$, *such that* G(S,s) *is isomorphic to* G(R,r) .

After proposition 1.1 and theorem 1.2, we may ask whether alphabetic rewriting systems, which have the same trace languages as the pda's, are also representatives of arbitrary rewriting systems as far as generated graphs are concerned. The next proposition answers this question negatively.

Proposition 1.5 . *The class of alphabetic graphs is a proper subset of the class of pushdown transition graphs.*

For instance, the prefix transition graph of figure A is not alphabetic. Nevertheless, in the restricted case where G(R,r) has at least one co-root state (reachable from every other state), we have the following result.

Theorem 1.6 . *From any pair* (R,r) *consisting of a rewriting system* R *and an axiom* r *such that its transition graph* G(R,r) *has a co-root, we can decide whether* G(R,r) *is an alphabetic graph, and in this case, the pair* (R,r) *may be effectively transformed into a pair* (S,s) *where* S *is a alphabetic rewriting system and* s *is a letter, such that* G(S,s) *is isomorphic to* G(R,r) .

The construction, got with R. Monfort, needs the forthcoming theorems 3.2 and 3.3 .

2. Prefix rewriting and rational transduction

In this section, we discard labels from transitions, and focus on the prefix rewriting relation $\overset{*}{\underset{R}{\longmapsto}}$ defined in section 1 . Recall that $\underset{R}{\longmapsto}$ is the componentwise concatenation $R.\Delta$ where Δ is the identity on the set of words. We show that for any R , the relation $\overset{*}{\underset{R}{\longmapsto}}$ generated by R is a rational transduction. A bunch of known results about prefix rewriting follows immediately. For instance,

a) the set of words originating infinite derivations along $\underset{R}{\longmapsto}$ is regular [Bo-Ni 84] ,

b) the set of accessible configurations (in $Q_R.P_R^*$) of a pushdown automaton R is rational [Bü 64] , [Au 87] (problem 14),

c) the equivalence generated by $\underset{R}{\longmapsto}$ is decidable [Ne-Op 80] ,

d) confluence and termination properties of $\underset{R}{\longmapsto}$ are decidable [Da-et al. 87] and [Hu-La 78] .

Henceforth, R is a finite subset of X^*xX^* , and uw $\underset{R}{\longmapsto}$ vw holds if u R v and $w \in X^*$. Let us state the main result of the section.

Theorem 2.1 . *For any* R , $\overset{*}{\underset{R}{\longmapsto}}$ *is a rational transduction, and a corresponding transducer is effectively constructible from* R .

The proof of this theorem is cut in two steps. We first construct from R a finite automaton A(R) (i.e. a pushdown automaton A(R) with empty stack alphabet $P_{A(R)} = \emptyset$) such that $L(A(R \cup \{r \to r\}), \varepsilon, \{r\})$ (see the beginning of section 1) coincides, for any $r \in X^*$, with the language generated from r according to the derivation relation $\overset{*}{\underset{R}{\longmapsto}}$ (i.e. with $\{ w \mid r \overset{*}{\underset{R}{\longmapsto}} w \}$) . The complexity of this construction is polynomial in time and space, instead of exponential as in Büchi [Bü 64] (lemma 3 and theorem 1). We then combine A(R) and the companion automaton $A(R^{-1})$ into a finite state machine.

Let us proceed to the construction of A(R) . For u in X^* , let left(u) denote the longest left factor of u in the domain of R , or ε by default, and let u = left(u).right(u) . The principle of the construction is to set a producer-consumer relation between the prefix rewriting system R and the automaton A(R) : segments laid down from right to left by prefix rewriting are taken from left to right by the automaton. Let us first 'revert' the rules of R. We set

$$H = \{ \text{left}(v) \xrightarrow{f} u \mid f = \text{right}(v) \text{ and } u R v \}$$

and $$I = \{ \text{left}(u) \xrightarrow{f} ua \mid f = \text{right}(u).a \text{ and } a \in X \text{ and } ua R v \}.$$

$H \cup I$ is not yet the desired automaton, because a 'left member' may be laid down in several slices, produced in successive steps in prefix rewriting. Thus, the transition of the automaton should be cut into corresponding slices. So, we set $A = \langle H \cup I \rangle$, where for any finite set G of labelled transitions $s \xrightarrow{f} t$, the sliced version $\langle G \rangle$ of G is its closure with respect to the following property :

if $s \xrightarrow{u'u''} t \in G$ and $v \in L(G,\varepsilon,\{s\})$ with $u' \neq \varepsilon$

then $vu' \xrightarrow{u''} t \in G$ if $vu' = \text{left}(vu'u'')$ and $u'' \notin L(G,vu',\{t\})$.

The above construction has been proved correct in [Ca 88] and is illustrated in figure B.

Let $R = \{\varepsilon \to ba, bab \to ba\}$. Then

$H = \{ \varepsilon \xrightarrow{ba} \varepsilon, \varepsilon \xrightarrow{ba} bab \}$

$I = \{ \varepsilon \xrightarrow{bab} bab \}$

The automaton $A(R) = \langle H \cup I \rangle$ is represented by

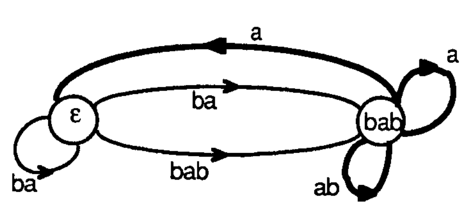


Fig. B . Automaton of a prefix rewriting.

As an immediate corollary, we get a decision algorithm for the termination of $\underset{R}{\longmapsto}$: $\underset{R}{\longmapsto}$ has a infinite derivation from r if and only if the connected component of $A(R \cup \{r \to r\})$ containing r has a cycle.

We are ready to prove theorem 2.1 . Let T be the finite union of product languages

$$L(A(R^{-1}),\varepsilon,\{u\}) \times L(A(R),\varepsilon,\{u\}) \quad \text{for } u \in Dom(R) \cup \{\varepsilon\} .$$

Clearly, T is a recognizable subset of X^*xX^* by Mezei's theorem [Be 79] , and a transducer of $\underset{T}{\longmapsto}$ is effectively constructible from R . Then $\underset{T}{\longmapsto}$ is the required transducer, as shown by the following lemma.

Lemma 2.2 . $u \overset{*}{\underset{R}{\longmapsto}} v$ *if and only if for some* $w \in X^*$ *and* $z \in Dom(R) \cup \{\varepsilon\}$

$$u = xw \ , \ v = yw \ \textit{and} \ x \overset{*}{\underset{R}{\longmapsto}} z \overset{*}{\underset{R}{\longmapsto}} y .$$

In view of theorem 2.1 , the derivation relation $\overset{*}{\underset{R}{\longmapsto}}$ generated by a finite relation R is a rational transduction. Furthermore, we have seen in the proof that $\overset{*}{\underset{R}{\longmapsto}} = \underset{T}{\longmapsto}$ for some recognizable relation T . We are indebted to J.M. Autebert for a positive answer to the question : when R is recognizable, does $\overset{*}{\underset{R}{\longmapsto}}$ still coincides with $\underset{T}{\longmapsto}$ for some recognizable T ?
This extended result allows us to state that one step prefix rewriting relations, according to recognizable relations, form a 'rational' family in the following sense.

Proposition 2.3 . *The family of relations* $\{ \underset{R}{\longmapsto} \mid$ R is a recognizable relation $\}$ *is closed under union, composition, and starred composition (of relations).*

This is in fact a generalization of theorem 2.1 , for the proof given in appendix is effective. Let us remark that the family of the proposition 2.3 is not closed under intersection, nor under complementation (by de Morgan law), because we have the following equality :

$$(c^* \times c^*).\Delta \cap \Delta = (\{c\}\times\{c\})^*.\Delta .$$

Let us remark that $(\{a\}\times\{\varepsilon\}).\Delta$ is not a prefix rewriting relation $\overset{*}{\underset{R}{\longmapsto}}$ over $\{a,b\}^*$.
Furthermore, we get for free a solution of the decision problem for the confluence of $\underset{R}{\longmapsto}$ for recognizable R .

Proposition 2.4 . *The confluence of* $\underset{R}{\longmapsto}$ *is decidable for recognizable* R .

3. Prefix rewriting and pattern graph

Since, for any finite relation R on X^*, the prefix rewriting relation $\underset{R}{\overset{*}{\longmapsto}}$ generated by R is a rational transduction, prefix rewriting has a regular behaviour. In particular, the set of words of any prefix transition graph is a regular language (over X^*). A natural question is then whether the regular structure of 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 theorem 1.1), and since Muller and Schupp [Mu-Sc 85] 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 (up to isomorphism) of connected components got after removing all vertices closer to a given vertex than a distance d , for any d . Thus, context-free graphs may be cut into slices of a finite number of 'patterns'.

Building up over Muller and Schupp's ideas, we devise an effective construction of patterns for context-free graphs given by pda's. We also relax the constraint of splitting up the graph 'by slices' and allow to remove patterns of arbitrary shapes and sizes. This adds nothing to Muller and Schupp decomposition, but gives more leeway for the construction of patterns. Furthermore, we establish the converse result : we give a procedure which, given any finite system 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.

Definition. A *graph grammar* on a graded alphabet F and set of vertices V , is a finite set of hyperarc replacement rules $fv_1...v_n \rightarrow H$ where the word $fv_1...v_n$ is an *hyperarc* labelled by the *non-terminal* $f \in F_n$, the v_i are vertices and H is a finite *hypergraph*, that is a (multi-)set of hyperarcs, but where the v_i are distinct vertices. Every *terminal* of the grammar, that is to say every label of a right member rule hyperarc which is not a non-terminal, is of arity 2 .

A graph grammar is *deterministic* if two different rules cannot have the same non-terminal f .

Figure C is an example of a deterministic graph grammar.

Let A , a , b be in F of respective arity 3 , 2 , 2 .

Let G = { (A123 , {a12,a14,a25,a36,A564}) } be a deterministic graph grammar.

A is the unique non-terminal of G , and G is represented as follows :

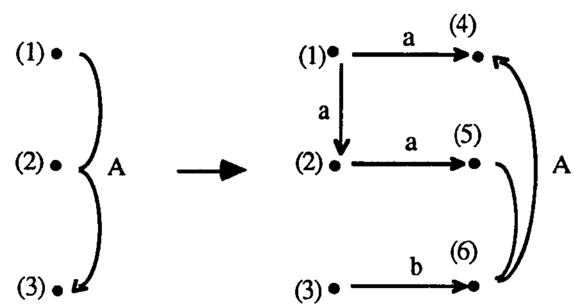


Fig. C . Deterministic graph grammar.

Each deterministic graph grammar defines a graph, resulting from a given axiom graph by iterating the graph rewriting [Ha-Kr 87] . We use the symbol + for the addition of multi-sets.

Definition. Given a graph grammar G on (F,V) and an hypergraph M on (F,V) , M *rewrites in one step* to a hypergraph N , and we note $M \rightarrow_G N$, if for some rule $fs_1 \ldots s_n \rightarrow H$, there exists a hypergraph M' such that $M = M' + \{ft_1 \ldots t_n\}$ and $N = M' + \{ hg(x_1) \ldots g(x_m) \mid hx_1 \ldots x_m \in H \}$ for some matching function g mapping s_i to t_i , and mapping injectively the other vertices of H to vertices outside of M .

Beware that $\rightarrow_G$ is not in general a functional relation, even though G is deterministic. Nevertheless, if we let $M \rightarrow_{G,X} N$ denote the rewriting of a non-terminal hyperarc X , then

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

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

$$M \Rightarrow_G N \quad \text{if} \quad M \rightarrow_{G,X_1} \circ \ldots \circ \rightarrow_{G,X_n} N \ ,$$

and M has exactly the n non-terminal hyperarcs $X_1, \ldots, X_n$. One step of complete parallel rewriting corresponds to the Kleene substitution. On that basis, we define $G^{\omega}(M)$, the set of hypergraphs generated from the axiom M according to the deterministic graph grammar G, as follows :

Definition. $G^{\omega}(M)$ is the set of hypergraphs N for which there exists an infinite sequence of hypergraphs $(N_n)_{n\geq 0}$, such that $N_0 = M$ and for all n, $N_n \Rightarrow_G N_{n+1}$, with limit

$$N = \{ fs_1...s_n \mid \exists\, i,\ fs_1...s_n \in N_i \ \text{ and } \ f \text{ is a terminal} \} .$$

Since G is deterministic, $G^{\omega}(M)$ has a single element 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 and Courcelle [Ba 89 a], [Co 89 a]. The grammar of the figure C generates from A123 the pattern graph of the following figure D.

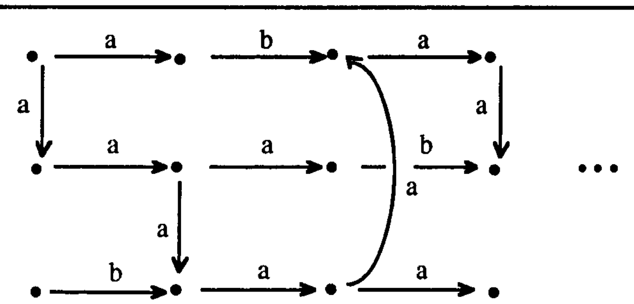


Fig. D . Pattern graph .

Let us recall that a graph G is of *finite degree* [resp. of *bounded degree*] if [resp. there exists a bound b such that] for every vertex s in G, the number of arcs to which s belongs is finite [resp. is smaller than b]. Let us point 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 is reachable from r . In particular, every prefix transition graph G(R,r) has a finite degree and a root r .

Our goal is to establish constructively the following statement.

Theorem 3.1 . *Prefix transition graphs coincide exactly with rooted pattern graphs of finite degree.*

This theorem may be equivalently restated in two others theorems, one for each the inclusions.

Theorem 3.2 . *Any pair* (R,r) *of a word rewriting system* R *on* (X,Σ) *and a axiom* $r \in X^*$ *, may be effectively transformed into a pair* (G,M) *of a deterministic graph grammar* G *and a hyperarc* M *, such that the corresponding graphs* G(R,r) *and* $G^\omega(M)$ *are isomorphic.*

A restricted version of theorem 3.2 was established in [Ba-Be-Kl 87] for grammatical graphs with a co-root with out-degree zero.

Theorem 3.3 . *Any pair* (G,M) *of a deterministic graph grammar* G *and of an axiom* $M = fs_1 \ldots s_n$ *, such that* $G^\omega(M)$ *has finite degree and has root* s_1 *, may be effectively transformed into a pair* (R,r) *, of a word rewriting system* R *and an axiom* r *, such that the corresponding graphs* $G^\omega(M)$ *and* G(R,r) *are isomorphic.*

After theorem 3.2 and theorem 3.3 , we can determine a word rewriting system of the inverse of any prefix transition graph with a co-root.

Proposition 3.4 . *Any triple* (R,r,c) *consisting of a rewriting system* R *, an axiom* r *and a co-root* c *of* G(R,r) *, may be effectively transformed into an another triple* (S,s,d) *such that there exists an isomorphism* f *from* G(S,s) *to the inverse of* G(R,r) *satisfying* f(s) = c *and* f(d) = r .

Theorems 3.2 and 3.3 allow the study of other effective transformations of prefix rewriting systems, and not only computing the inverse as in proposition 2.4 .

We shall now establish effectively the characterisation of Muller and Schupp [Mu-Sc 85] . The next definition translates their notion of finite decomposition into the framework of generating grammars.

Definition. A *uniform* grammar is a deterministic graph grammar when all rules $fv_1...v_n \to H$ of G satisfy the following conditions :

(1) all vertices of all non-terminal hyperarcs of H are distinct,

(2) every vertex of a non-terminal hyperarc of H also belongs to a terminal arc of H , and is different of the v_i ,

(3) every terminal arc of H goes through at least one v_i ,

(4) $G^{\omega}(fv_1...v_n)$ is connected.

For instance, the grammar of figure C is uniform. It is obvious to see that a *context-free* graph is a graph with a co-root which can be generated by a uniform grammar. So, a context-free graph is a finite degree pattern graph. Our goal is to establish constructively the following characterisation of Muller and Schupp [Mu-Sc 85] .

Theorem 3.5 . *Context-free graphs coincide exactly with pushdown transition graphs.*

From theorems 3.3 and 1.2 , every context-free graph is effectively a pushdown transition graph. The converse follows from theorem 3.2 and from the theorem below.

theorem 3.6 . *Any pair* (G,M) *of a deterministic graph grammar* G *and of an axiom* M *such that* $G^{\omega}(M)$ *is a finite degree connected graph, may be effectively transformed into a uniform grammar* H *such that* $H^{\omega}(M)$ *is equal to* $G^{\omega}(M)$.

A non-effective version of theorem 3.6 has been given by Bauderon [Ba 89 b] . After theorem 3.6

and theorem 3.2 , we can decide that two 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 3.7 . *From all triples* (R,r,r') *and* (S,s,s') *consisting of a rewriting system, an axiom and a vertex of the generated prefix transition graph, we can decide whether there exists an isomorphism* f *from* G(R,r) *to* G(S,s) *such that* f(r') = s' .

Let us point out that proposition 3.7 is also a consequence of theorem 3.2 and of corollary 4.5 of [Co 89 b] .

Aknowledgments

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

References

Au 87 J.-M. Autebert 'Langages algébriques' , Ed. Masson .

Ba-Be Kl 87 J.C.M. Baeten, J.A. Bergstra, J.W. Klop 'Decidability of bisimulation equivalence for processes generating context-free languages' , LNCS 259 , pp 94-111 .

Ba 89 a M. Bauderon , 'On systems of equations defining infinite graphs' , LNCS 344 , pp 54-73 .

Ba 89 b M. Bauderon , 'Infinite hypergraphs : basic properties and systems of equations' , Internal report I-8920 .

Be 79 J. Berstel 'Transductions and context-free languages' , Ed. Teubner .

Bo-Ni 84 L. Boasson , M. Nivat 'Centers of context-free languages' , Report LITP 84-44 .

Bü 64 R. Büchi 'Regular canonical systems' , Archiv für Mathematische Logik und Grundlagenforschung 6 , pp 91-111.

Ca 88 D. Caucal 'Récritures suffixes de mots' , Report Inria 871 .

Ca 90 D. Caucal 'On the regular structure of prefix rewriting' , Internal report 507 .

Co 89 a B. Courcelle 'The monadic second-order logic of graphs, II : Infinite graphs of bounded width' , Math. Syst. Theory 21 , pp 187-222.

Co 89 b — B. Courcelle 'The definability of equational graphs in monadic second-order logic', Icalp, LNCS 372, pp 207-221.

Da-Ti He-Le 87 — M. Dauchet, S. Tison, T. Heuillard, P. Lescanne 'The confluence of ground term rewriting systems is decidable' Preliminary proceeding of the conference on resolution of equations in algebraic structures. Austin.

Ha-Kr 87 — A. Habel, H.J. Kreowski 'Some structural aspects of hypergraph languages generated by hyperedge replacement', LNCS 247, pp 207-219.

Hu-La 78 — G. Huet, D. Lankford 'On the uniform halting problem for term rewriting systems' Report INRIA 283.

Mu-Sc 85 — D. Muller, P. Schupp 'The theory of ends, pushdown automata, and second order logic', TCS 37, pp 51-75.

Ne-Op 80 — G. Nelson, D. Oppen 'Fast decision procedures based on congruence closure' JACM 27 - 2, pp 356-364.

Sa 79 — J. Sakarovitch 'Syntaxe des langages de Chomsky. Essai sur le déterminisme', Thèse d'état, LITP.

Appendix

Lemma 1.3 . *Any pair* (R,r) *consisting of a rewriting system* R *on* (X,Σ) *and an axiom* $r \in X^*$ *normalizes effectively to another pair* (S,s) *, where* S *is a normal* ε*-free rewriting system on* (Y,Σ) *and* $s \in Y$ *, such that* G(S,s) *is isomorphic to* G(R,r) .

Proof.

We may suppose R ε-free and $r \neq \varepsilon$. Otherwise, we could take a letter a in X appearing neither in R nor r , and replace (R,r) by (aR,ar) with $aR = \{ (au,f,av) \mid (u,f,v) \in R \}$; so aR is ε-free, $ar \neq \varepsilon$ and G(aR,ar) = aG(R,r) .

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

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

Let us extend an injection i from $\{ u \in X^* \mid 1 \leq |u| \leq m \}$ to some given alphabet Y to an injection j : $X^* \to Y^*$ so that $j(\varepsilon) = \varepsilon$ and for every word $u \neq \varepsilon$, j is defined recursively by

$$j(u) = j(v)i(w) \text{ with } vw = u \text{ and } |w| = \min(m,|u|) .$$

The rewriting system S over (Y,Σ) defined as

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

is normal and ε-free. Moreover s = j(r) is a letter and we show that

$$G(S,s) = \{ (j(u),f,j(v)) \mid (u,f,v) \in G(R,r) \} .$$

♦

Lemma 1.4 . *Any pair* (R,r) *consisting of a normal* ε*-free rewriting system* R *on* (X,Σ) *and an axiom* $r \in X^*$ *may be effectively transformed into a pair* (S,s) *, where* S *is a normal pushdown automaton on* (Y,Σ) *and* $s \in Q_S$ *, such that* G(S,s) *is isomorphic to* G(R,r) .

Proof.

Given the following alphabets :

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

and $$\Gamma = \{ u(i) \mid 2 \leq i \leq |u| \wedge \exists f \, \exists v \; ((u,f,v) \in R \vee (v,f,u) \in R) \} ,$$

where u(i) is the i^{th} letter of u .

With 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 :

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

The rewriting system S on $(P \cup Q, \Sigma)$ defined by

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

is a normal and pushdown automaton with $Q_S = Q$ and $P_S = P$. Furthermore, we show that

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

with $s = i(r) = r \in Q_S$. ♦

Proposition 1.1 . *Finitely accepted languages (respectively context-free accepted languages) coincide with context-free languages.*

Proof.

Let X and Σ be alphabets, R a rewriting system over (X,Σ) , r a word in X^* and A a context-free part of X^* . To lay down proposition 1.1 , it is sufficient to established that L(R,r,A) is context-free. From lemma 1.3 and lemma 1.4 , we may assume R is a pushdown automaton and r is a state of R . For any state q , the language $L_q = q(q^{-1}.A)$ is context-free, and from theorem 5 p. 110 of Sakarovitch [Sa 79] , L(R,r,A) is context-free. ♦

Proposition 1.5 . *The class of alphabetic graphs is a proper subset of the class of pushdown transition graphs.*

Proof.

Any alphabetic graph is a prefix rewriting one and after theorem 1.2 , algebraic as well. Let us show that the algebraic graph of figure A is not alphabetic. Let us take R = { (p,a,q) , (p,b,pA) , (pA,c,p) , (qA,d,q) } and suppose there exist an alphabet X , an alphabetic system S on (X,{a,b,c,d}) and a word s in X^* so that G(R,p) is isomorphic to G(S,s) according to a bijection f . Then there is an integer m so that

$$|f(pA^{m+1})| > |f(pA^m)| \geq |f(q)| .$$

Let us put $v = f(pA^m)$ and $w = f(pA^{m+1})$. As S is alphabetic, $|w| > |v|$ and $(w,c,v) \in G(S,s)$, there is $B \in X$ with $w = Bv$. S being alphabetic and $(w,a,f(qA^{m+1})) \in G(S,s)$, there exists $x \in X^*$ such that $f(qA^{m+1}) = xv$. As there exists an unique path in $G(S,s)$ from $f(qA^{m+1})$ to $f(q)$, and $|v| \geq |f(q)|$ in the alphabetic S, there is an integer n, $0 \leq n \leq m+1$ and $f(qA^n) = v$. So $f(pA^m) = f(qA^n)$, then $pA^m = qA^n$, so $p = q$ which is a contradiction. ♦

Lemma 2.2 . $u \overset{*}{\underset{R}{\longmapsto}} v$ *if and only if for some* $w \in X^*$ *and* $z \in Dom(R) \cup \{\varepsilon\}$

$$u = xw \ , \ v = yw \ \textit{and} \ x \overset{*}{\underset{R}{\longmapsto}} z \overset{*}{\underset{R}{\longmapsto}} y .$$

Proof.

The if part is immediate. The only part is established by induction on the length of the derivation. ♦

Proposition 2.3 . *The family of relations* $\{ \underset{R}{\longmapsto} \mid R$ is a recognizable relation $\}$ *is closed under union, composition, and starred composition (of relations).*

Proof.

The closure of the family $\mathcal{F} = \{ R.\Delta \mid R \text{ recognizable} \}$ by union follows from the distributivity of the concatenation over the union. Moreover, $\mathcal{F}$ is closed by composition because

$$(AxB).\Delta \circ (CxD).\Delta = (A(B^{-1}C)xD \cup AxD(C^{-1}B)).\Delta ,$$

where $B^{-1}C = \{ u \mid \exists v \in B , vu \in C \}$ is the left quotient of C by B.

Finally, the proof of the closure of $\mathcal{F}$ for the starring of composition is given by J.-M. Autebert. Let us consider a relation $R = \bigcup\{ A_ixB_i \mid 1 \leq i \leq p \}$ recognizable, which means that for every i between 1 and p, the languages A_i and B_i are rationals on X^*. Taking an alphabet $\underline{X}$ of 'underlined' letters in bijection with X. To any word $u = a_1 \ldots a_n$ with $a_i \in X$, there corresponds the word $\underline{u} = \underline{a_n} \ldots \underline{a_1}$, and to any language L on X, the language $\underline{L} = \{ \underline{u} \mid u \in L \}$. The language

$$A = (\bigcup\{ B_i\underline{A_i} \mid 1 \leq i \leq p \})^*$$

is rational on $X \cup \underline{X}$. From Benois theorem [Be 79],

$$\rho(A) = \overset{*}{\underset{S}{\longrightarrow}}(A) \ \text{ with } S = \{ (\underline{a}a,1) \mid a \in X \}$$

is a rational language on $X \cup \underline{X}$. So its intersection with $X^*\underline{X}^*$ can be written in the following way

$$\rho(A) \cap X^*\underline{X}^* = \cup\{ C_i\underline{D}_i \mid 1 \leq i \leq q \} .$$

The relation between any language L and $\cup\{ C_i(D_i^{-1}L) \mid 1 \leq i \leq q \}$ is a rational transduction, belonging to $\mathcal{F}$ and corresponding to $(\Delta.R)^* = \overset{*}{\underset{R}{\longmapsto}}$. ♦

Proposition 2.4 . *The confluence of* $\underset{R}{\longmapsto}$ *is decidable for recognizable* R .

Proof.

To establish the confluence of $\underset{R}{\longmapsto}$ where R is recognizable, it is necessary and sufficient to decide of the following inclusion :

$$(R^{-1}.\Delta)^* \circ (R.\Delta)^* \subseteq (R.\Delta)^* \circ (R^{-1}.\Delta)^* .$$

From the proof of proposition 2.3 , it is sufficient to prove the decidability of the inclusion on $\mathcal{F}$. To this end, to every relation R on X^* is associated the set $\langle R \rangle$ of pairs of words obtained by cancelling the greatest common suffix from any pair of R , i.e.

$$\langle R \rangle = (R\Delta^{-1}) - ((X^*xX^*).\{(a,a) \mid a \in X\}) .$$

If R is recognizable, so is $\langle R \rangle$. As $R.\Delta = \langle R \rangle \Delta$ and $\langle R.\Delta \rangle = \langle R \rangle$, we have

$$R.\Delta \subseteq S.\Delta \text{ if and only if } \langle R \rangle \subseteq \langle S \rangle .$$

The inclusion being decidable on the set of recognizable relations on X^* , so is it on $\mathcal{F}$. ♦

Theorem 3.2 . *Any pair* (R,r) *of a word rewriting system* R *on* (X,Σ) *and a axiom* $r \in X^*$ *, may be effectively transformed into a pair* (G,M) *of a deterministic graph grammar* G *and a hyperarc* M *, such that the corresponding graphs* G(R,r) *and* $G^{\omega}(M)$ *are isomorphic.*

Proof.

From lemma 1.3 , we may take R normal and ε-free, and a letter r . The grammar G to be constructed generates G(R,r) by vertices of growing length.

Taking a vertex u of G(R,r) , we note $G(R,r)_u$ the connected component of G(R,r) restricted to the vertices of length $\geq |u|$, and containing u .

From theorem 2.1 , we can determine the set $V(u)$ of vertices of $G(R,r)_u$ of length $|u|$. Indeed if $R_0 = \{ (u,v) \mid \exists f,\ (u,f,v) \in R \}$ is the set of unlabelled rules of R , we can construct the automaton $A(R_0 \cup \{(r,r)\})$ recognizing the language $\underset{R_0}{\overset{*}{\longmapsto}} = \{ w \mid r \underset{R_0}{\overset{*}{\longmapsto}} w \}$ of vertices in $G(R,r)$. So, we can determine the finite set $D(u) = \underset{R_0}{\overset{*}{\longmapsto}} \cap X^{|u|}X^*$ of vertices in $G(R,r)$ of the same length as u . To determine $V(u)$, we construct the system S of unlabelled word rewritings on $X^{|u|}X^*$, defined by

$$S = \{ (xz,yz) \mid x\,R_0\,y \wedge |z| = \max(0,|u|-|x|) \} .$$

To decide if two elements in $D(u)$ are connected in the restriction of $G(R,r)$ to the vertices having length $\geq |u|$, we determine the following relation T on $D(u)$:

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

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

$$V(u) = T^*(u) = \{ v \mid u\,T^*\,v \} .$$

As R is normal, all vertices in $G(R,r)_u$ have a common suffix s_u of length $\max(0,|u|-2)$. Indeed $G(R,u)_u = G(R,V(u))_u$ where $G(R,E) = \bigcup\{G(R,e) \mid e \in E \}$ is the set of arcs in $G(R,e)$ with $e \in E$. Then for every proper subset P of $V(u)$, there exists two paths in $G(R,r)_u$, whose source are in P for the first one, and in $V(u) - P$ for the second one, i.e.

$$\forall P ,\ \emptyset \neq P \subset V(u) ,\ \exists x \in V(u) - P , \exists y \in P ,\ \underset{S}{\overset{*}{\longmapsto}}(x) \cap \underset{S}{\overset{*}{\longmapsto}}(y) \neq \emptyset .$$

So, all vertices in $V(u)$ have the same suffix s_u of length $\max(0,|u|-2)$, and the same holds for all vertices in $G(R,r)_u$.

Two vertices u and v of $G(R,r)$ are equivalent, noted $u \equiv v$, if $V(u).s_u^{-1} = V(v).s_v^{-1}$. If $u \equiv v$ then $G(R,r)_u$ is isomorphic to $G(R,r)_v$. 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 $G(R,r)_u$ with a vertex of length $|u|$. To construct the grammar G , we howe only to add to each H_u a set K_u of non-terminal hyperarcs which generates by G the graph $G(R,r)_u$ restricted to the vertices of length $> |u|$.

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

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

For any $u \in U$, we define

$$K_u = \{ f(s_1 s_w)\ldots(s_n s_w) \mid w \text{ is a vertex of } G(R,r)_u ,\ \exists v \in U ,\ |w| = |u|+1 ,\ w \equiv v$$

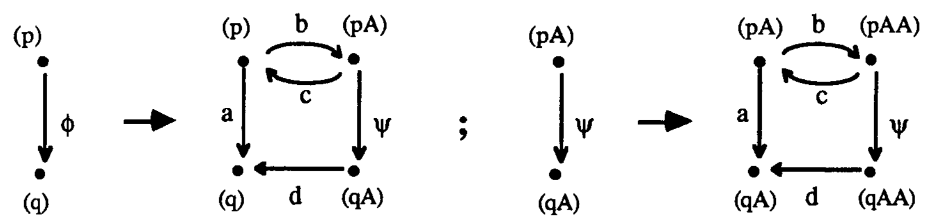
and $j(v) = f(s_1 s_v)\ldots(s_n s_v)$ } ,

and we define the deterministic graph grammar $G = \{ (j(u), H_u \cup K_u) \mid u \in U \}$.

For any $u \in U$, $G(R,r)_u$ is isomorphic to $G^{\omega}(j(u))$ and in particular for $M = j(r)$: $G(R,r)$ is isomorphic to $G^{\omega}(M)$. ♦

The construction of theorem 2.2 is illustrated in figure E .

Applied to the pair (R,p) of figure A , the construction of theorem 3.2 gives the following grammar G :



Then the prefix transition graph G(R,p) is isomorphic to the pattern graph $G^{\omega}(\phi pq)$ generated by G from ϕpq .

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

Theorem 3.3 . *Any pair* (G,M) *of a deterministic graph grammar* G *and of an axiom* $M = fs_1\ldots s_n$ *, such that* $G^{\omega}(M)$ *has finite degree and has root* s_1 *, may be effectively transformed into a pair* (R,r) *, of a word rewriting system* R *and an axiom* r *, such that the corresponding graphs* $G^{\omega}(M)$ *and* G(R,r) *are isomorphic.*

Proof.

Let us consider a graph grammar G on (F,V) and a hyperarc M in the domain dom(G) of G such that $G^{\omega}(M)$ has finite degree and has root M(2) .

By removing useless rules, we can suppose G to be *reduced,* which means that for every hyperarc X in Dom(G) , $G^{\omega}(X)$ is not empty and the non-terminal X(1) is accessible from M , i.e. $\exists N , \exists Y \in N$, $M (\rightarrow_G)^* N \wedge Y(1) = X(1)$. As $G^{\omega}(M)$ is of finite degree, the same holds for $G^{\omega}(X)$ with $X \in$ Dom(G) ; such a grammar G is said of *finite degree.* The removal of useless vertices from the hyperarcs in Dom(G) , yields a *proper* grammar : for such a grammar G , every vertex of an hyperarc X in Dom(G) is also a vertex of any hypergraph in $G^{\omega}(X)$. For instance, the grammar with the only rule $A12 \rightarrow \{a13,A32\}$ is not proper. As G is proper and of finite degree, rewritings by $(\Rightarrow_G)^*$ of the right members of the rules in G convert G in a *Greibach normal form*, that is to say for each rule (Y,H) of G, the vertices of every non-terminal hyperarc of H are disjoints from the Y's ones. After a possible renaming (and adding rules), we may suppose G with *separated outputs*, so that for any rule (Y,H) of G, two non-terminal hyperarcs of H have no common vertex, and every non-terminal hyperarc of H has distinct vertices. For easy writing and after a possible renaming, we may suppose that any right member hypergraph of a rule in G doesn't have two non-terminal hyperarcs with the same label. Finally and after a possible renaming of vertices, we may assume that the right member hypergraphs of the rules have no common vertex.

Let N be the set of non-terminals of G, and V_H be the set of vertices of a hypergraph H. To each rule (Y, H) of G, we associate a total function p_Y from V_H to $V \cup V.N$, which is the identity on the set of vertices in H which do not belong to non-terminal hyperarcs of H. For any vertex s of a non-terminal hyperarc Z of H, we have $p_Y(s) = T(i)T(1)$ where T is the non-terminal hyperarc in the domain of G with the same label as Z, and i is the place of s in Z, i.e.

$$p_Y(s) = s \qquad \text{for any } s \in V_H \text{ such that } s \notin V_J \text{ for any } J \in H \text{ and } J(1) \in N$$

$$p_Y(s) = T(i)Z(1) \quad \text{if there exist } Z \in H \text{ and } T \in dom(G) \text{ such that } Z(i) = s \text{ and } T(1) = Z(1).$$

As G has separated outputs, p_Y is well defined.

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

$$R(G) = \{ (p_Y(s).Y(1),a,p_Y(t).Y(1)) \mid \exists H, (Y,H) \in G \text{ and } ast \in H \text{ and } a \notin N \}.$$

We show that $G(R(G),M(2).M(1))$ is isomorphic to $G^{\omega}(M)$. ♦

Applied to the grammar G of figure E, the construction of theorem 3.3 gives the following rewriting system S :

$$S = \{ p\phi \xrightarrow{a} q\phi,\ p\phi \xrightarrow{b} [pA]\psi\phi,\ [pA]\psi\phi \xrightarrow{c} p\phi,\ [qA]\psi\phi \xrightarrow{d} q\phi,$$
$$[pA]\psi \xrightarrow{a} [qA]\psi,\ [pA]\psi \xrightarrow{b} [pA]\psi\psi,\ [pA]\psi\psi \xrightarrow{c} [pA]\psi,\ [qA]\psi\psi \xrightarrow{d} [qA]\psi \}.$$

So $G(S,p\phi)$ is isomorphic to $G^{\omega}(\phi pq)$.

Proposition 3.4 . *Any triple* (R,r,c) *consisting of a rewriting system* R , *an axiom* r *and a co-root* c *of* G(R,r) , *may be effectively transformed into an another triple* (S,s,d) *such that there exists an isomorphism* f *from* G(S,s) *to the inverse of* G(R,r) *satisfying* f(s) = c *and* f(d) = r .

Proof.

By renaming, we can suppose that two rules in R have not the same label. From theorem 3.2 , we can transform (R,r) in (G,M) where G is a deterministic graph grammar, and M is the left member hyperarc of a rule in G , such that an isomorphism f from G(R,r) to $G^{\omega}(M)$ exists. The system R being 'deterministic' , the co-root c is unambiguously determined by a path from r to c . Provided that we rewrite M sufficiently often, we can assume that it contains f(r) and f(c) . The grammar H is constructed by inverting the right members of the rules in G , i.e. $H = \{ (X,K^{-1}) \mid (X,K) \in G \}$. So $H^{\omega}(M)$ is the inverse graph of $G^{\omega}(M)$, s = f(c) is a root and d = f(r) is a co-root of $H^{\omega}(M)$. Consequently by handing back the renamed labels, the system S constructed in theorem 3.3 works. ◆

theorem 3.6 . *Any pair* (G,M) *of a deterministic graph grammar* G *and of an axiom* M *such that* $G^{\omega}(M)$ *is a finite degree connected graph, may be effectively transformed into a uniform grammar* H *such that* $H^{\omega}(M)$ *is equal to* $G^{\omega}(M)$.

Proof.

As in proof of theorem 3.3 , even if it entails the transformation of G , we can suppose G reduced. As $G^{\omega}(M)$ is connected, by cutting up the rules of G , we can put G in a connected form, that is to say $G^{\omega}(fv_1...v_n)$ is connected for every left member hyperarc $fv_1...v_n$ of a rule in G . We can transform G into the grammar $\{(M,M')\} \cup G'$ such that $M' \in Dom(G')$ and G' is proper : every vertex of an hyperarc X in Dom(G') is also a vertex of any hypergraph in $G^{\omega}(X)$. As in proof of theorem 3.3 , we can put G' , hence G , in Greibach normal form. In addition, we can suppose G has separated outputs, the non-terminal hyperarcs in the right member hypergraph of every rule have different labels, and the right members hypergraphs of the rules of G have no common vertex.

Let S be the vertex set of the right members of G , N the set of its non-terminals, and T the set of its terminals. We define an order $\leq$ on $S.N^*$ preserved by right concatenation, and for all graph C with vertices in $S.N^*$ and every word u in N^* , we write

$C.u = \{ f(s_1u)...(s_nu) \mid fs_1...s_n \in C \}$ the suffixing of the vertices of C by u,

$C.u^{-1} = \{ fs_1...s_n \mid f(s_1u)...(s_nu) \in C \}$ the right quotient of C by u,

S_C the set of vertices in C,

and s_C the greatest suffix in N^* of the vertices of C.

We define a representative $R(G,M)$ of the set $G^\omega(M)$ of the pattern graphs generated by G from M. To do this, let us consider a sequence $(N_n,f_n)_{n\geq 0}$ where N_n is an hypergraph whose vertices are in $S.N^*$ and f_n is an injection of non-terminal hyperarcs of N_n into N^*, and we have $R(G,M) = \{ X \mid \exists i,\ X \in N_i \wedge X(1) \notin N \}$. This sequence is defined as follows :

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

$N_{n+1} = \{ X \in N_n \mid X(1) \notin N \} \cup \bigcup \{ G_X \mid X \in N_n \wedge X(1) \in N \}$ where for every non-terminal hyperarc X of N_n and for every rule (Y,K) of G such that $Y(1) = X(1)$, we have

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

and 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 non-terminal hyperarc of K

$g_X(s) = U(i).V(1).f_n(X)$ if s is the i^{th} vertex of a non-terminal hyperarc V of K and U is the left member hyperarc of a rule in G with the same label as V,

and for every non-terminal hyperarc Z of G_X, $f_{n+1}(Z(1)g_X(Z(2))...g_X(Z(|Z|))) = Z(1).f_n(X)$.

We verify that $R(G,M)$ is well defined and is a member of $G^\omega(M)$.

For every $n \geq 0$, 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 the distance $d(s,M)$ at M is at most $n - 1$. So $G_0 = \emptyset$ and as G is in Greibach normal form, $G_n \subseteq N_n$. The grammar H to be constructed and satisfying the proposition, must be able to generate from M in n steps of parallel rewritings, a graph whose the set of terminal arcs is G_n. With the exception of $M(1)$, a non-terminal of H will be a couple (P,Q) where P is a finite set of terminal arcs with vertices in $S.N^*$, and Q is a subset of vertices of P.

Let $n \geq 1$. We will determine a set $[G_n]$ of non-terminal hyperarcs allowing the generation of the graph $R(G,M) - G_n$ according to H. To do this, we determine the connected components $D_1,...,D_p$ of $N_{n+1} - G_n$. For $1 \leq i \leq p$, we take the set $C_i = \{ fst \in D_i \mid f \notin N \wedge (s \in S_{G_n} \vee t \in S_{G_n}) \}$ of terminal arcs of D_i whose a vertex is also a vertex of G_n. The hypergraph $[G_n]$ is defined by

$$[G_n] = \{ (C_i.(s_{C_i})^{-1} , (S_{C_i} \cap S_{G_n}).(s_{C_i})^{-1}).u_{i,1}...u_{i,q_i} \mid 1 \leq i \leq p \wedge \{u_{i,1},...,u_{i,q_i}\} = S_{C_i} \cap S_{G_n} \wedge \forall j,\ 1 \leq j < q_i,\ u_{i,j} < u_{i,j+1} \}.$$

The grammar H we look for, is defined as the union of a sequence of grammars $(H_n)_{n\geq 0}$. This sequence is inductively constructed as follows :

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

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

where M_n is the set of the non-terminals of $H_1, \ldots, H_n$.

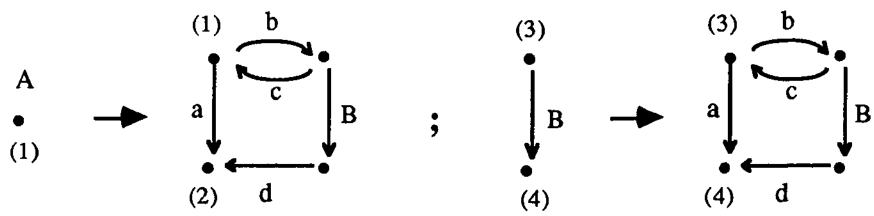
To prove the existence of H , i.e. H_n is empty after a rank n , it suffices to show the existence of a finite number of possible non-terminals for H . To do this, it suffices to find a bound of the distance in R(G,M) of vertices common to C and G_n , for every n and every connected component C of R(G,M) - G_n .

As G is connected, we can take the integer $b = \max\{ d_{R(G,Y)}(s,t) \mid \exists (Y,K) \in G ,\ s,t \in S_K \}$. 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 of b on 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 normal form, this path goes through a vertex v with the same suffix in N^* as u , i.e. $v(2)\ldots v(|v|) = u(2)\ldots u(|u|) = s_C$. Also $d_{R(G,M)}(u,v) \leq d_{R(G,Y)}(u(1),v(1))$ for the rule (Y,K) of G such that u(1) is a vertex in K ; hence $d_{R(G,M)}(u,v) \leq b$. As s is a border vertex of C and G_n , we have

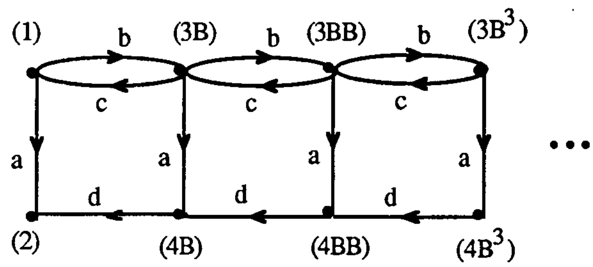
$$d_{R(G,M)}(s,v) + d_{R(G,M)}(v,M) = d_{R(G,M)}(s,M) \leq d_{R(G,M)}(u,M) \leq d_{R(G,M)}(u,v) + d_{R(G,M)}(v,M) .$$

So $d_{R(G,M)}(s,v) \leq d_{R(G,M)}(u,v) \leq b$; hence $d_{R(G,M)}(u,s) \leq d_{R(G,M)}(u,v) + d_{R(G,M)}(v,s) \leq 2b$ and the same holds for $d_{R(G,M)}(u,t)$. Consequently $d_{R(G,M)}(s,t) \leq 4b$, and the existence of H is proved. By construction $M \in \mathrm{Dom}(H)$, H is uniform, and $R(G,M) \in H^\omega(M)$ so $H^\omega(M) = G^\omega(M)$. ♦

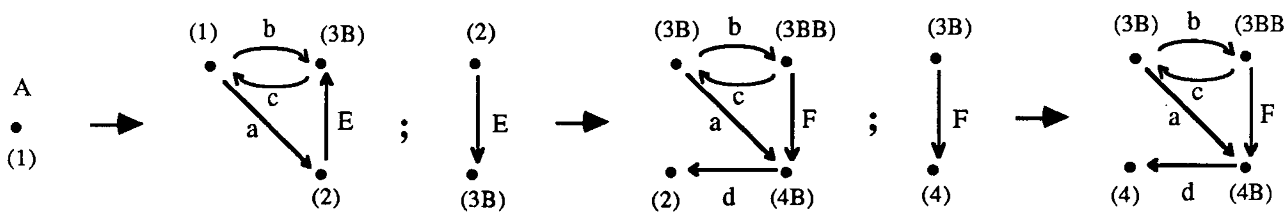
Let us consider the following (non uniform) grammar G :



The representative R(G,A1) of G^{ω}(A1) is the following graph :



The transformation in proof of theorem 3.6 gives the following uniform grammar H :



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

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

with the natural order on {1,2,3,4} extended lexicographically on $\{1,2,3,4\}.B^*$.

Hence H^{ω}(A1) = G^{ω}(A1) .

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

Proposition 3.7 . *From all triples* (R,r,r') *and* (S,s,s') *consisting of a rewriting system, an axiom and a vertex of the generated prefix transition graph, we can decide whether there exists an isomorphism* f *from* G(R,r) *to* G(S,s) *such that* f(r') = s' .

Proof.

From theorems 3.2 and 3.6 , we transform (R,r,r') into a uniform grammar G and an axiom a_0r'

such that the prefix transition graph of R from r be a graph generated by G from a_0r', i.e. $G(R,r) \in G^{\omega}(a_0r')$. In the same way, we transform (S,s,s') into (H,b_0s'). Let us denote N_G [resp. N_H] the set of non-terminals in G [resp. H]. We will now compare the right member hypergraphs of the uniform grammar rules G and H, starting from the ones 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 non-terminal hyperarc of the first one, a non-terminal of the other, up to a permutation of vertices. To do this, we consider the set E of the couples $e = (a,b\pi(1)\ldots\pi(n))$ where a and b are non-terminals of arity n, from G and H respectively, and π is a permutation of $\{1,\ldots,n\}$. To such a couple 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 :

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

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

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

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

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

Hence there exists an isomorphism f of $G(R,r)$ onto $G(S,s)$ 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_01) 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 with $E_{e,h}$ is the ends of arcs of C starting at e. As the set C of such graphs is finite and constructible, we can decide on the isomorphism of $G(R,r)$ and $G(S,s)$ associating r' with s'. ♦

LISTE DES DERNIERES PUBLICATIONS INTERNES

PI 506 VM_pRAY, AN EFFICIENT RAY TRACING ALGORITHM ON DISTRIBUTED MEMORY PARALLEL COMPUTER
Didier BADOUEL, Thierry PRIOL
Janvier 1990, 50 Pages.

PI 507 ON THE REGULAR STRUCTURE OF PREFIX REWRITINGS
Didier CAUCAL
Janvier 1990, 32 Pages.

PI 508 RAY TRACING ON DISTRIBUTED MEMORY PARALLEL COMPUTERS: STRATEGIES FOR DISTRIBUTING COMPUTATIONS AND DATA.
Didier BADOUEL, Kadi BOUATOUCH, Thierry PRIOL
Janvier 1990, 16 Pages.

PI 509 STABILITY ANALYSIS AND IMPROVEMENT OF THE BLOCK GRAM-SCHMIDT ALGORITHM
William JALBY, Bernard PHILIPPE
Janvier 1990, 24 Pages.

PI 510 TESTING FOR THE UNBOUNDEDNESS OF FIFO CHANNELS IN PROGRAMS
Thierry JERON
Janvier 1990, 30 Pages.

PI 511 AUTOMATIC ANIMATION CONTROL OF PHYSICAL SYSTEMS
Georges DUMONT, Bruno ARNALDI, Gérard HEGRON
Janvier 1990, 22 Pages.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique