On the regular structure of prefix rewriting

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 $qA_1 \ldots A_n$ where q is a state of the automaton and A_1 is the top of the stack contents $A_1 \ldots A_n$. 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 .

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 [Ca 90].

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 $u \overset{f}{\longrightarrow} v$ 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 $\{ ab^ic \mid 1 \le i \le n \} \cup cb^*a$.

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

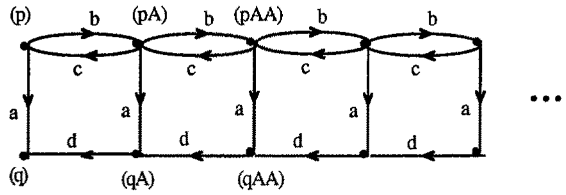

Fig. A Prefix transition graph

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 $\underset{R}{\overset{*}{\longmapsto}}$ 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 $\underset{R}{\overset{*}{\longmapsto}}$ 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

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

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

$H \cup I$ 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 $s \overset{f}{\longrightarrow} t$, the sliced version $\langle G \rangle$ of G is its closure with respect to the following property :

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

then $vu' \xrightarrow{u''} t \in G$ if $vu' = \mathrm{left}(vu'u'')$ 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

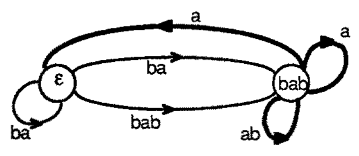

Fig. B Automaton of a prefix rewriting

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

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

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* $z \in \mathrm{Dom}(R) \cup \{\varepsilon\}$

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

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 $\overset{*}{\underset{R}{\longmapsto}}$ 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 :

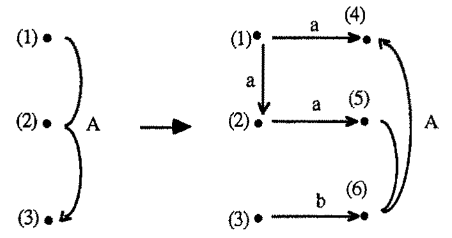

Fig. C Deterministic graph grammar

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 $\{1,\ldots,n\}$. Thus, it makes sense to define steps of complete parallel rewriting $M \Rightarrow_G N$ as follows :

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

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

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

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.

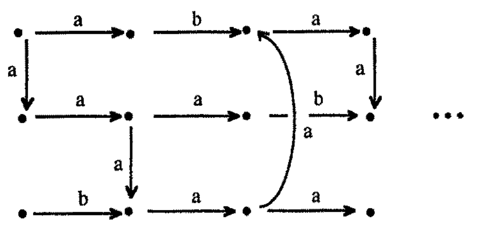

Fig. D Pattern graph

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 $fv_1 \ldots v_n \to H$ 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) $G^{\omega}(fv_1 \ldots v_n)$ 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. Montfort 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.