AUTOMATA AND RATIONAL EXPRESSIONS ON PLANAR GRAPHS[1]

Francis Bossut, Max Dauchet and Bruno Warin
U.A. 369 C.N.R.S.
LIFL, Université de Lille I, UFR d'IEEA, Bat. M3
59655 VILLENEUVE D'ASCQ Cedex FRANCE

Abstract. We study languages (i.e. sets) of planar directed acyclic graphs (pdags). These pdags are constructed by parallel composition and serial composition of letters and pdags on a doubly ranked alphabet. Our purpose is to introduce an algorithmic process (generalization of Kamimura and Slutzki's parallel automata) for accepting pdag languages and a specification of these languages by means of well-suited rational expressions. So our main result is a Kleene-like theorem proving the equivalence between rationality and automaton-definability.

1. Introduction

Directed Acyclic Graphs (dags) are for instance used for unification algorithms (Martelli & Montanari, 1982), to model parallelism (Degano & Montanari, 1985; Grabowsky, 1981; Ochmanski, 1985; Winkowski, 1979), relational data bases, program schemes or to study the complexity of computations (Claus, Ehrig and Rozenberg, 1979; Ehrig, Pfender and Schneider, 1983). Planar dags (pdags) specially model derivations of phrase-structure grammars (Kamimura & Slutzki, 1981). Graph languages have been studied from a categorical point of view (Ehrig, 1979) and Courcelle (1987) have recently studied a notion of recognizability for graph languages.

Our purpose is to introduce an algorithmic process (automata) for accepting pdag languages and a specification of these languages by means of well-suited rational expressions. So our main result is a Kleene-like theorem proving the equivalence between rationality and automaton-definability. For a class of non-planar but very particular dags, such a result is obtained by Ochmanski (1985).

From the automaton point of view, our approach is closely related to Kamimura & Slutzki's work (1981, 1982) but we generalize their pdags. Let us point out that our automaton definability is only algorithmic, it can't be studied as recognizability in the algebraic sense (Courcelle, 1988) because we consider as initial or final controls non bounded sequences of states.

Concerning rationality, we consider the closure of an alphabet of vertices (or nodes) under serial composition (denoted "."), parallel asynchronous composition (denoted &) and their iterations. With these two basic operations, magmoïd (Arnold & Dauchet, 1978, 1979) becomes our implicit framework.

Two important facts might be pointed out. The first one concerns the class of graphs, the second one the nature of Kleene's theorem.
1) our class is between the dags accepted by Kamimura & Slutzki's automata and the general class of graph studied from an algebraic point of view by Courcelle. The class defined by Kamimura & Slutzki is too narrow to lead to a "rational definition": it is not closed under parallel composition. Kamimura & Slutzki only study single rooted pdags when our pdags have a non-bounded number of roots and are

1 : Supported by PRC (Programme de Recherche Coordonnées) mathématique et informatique

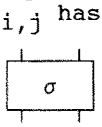
generally non connected. Thus new problems arise about determinism, rational control of roots, connected sub-pdags. On the other hand, the general point of view of Courcelle and Ehrig does not permit an practical definition of automaton.

2) our Kleene-like theorem is not an extension of general results à la Eilenberg. It is, as for Ochmanski, a somewhat difficult and technical result, which establishes the adequation between the specification and the algorithmic points of view. The proof of this theorem underlines the importance of connectivity (we use the decomposition of pdags in connected components).

2. Pdags and pdag automata - Definitions

We will assume that the reader is familiar with the rudiments of tree language theory (Engelfriet, 1975; Thatcher, 1973) and dag automaton à la Kamimura & Slutzki (1981, 1982). We consider labelled, directed, acyclic, ordered and planar graphs that we call pdags. Our labels are symbols from a finite doubly ranked alphabet $\Sigma = \cup \Sigma_{i,j}$. We suppose $i \geq 1$ and $j \geq 1$. A symbol $\sigma \in \Sigma_{i,j}$ has head-rank i and tail-rank j.

we also denote it:



i inputs

j outputs

We connect pdags in series by means of a binary operator denoted "." (or omitted) and we connect pdags in parallel by means of a binary operator denoted "θ". We introduce a special pdag denoted "e" $\in \Sigma$ that allows to define neutral elements for serial composition. Formally, pdag algebra is defined inductively as follows:

2.1. Definition. Let Σ be a finite doubly ranked alphabet and a new symbol **e** of head-rank and tail-rank 1.
The set **DP(Σ)** of pdags over Σ and its subsets $\mathbf{DP(\Sigma)_{i,j}}$ of pdags of head-rank i and tail-rank j, are defined inductively as follows:

(i) $e \in DP(\Sigma)_{1,1}$, if $a \in \Sigma_{i,j}$ then $a \in DP(\Sigma)_{i,j}$,

(ii) Let $d \in DP(\Sigma)_{i,j}$ and $d' \in DP(\Sigma)_{i',j'}$
$(d \theta d') \in DP(\Sigma)_{k,l}$ with $k=i+i'$ and $l=j+j'$,
$(d.d')$ is defined if $j=i'$ and $(d.d') \in DP(\Sigma)_{i,j'}$,

(iii) $DP(\Sigma)$ is the union of all the $DP(\Sigma)_{i,j}$ for $i,j \geq 1$.

Two expressions are equivalent if and only if they are equivalent with respect to the congruence relation generated by these axioms:

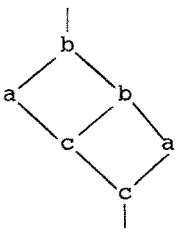
(i) associativity of θ and .

(ii) e is a neutral element.
We note for $p>0$, $e_p = e\theta...e\theta e$, p times,
then for all $\delta \in DP(\Sigma)_{p,q}$, $e_p.\delta=\delta$ and $\delta.e_q=\delta$

(iii) If d.d' and f.f' are defined then $(d.d')\theta(f.f')=(d\theta f).(d'\theta f')$.

2.2. Proposition. Two expressions d and f are equivalent if and only if they denote the same pdag.

Example. Let Σ be such that $\Sigma_{11} = \{a\}$, $\Sigma_{12} = \{b\}$ and $\Sigma_{21} = \{c\}$. The expression $b.(a\theta e).(e\theta b).(c\theta a).c$ is equivalent to $b.(a\theta b).(c\theta a).c$. And they are equivalent to:



Remark. We identify an expression with its class, i.e. with the pdag that this class represents.

Basic operators extended to sets. Let A, B be two sets of pdags then:
$A.B = \{\delta.\delta' / \delta \in A \text{ and } \delta' \in B\}$; $A\Theta B = \{\delta\Theta\delta' / \delta \in A \text{ and } \delta' \in B\}$
$A^* = A \,.\, A^* \cup \{ e_p / p \in N \}$; $A^\Theta = \{\delta_1\Theta\delta_2\Theta\ldots\Theta\delta_n / \delta_i \in A, n>0\}$

A (finite) top-down pdag automaton M is a system <Σ,Q,T,B,R> where:
.**Σ** is a finite doubly ranked alphabet.
.**Q** is a finite set of states. For every state q,
tail-rank(q)=head-rank(q)=1
.**T and B** are two recognizable word languages over Q in the usual sense, with Θ identified to the word concatenation. **T** is called the top control and **B** is the bottom control.
.**R** is a finite set of rules of the form:

$$(p_1\Theta p_2\Theta\ldots\Theta p_m).\sigma \vdash \sigma.(q_1\Theta q_2\Theta\ldots\Theta q_n)$$

with $\sigma \in \Sigma_{m,n}$, $p_1,\ldots p_m, q_1\ldots q_n \in Q$.

Remark about notation. Q^Θ denotes the set of words over Q, usualy denoted by Q^*. So, a word abc... is denoted by aΘbΘc...

Elementary move. We define the binary relation ⊢R− between pdags over Σ ∪ Q as follows: δ ⊢R− δ' if and only if δ' is obtained from δ by replacing, in δ, exactly one occurrence of the left-hand side of a rule of R by its right-hand side. $\vdash R^*-$ denotes the reflexive and transitive closure of ⊢R− .

Pdag language accepted by a top-down automaton. A pdag δ is accepted by M iff, for some $q_1\Theta\ldots\Theta q_m \in T$ and $q'_1\Theta\ldots\Theta q'_n \in B$:

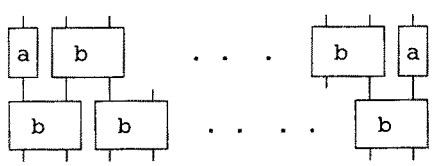
$$(q_1\Theta\ldots\Theta q_m).\delta \vdash R^*- \delta.(q'_1\Theta\ldots\Theta q'_n)$$

L(M) denotes the pdag language accepted by M. We obtain a **bottom-up automaton** by reversing the direction of the elementary move and exchanging the role played by B and T. A pdag language P is said to be **automaton-definable** if and only if P = L(M) for pdag automaton M. The family of automaton-definable languages is denoted by **AD** (Automaton-Definable).

2.3. Example

Let Σ be { 

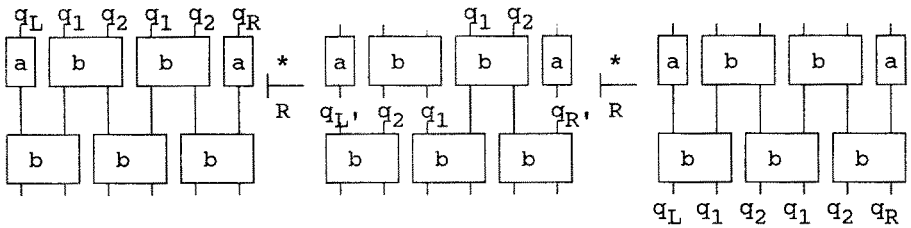
, 

} and δ be 

Then the "set of walls" $W = \{\delta' \in DP(\Sigma) / \delta' = \delta.\delta\ldots\delta\}$ is accepted by the automaton $M = <\Sigma,Q,B,T,R>$ where

$Q = \{ q_L, q_R, q_1, q_2, q_{L'}, q_{R'} \}$; $B = T = q_L \Theta (q_1 \Theta q_2)^\Theta \Theta q_R$
$R = q_L.a \vdash a.q_{L'}$; $(q_{L'}\Theta q_2).b \vdash b.(q_L\Theta q_1)$
$q_R.a \vdash a.q_{R'}$; $(q_1\Theta q_{R'}).b \vdash b.(q_2\Theta q_R)$; $(q_1\Theta q_2).b \vdash b.(q_2\Theta q_1)$

For instance we have the accepting computation:



3. PROPERTIES OF PDAG AUTOMATA

3.0. Preliminaries.

1-Let $A = \langle \Sigma, K, K_0, K_F, \mu \rangle$ be a finite word automaton. K denotes the set of states of A, K_0 and K_F denote the subsets of initial and final states, respectively. μ denotes the set of transitions, they are of the form k.x —> x.k' also denoted (k,x,k') with $x \in \Sigma$, k and $k' \in K$.
For every k and $k' \in K$, let us define
$P_{k,k'} = \{ m \in (K \times \Sigma \times K)^* \mid m=(k,x_1,k_1).(k_1,x_2,k_2)\ldots(k_{n-1},x_n,k') \}$.

2-Let Σ be a product of sets : $\Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_n$.
Let us denote by $\pi_{\Sigma i}$ the morphism from Σ^* to Σ_i^* defined by :

$$\forall (x_1,x_2,\ldots,x_n) \in \Sigma \quad \pi_{\Sigma i}((x_1,x_2,\ldots,x_n)) = x_i$$

3.1. Discussion about top and bottom control. Kamimura & Slutzki's pdags have a single root. So, the top-control does not arise and bottom control is not introduced either. We study unconnected pdags and then we introduce controls B and T to check the relationships between connected components.

3.1.1. Definition. A pdag automaton is said to be without bottom control iff its bottom control B is the set of all the words over $Q_f \subset Q$. Qf is said to be the set of final states of the automaton. A corresponding definition can easily be stated for top control.

The following proposition states that top control **and** bottom control have the same power as top **or** bottom control.
3.1.2. Proposition. For every pdag automaton M, we can construct a pdag automaton M' without bottom control (or top control) that accepts the same pdag language.

Proof. We give a construction that suppresses bottom control for a top-down automaton.
Let M be the pdag automaton $\langle \Sigma, Q, T, B, R \rangle$. We construct a pdag automaton $M' = \langle \Sigma, Q', T', B', R' \rangle$ without bottom control as follows:

Let $A = \langle Q, K, K_0, K_F, \mu \rangle$ be a finite word automaton which accepts B in the usual sense. Then we define

. $Q' = K \times Q \times K$. . $B' = \mu^{\Theta}$.
. $T' = \{ m \in P_{k,k'} \mid k \in K_0,\ k' \in K_F$ and $\pi_Q(m) \in T \}$.
. $R' = \{ m.a \vdash a.m' \mid \pi_Q(m).a \vdash a.\pi_Q(m') \in R$
with m and m' $\in P_{k,k'}$ for some k,k' $\}$

It is obvious that $L(M) \subset L(M')$.
Finally, let us sketch the proof that $L(M') \subset L(M)$:
let δ be a pdag of $L(M')$.

Then there exists $m_i \in T$ and $m_f \in \mu^{\Theta}$ such that:

$$m_i.\delta \vdash R'^{*} \dashv \delta.m_f$$

So m_i and m_f belong to the same $P_{k,k'}$, moreover k belongs to K_0 and k' to K_F, therefore $\pi_Q(m_f)$ belongs to B.
Finally $\pi_Q(m_i).\delta \vdash R^{*} \dashv \delta.\pi_Q(m_f)$ and $\delta \in L(M)$. ∎

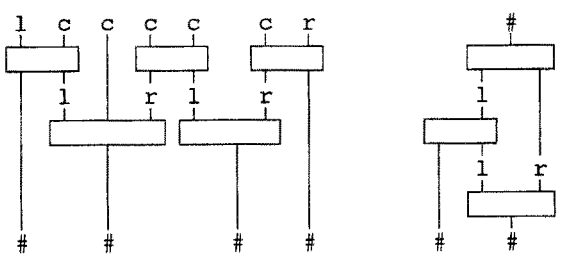
3.2. Connectivity on pdags. In the study of rationality, we shall find out the importance of considering the pdag connected components. Top-down automaton C presented below enables us to solve some problems dealing with connectivity.

Definitions. We shall say that an edge of δ is a **right-hand edge** (respectively a **left-hand edge**) if and only if this edge is the rightmost input (respectively leftmost) of a connected subgraph δ'' of δ such that $\delta = \delta.(e_n \Theta \delta'' \Theta e_p)$ where e_0 denotes the absence of e.

Automaton C. $C = < \Sigma, H, T, B, P_{co} >$ where
$H = \{ 1, c, r, \# \}$, $T = H^{\Theta}$, $B = \#^{\Theta}$,
P_{co} is the union, for all letters x of Σ, of the rules of the following forms:

i) $(1 \Theta c^{\Theta}).x \vdash x.((1 \Theta c^{\Theta} \Theta r + \#)^{\Theta} \Theta 1 \Theta c^{\Theta})$
ii) $c^{\Theta}.x \vdash x.(c^{\Theta} + c^{\Theta} \Theta r \Theta (1 \Theta c^{\Theta} \Theta r + \#)^{\Theta} \Theta g \Theta c^{\Theta})$
iii) $(c^{\Theta} \Theta r).x \vdash x.(c^{\Theta} \Theta r \Theta (1 \Theta c^{\Theta} \Theta r + \#)^{\Theta})$
iv) $(\# + 1 \Theta c^{\Theta} \Theta r).x \vdash x.(\# + 1 \Theta c^{\Theta} \Theta r)^{\Theta}$

Properties of automaton C. The properties that result from the behaviour of automaton C can be introduced by the following accepting computation:



We observe that the rules have been elaborated so that:
. state **r** (resp. **1**) "walks along" right-hand (resp. left-hand) edges;
. state **#** "walks along" right-hand and left-hand edges;
. state **c** "walks along" the ones that are neither right-hand nor left-hand edges.

The following properties are easy to verify
Property 1 : $L(C) = DP(\Sigma)$;
Considering the top control $1 \Theta c^{\Theta} \Theta r + \#$, we obtain:

Property 2 : The subset of DP(Σ) connected pdags denoted **CO(Σ)** is automaton-definable.

If we restrict automaton-definability to connected graphs, the following proposition states that we can consider automata without controls.

Proposition 3.2.1. For any automaton-definable pdag language G, there is a pdag automaton $M = \langle \Sigma, Q, QI^{\Theta}, QF^{\Theta}, R \rangle$ where QI and QF are subsets of Q, which accepts **among the connected pdags** the ones belonging to G.

Construction. According to proposition 3.1.2., there exists an automaton $M_1 = \langle \Sigma, Q_1, T_1, QF_1^{\Theta}, R_1 \rangle$ where QF_1 is a subset of Q_1 such that $L(M_1) = G$.
Let us consider a word automaton $A = \langle Q_1, V, V_i, V_f, \mu \rangle$ which recognizes (word) language T_1. We denote by μ_i transitions of μ beginning by an element of V_i, μ_f transitions ending by an element of V_f, and μ_{if} intersection of μ_i and μ_f.
Then, the searched automaton is the following:

$M = \langle \Sigma, Q, QI^{\Theta}, QF^{\Theta}, R \rangle$ where
- $Q = V \times Q_1 \times V \times H$
- $QI = \mu_i \times \{l\} \cup \mu_f \times \{r\} \cup \mu_{if} \times \{\#\} \cup \mu \times \{c\}$
- $QF = V \times QF_1 \times V \times \{\#\}$
- A rule $w.\delta \vdash \delta.w'$ belongs to R if and only if
.its projection on Q_1 belongs to R_1;
.its projection on H belongs to P_{co};
.the projections of w and w' on $V \times Q_1 \times V$ belongs to the same $P_{vi,vj}$.

Proof. For δ accepted by M, there exists w_i belonging to QI^{Θ} and w_f to QF^{Θ} such that $w_i.\delta \vdash R^{*} - \delta.w_f$. If we suppose δ connected, then:
. $\pi_H(w_i)$ is a word of $l \Theta c^{\Theta} \Theta r + \#$,
. $\pi_{VxQ1xV}(w_i)$ and $\pi_{VxQ1xV}(w_f)$ belong to a same $P_{v,v'}$
So $\pi_{Q1}(w_i)$ is a word of T_1, and therefore δ is a connected pdag of G ■

4. Rationality on pdags

We attempt, as in the case of word or tree languages to describe the automaton-definable pdag languages by "rational expressions". Such a project requires the introduction of new concepts: many-sorted letters and an asynchronous parallel composition operator.

Definitions. A **many-sorted doubly ranked alphabet (Σ,S)** is a finite subset of $S^{*}x\Sigma xS^{*}$, where Σ is a finite set of labels, and S a finite set of sorts. A triplet (s,a,s') belonging to (Σ,S) will be noted

$\begin{matrix} s \\ a \\ s' \end{matrix}$; s is called the coarity and s' the arity of $\begin{matrix} s \\ a \\ s' \end{matrix}$

With such a many-sorted alphabet (Σ,S), we construct by serial and parallel composition a set of graphs that are still planar, acyclic,

ordered, directed but whose nodes and edges are labelled. Arity and coarity definitions are extended in a canonical way to this new class of graphs denoted **DP(Σ,S)**. We also extend the algebraic frame, in such manner that:

- the serial composition operator respects arity and coarity;
- we introduce a new set of symbols $\{e_s,\ s \in S\}$, arity and coarity of e_s are s; for $w = s_1 s_2 \ldots s_n$ in S^+, we denote by e_w the parallel composition of e_{s1} , $e_{s2} \cdots e_{sn}$.
- for every pdag δ, the arity of which is w, $\delta.e_w = \delta$, and for every pdag δ, the coarity of which is w , $e_w.\delta = \delta$.

Example. The pdag of fig. 1 can be written:

$$a^{s_1}_{s_1 s_0} \cdot \left[a^{s_1}_{s_1 s_0} \ \Theta\ a^{s_0}_{s_1 s_0} \right] \cdot \left[b^{s_1 s_0}_{s_1} \Theta\ b^{s_1 s_0}_{s_0} \right] \cdot b^{s_1 s_0}_{s_1}$$

Notation. Henceforth, we shall denote by τ the subset $\{e_w / w \in S^+\}$.

4.1. Rational expressions on (Σ,S). The set of rational expressions on (Σ,S), denoted by Exp(Σ,S), is recursively defined by:

. if $x \in (\Sigma,S)$ then $x \in Exp(\Sigma,S)$
. if exp,exp' $\in Exp(\Sigma,S)$ then
(exp + exp'), (exp . exp'), (exp & exp'), exp^*, $exp^{\&} \in Exp(\Sigma,S)$

So, with the (Σ,S) letters, we construct a set of well-formed expressions. In order to give a "meaning" to these expressions, we must interpret them.

We call **&** the mapping that associates the set

$$(\ E \ \Theta\ E'\ )\ U\ (\ E\ \Theta\ \tau\ )\ U\ (\ \tau\ \Theta\ E')$$

to a pair of subsets E,E' of DP(Σ,S).And **&** superscript the one that associates $(\ E\ U\ \tau\ )^{\Theta}$ to a subset E of DP(Σ,S).

Operator & simulates an **asynchronous** parallel composition for it doesn't require simultaneous presence of elements belonging to the concerned subsets.

The interpretation of pdag rational expressions. To each rational expression is associated a subset of DP(Σ,S) by the following interpretation function $I : Exp(\Sigma,S) \longrightarrow 2^{DP(\Sigma,S)}$ recursively defined by:

. if $x \in (\Sigma,S)$ then $I(x) = \{x\}$
. if exp, exp' $\in Exp(\Sigma,S)$ then
$I(exp+exp') = I(exp) \cup I(exp')$; $I(exp.exp') = I(exp).I(exp')$;
$I(exp^*) = [I(exp)]^*$; $I(exp\&exp') = I(exp)\ \&\ I(exp')$;
$I(exp^{\&}) = [I(exp)]^{\&}$

Notation. For any subset E of D(Σ,S), and any expressions exp, exp' of Exp(Σ,S), we write exp ≡ E if I(exp) = E and exp ≡ exp' if I(exp) = I(exp'). We shall use a particular morphism Φ from **DP(Σ,S)** to **DP(Σ)** that erases edge's sorts.

4.2. Rational pdag-languages. A **rational language of pdags** is given by a triplet (T,exp,B) where T, B are rational word languages of S^*, and exp is a rational expression of Exp(Σ,S). A pdag δ belongs to

(T,exp,B) if and only if there exists a pdag δ' of I(exp) the coarity and arity of which belong respectively to T and B such that $\Phi(\delta')= \delta$.

Example. The rational language:

$$\left[s_0 \,,\, \left[\begin{matrix} s_0 \\ a \\ s_0 s_1 \end{matrix} \right]^{*} \cdot \left[\left[\begin{matrix} s_0 & s_1 \\ a & a \\ s_0 s & s s_1 \end{matrix} \right]^{\&} \right]^{*} \cdot \left[\left[\begin{matrix} s_0 s & s s_1 \\ b & b \\ s_0 & s_1 \end{matrix} \right]^{\&} \right]^{*} \cdot \left[\begin{matrix} s_0 s_1 \\ b \\ s_1 \end{matrix} \right], s_1 \right]$$

represents the set of pdags that have the figure 2 pattern.

The set of rational pdag-languages on Σ will be denoted by **Rat(Σ)** or more simply **Rat** if there is no ambiguity.

4.3. A Kleene like theorem in pdags. In the string case, it is well known that any recognizable language is rational and conversely. We are going to show that our definitions leads to a comparable result in pdags. First, we present lemmas and definitions to prove theorem 4.3.7.

Lemma 4.3.1. If exp $\in$ Exp(Σ,S) and $\delta \in$ I(exp) then:

$$\delta = \delta_1 \ominus \delta_2 \ominus \ldots \ominus \delta_n \Longrightarrow \{\delta_1\} \,\&\, \{\delta_2\} \,\&\ldots\&\, \{\delta_n\} \lessdot I(exp)$$

Corollary. Each δ_i belongs to I(exp).

Definitions. We denote by **CO(Σ,S)** the subset of **DP(Σ,S)** of connected pdags. Given exp belonging to Exp(Σ,S), we call:

C(exp) = I(exp) ∩ CO(Σ,S)
C_l(exp) = { $\delta \in$ CO(Σ,S) / $\exists\ \delta' \in$ DP(Σ,S) : $\delta \ominus \delta' \in$ I(exp) }
C_r(exp) = { $\delta \in$ CO(Σ,S) / $\exists\ \delta' \in$ DP(Σ,S) : $\delta' \ominus \delta \in$ I(exp) }
C_c(exp) = { $\delta \in$ CO(Σ,S) / $\exists\ \delta', \delta'' \in$ DP(Σ,S) : $\delta' \ominus \delta \ominus \delta'' \in$ I(exp) }

Lemma 4.3.2. For exp rational expression, we denote by **Ω(exp)** the set:

$$\mathbf{C(exp)\ \cup\ (C_l(exp) \ominus \tau)\ \cup\ (\tau \ominus C_r(exp))\ \cup\ (\tau \ominus C_c(exp) \ominus \tau)}$$

then, for every rational expression exp: $\mathbf{exp^* \equiv \Omega(exp)^*}$.

Lemma 4.3.3. We can also establish that:

$$\mathbf{C_r(exp^*) = ((\tau \ominus C_c(exp) \ominus \tau) \cup (\tau \ominus C_r(exp)))^* \cap CO(\Sigma,S)}$$
$$\mathbf{C_l(exp^*) = ((\tau \ominus C_c(exp) \ominus \tau) \cup (C_l(exp) \ominus \tau))^* \cap CO(\Sigma,S)}$$
$$\mathbf{C_c(exp^*) = (\tau \ominus C_c(exp) \ominus \tau)^* \cap CO(\Sigma,S)}$$

Definition 4.3.4. We shall say that a subset E of DP(Σ,S) is **(Σ,S)-automaton-definable** if and only if there exists a pdag automaton M = <Σ,QxS,T,B,R> such that:

1 . L(M) = Φ(E)
2 . for every rule $w.a \vdash a.w'$ of R, $a^{\pi_S(w)}_{\pi_S(w')}$ belongs to (Σ,S).

Lemma 4.3.5. For every $E \subset CO(\Sigma,S)$:

E is (Σ,S)-automaton-definable $\Longrightarrow$ $(E^{\&})^*$ is (Σ,S)-automaton-definable

Proof.
Step 1: Let $M_1 = <\Sigma,Q_1,T_1,B_1,R_1>$ be an automaton that verifies points 1, 2 of definition 4.3.4 for E. We apply successively to this automaton the transformations seen in 3.1 (suppression of bottom control) and in 3.5 (supression of top control). Then we obtain an automaton $M_2 = <\Sigma,Q_2,QI^{\Theta},QF^{\Theta},R_2>$ that verifies points 1, 2 of definition 4.3.4 for set E^{Θ}.
Step 2: Let us consider a finite substitution s that transforms an initial state (in QI) into a final state (in QF) without altering its projection on S.
So automaton $M = <\Sigma,Q,(QI \cup QF)^{\Theta},QF^{\Theta},R>$ where

$$R = \{\ w.a \vdash a.s(w')\ /\ w.a \vdash a.w' \in R_2\ \}$$

verifies the points 1,2 of definition 4.3.4 for $(E^{\&})^*$.

As a matter of fact:
- M verifies point 2 : obvious.
- $\Phi((E^{\&})^*) \subset L(M)$: obvious.
- On the other hand , we give only an intuitive argument and leave out formal details. Given a pdag δ ($\neq \tau$) accepted by M, then there exists w_i ($\in QI^{\Theta}$) and w_f ($\in QF^{\Theta}$) such that $w_i.\delta \vdash_R^* \delta.w_f$. Here we admit necessarily a substring v_1 of w_i is such that $\pi_H(v_1) \in (1 \Theta c^{\Theta} \Theta r + \#)$. Let us consider pdag δ_1 accepted from v_1 until final states or initial states resulting from substitution s. It is easy to check that δ_1 belongs to $\Phi(E)$. So δ can be written as the serial composition of a pdag of $\Phi(E^{\&})$ and a pdag δ' of L(M). We can repeat this splitting on δ' and so we conclude that δ belongs to $\Phi((E^{\&})^*)$. ∎

After this sequence of lemmas, we can state the main lemma:

Lemma 4.3.6. For every exp belonging to Exp(Σ,S), I(exp), C(exp), C_L(exp), C_r(exp) and C_c(exp) are (Σ,S)-automaton-definable.

Proof. By induction on the construction of rational expressions,

Kleene-like theorem 4.3.7. For any doubly ranked alphabet Σ, the class RAT of rational languages is exactly the class AD of automaton-definable sets.

Proof.
- The automaton's states can be simulated by sorts in a many sorted alphabet and so AD $\subset$ RAT.
- Conversely: given a rational language (A,exp,Z), we know that I(exp) is (Σ,S)-automaton-definable by lemma 4.3.6.
Let $M = <\Sigma,Q,T,B,R>$ be an automaton which verifies points 1,2 of the definition 4.3.4 for I(exp). Then, if we restrict its top and bottom controls in the following way:

T become $T' = \{\ w \in T\ /\ \pi_S(w) \in A\ \}$ and
B become $B' = \{\ w \in B\ /\ \pi_S(w) \in Z\ \}$

then this new automaton will accept all the pdags of (A,exp,Z) and only these ones.

Figures.

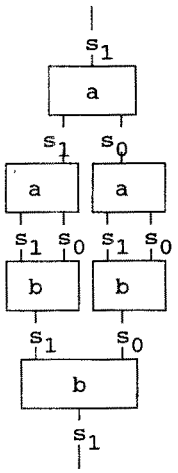


figure 1

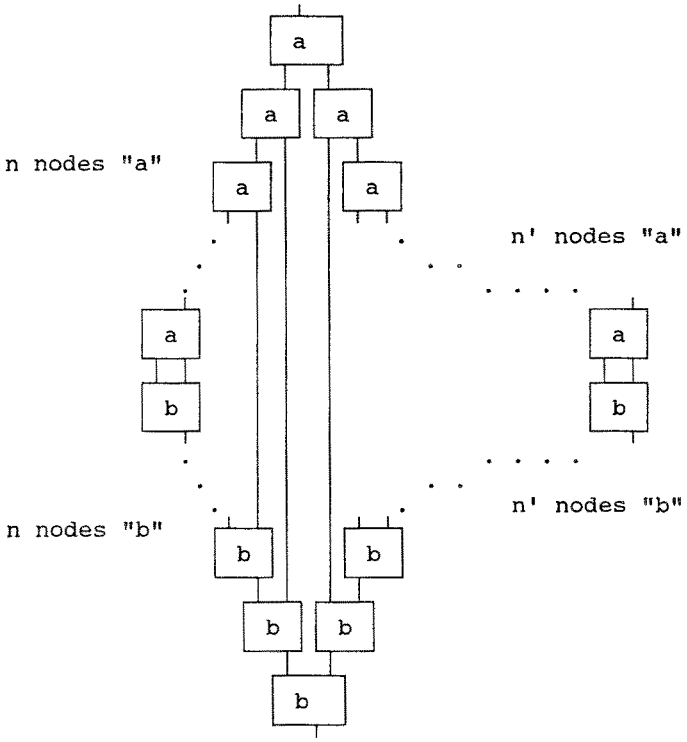


figure 2

REFERENCES

ARNOLD,A. and DAUCHET,M. (1978), Théorie des Magmoides (I), RAIRO Informatique Théorique 12-3, 235-257.
ARNOLD,A. and DAUCHET,M. (1979), Théorie des Magmoides (II), RAIRO Informatique Théorique 13-2, 135-154.
BOSSUT,F. and WARIN.B.(1986), "Rationalité et reconnaissabilité dans des Graphes Acycliques", Doctorat, Université de Lille I.
CLAUS,V., EHRIG,H., ROZENBERG,G.Ed. (1979), "Graph Grammars and their Applications to Computer Sciences and Biology", L.N.C.S. 73.
COURCELLE,B. (1987), Recognizability and Second-Order Definability for Sets of Finite Graphs, Technical Report n° I-8634, University of Bordeaux.
COURCELLE,B. (1988), On recognizable sets and tree automata, To appear in Proceedings of the Colloquium on Resolution Equations in Algebraic Structures, Austin
DEGANO,P. and MONTANARI,V. (1985), Specification Languages for Distribued Systems, TAPSOFT'85-L.N.C.S. 185, 29-51.

EHRIG,H., PFENDER,M., SCHNEIDER,H,J.,Ed. (1983), "Graph Grammars and their Applications to Computer Sciences", L.N.C.S. 153.
EILENBERG,S. and WRIGHT,J.B. (1967), Automata and tree grammars, Inf. and Control 11, 217-231.
ENGELFRIET,J. (1975), Bottom-up and Top-down tree transformations -a comparison, Math Syst Theory 9, 193-231.
GRABOWSKY,J. (1981), On Partial Languages, Fund. Informaticae 4, 427-498.
KAMIMURA,T. and SLUTZKI,G. (1981), Parallel and Two-way Automata on Directed Ordered Acyclic Graphs, Information and Control 49, 10-15.
KAMIMURA,T. and SLUTZKI,G. (1982), Transductions of Dags and Trees, Math. SysT. Theory 15, 225-249.
MARTELLI and MONTANARI,V. (1982), An Efficient Unification Algorithm, Transactions on Programming Languages and Syst., 256-282.
OCHMANSKI,E. (1985), Regular Behaviour of Concurrent Systems, Bull. EATCS 27.
THATCHER,J.W. (1973), Tree automata: Informal Survey, In Currents in the Theory of Computing (A.V,Aho,Ed.), Prentice-Hall, Englewood Cliffs, 143-172.
WINKOWSKI,J. (1979), An Algebraic Approach to Concurrence, MFCS 79-L.N.C.S. 74, 523-532.