Revue française d'informatique et de recherche opérationnelle, série rouge

J. Chauché

# Brève communication. Reconnaissance de sous-arborescences

*Revue française d'informatique et de recherche opérationnelle, série rouge*, tome 5, n° 3 (1971), p. 89-95.

<http://www.numdam.org/item?id=M2AN_1971__5_3_89_0>

(5e année, R-3, 1971, p. 89-95)
R.I.R.O.

# RECONNAISSANCE DE SOUS-ARBORESCENCES

par J. CHAUCHÉ (*)

Sommaire. — *La définition des arborescences comme classe de langage structuré construit sur le monoïde libre engendré par $N^+$ permet de définir précisément les sous-arborescences. On a alors le moyen d'accepter par un automate à pile toutes les arborescences contenant une sous-arborescence donnée. Cette reconnaissance se fait sans contrainte sur la position relative de la sous-arborescence avec l'arborescence donnée.*

## DEFINITION ET PROPRIETES DES ARBORESCENCES ET SOUS-ARBORESCENCES

### 1° Ensembles fondamentaux

— $N^+$ : ensemble des entiers naturels non nuls : $\{1, 2, ..., n, ...\}$.

— $U$ : le monoïde libre engendré par $N^+$ et l'opération de concaténation « $\cdot$ » de deux entiers. « 0 » est l'élément neutre de ce monoïde.

— $g^*$ : groupe des permutations récursives de $N^+$.

— $\mathcal{F}$ : ensemble des applications récursives de $U \to g^*$.

— $\leqslant$ ordre naturel de $N^+$.

— $\underset{\bullet}{\leqslant}$ : ordre défini sur $U$ par :

$$a \underset{\bullet}{\leqslant} b \Leftrightarrow \exists\, x \in U \qquad a \cdot x = b$$

— $l$ : longueur d'un mot, application de $U \to N^+$ défini par :

$$l(0) = 1$$
$$l(W \cdot x) = l(W) + 1 \qquad \forall\, x \in N^+,\ W \in U$$

### 2° Elément d'arborescence

Un élément d'arborescence est une partie finie $A$ de $\mathfrak{T}(U)$ fermée à gauche pour $\underset{\bullet}{\leqslant}$.

(1) IMA Grenoble (G.E.T.A.).

$A$ est un élément d'arborescence $\Leftrightarrow A \in \mathcal{T}(U)$, card $(A)$ est fini et

$$x \in A, y \in U, x \underset{\bullet}{\leqslant} y \Rightarrow y \in A.$$

**3° Fonction d'équivalence**

Soit $f \in \mathcal{F}$, on définit $\varphi_f$ dite fonction d'équivalence comme l'extension naturelle aux parties de $U$ de la fonction $\varphi'_f$ définie par :

$$\varphi'_f(0) = 0$$

$$\varphi'_f(W \cdot x) = \varphi'_f(W) \cdot f_W(x)$$

Où $f_W$ est employée pour $f(W)$ qui est une permutation récursive de $N^+$.

**4° Arborescence**

Une fonction d'équivalence étant une permutation de $U$, on en déduit :

— Une arborescence est une classe d'équivalence pour la relation :

$$A \sim B \Leftrightarrow \exists f \in \mathcal{F} : B = \varphi_f(A)$$

— Élément simple :

Un élément d'arborescence $A$ est dit simple si et seulement si :

$$\forall\, x \cdot i \cdot y \in A, \qquad x, y \in U, \qquad i \in N^+$$

$$j \leqslant i \Rightarrow x \cdot j \cdot y \in A$$

$A$ étant fermé pour $\underset{\bullet}{\leqslant}$ cette définition est équivalente à

$$\forall\, x \cdot i \in A, \qquad j \leqslant i \Rightarrow x \cdot j \in A$$

**5° Théorème**

$\exists$ une fonction récursive $\xi$ de $\mathcal{T}(U) \to \mathcal{F}$ tel que pour tout élément d'arborescence $A$, $\varphi_{\xi(A)}(A)$ est un élément simple.

REMARQUE

Le nombre des éléments simples d'une arborescence est fini.

**6° Fonction d'effacement, sous-arborescence**

— Une fonction d'effacement notée $\mathcal{E}_x$ est l'extension naturelle aux parties de $U$ de la fonction $\mathcal{E}'_x$ définie par :

$$\mathcal{E}'_x(x \cdot y) = y \qquad \forall\, x \cdot y \in U$$

$$\mathcal{E}'_x(y) = 0 \qquad \forall\, y \in U \wedge x \not< y$$

— Une arborescence $\{ B \}$ est une sous-arborescence de $\{ A \}$ si et seulement si : $\exists f \in \mathcal{F}$, $A \in \{ A \}$, $B \in \{ B \}$, $x \in A$ tel que :

$$B \subset \varphi_f(\mathcal{E}_x(A))$$

Cette définition est cohérente c'est-à-dire que si cette propriété est vérifiée par deux éléments $B$ et $A$ de $\{ B \}$ et $\{ A \}$, elle est vérifiée pour tous les éléments de $\{ B \}$ et $\{ A \}$.

**Propriété :**

Soient $\{ B \}$ et $\{ A \}$ deux arborescences et $B$ et $A$ deux éléments quelconques de $\{ B \}$ et $\{ A \}$.

$\{ B \}$ est une sous arborescence de $\{ A \}$ si et seulement si il existe

$$x \in \varphi_{\xi(A)}(A) \quad \text{et} \quad f \in \mathcal{F}$$

tel que

$$\varphi_{\xi(B)}(B) \subset \varphi_f(\mathcal{E}_x(\varphi_{\xi(A)}(A))$$

**7° Représentation linéaire**

Soit $L$ le langage sur $\{ [, ] \}^*$ défini par :

1) $[\,] \in L$

2) $W_1, \ldots, W_n \in L \Rightarrow [W_1, \ldots W_n] \in L$

**Propriété :**

Il existe une bijection $\mathcal{L}$ entre l'ensemble des éléments simples et le langage $L$ tel que $\mathcal{L}(\{ 0 \}) = [\ ]$.

*Définition :*

Si $A$ est un élément quelconque de l'arborescence $\{ A \}$ alors $\mathcal{L}(\varphi_{\xi(A)}(A))$ est un représentant linéaire de l'arborescence $\{ A \}$.

**8° Sous-mot**

$U$ est un sous-mot propre de $W$ si et seulement si $U \in L$, $W \in L$, $\exists V \cdot Z \in L \cup \{ \Lambda \} : W = V \cdot U \cdot Z$

$U$ est un sous-mot de $W$ si et seulement si

$$W = \prod_{i=1}^{2n+1} W_i \qquad U = \prod_{i=1}^{n} W_{2j}$$

avec $W_1 \cdot W_{2n+1} \in L \cup \{ \Lambda \}$ et $\forall h$, $1 < h < n$ $[W_{2h+1}] \in L$

**Propriété :**

$B$ est une sous-arborescence de $\{ A \}$ si et seulement si il existe deux éléments simples $B$ de $\{ B \}$ et $A$ de $\{ A \}$ tel que $\mathcal{L}(B)$ soit un sous-mot de $\mathcal{L}(A)$.

**9° Automate à pile (déterministe)**

Un automate à pile est un sextuplet $T = (V_E, V_p, Q, \delta, q_0, F)$ où

$V_E$ est un ensemble fini vocabulaire d'entrée

$V_p$ est un ensemble fini vocabulaire de pile

$Q$ est un ensemble fini, ensemble des états

$q_0 \in Q$ l'état initial

$F \subset Q$ l'ensemble des états finaux

$\delta$ une application $V_E \times V_p \cup \{ \Lambda \} \times Q \rightarrow V_p^* \times Q$

Une configuration de l'automate est un doublet sur $V_p^* \times Q$.

*Marche de l'automate*

$$U \in V_E^*, \sigma \in V_E, U\sigma : (Z\gamma, q) \vdash U : (ZZ', p) \quad \text{si} \quad \delta(\sigma, \gamma, q) = (Z', p)$$

*Langage accepté*

Si $\overset{*}{\vdash}$ est l'extension réflexive et transitive de $\vdash$, on a :

$$L(T) = \{ W | W : (\Lambda, q_0) \overset{*}{\vdash} \Lambda : (\Lambda, q_f), q_f \in F \}$$

**10° Théorème**

Soit $\{ B \}$ une arborescence quelconque donnée; alors il existe un automate à pile $T$ ne dépendant que de $\{ B \}$ tel que les propriétés suivantes soient équivalentes pour toutes arborescences $\{ A \}$ :

— $\{ B \}$ est une sous-arborescence de $\{ A \}$

— $\mathcal{L}(\varphi_{\xi(A)}(A)) \in L(T) \qquad \forall A \in \{ A \}$

*Démonstration*

On démontre la propriété par récurrence sur le nombre de point de l'arborescence $\{ B \}$.

Si $\{ B \}$ contient un seul point alors $\mathcal{L}(\varphi_{\xi(B)}(B)) = [\,]$, $\forall B \in \{ B \}$, l'automate est donné par :

$$T = (\{ [, ] \}, Q, Q, \delta, q_0, F)$$

$$Q = \{ q_0, q_1, q_2 \}$$

$$F = \{ q_2 \}$$

$$\delta : \delta([, \Lambda, q_0) = (q_1, q_1) \qquad \delta([, q_1, q_1) = (q_1 q_1, q_1)$$

$$\delta(], q_1, q_1) = (\Lambda, q_2) \qquad \delta(\sigma, q_1, q_2) = (\Lambda, q_2) \,\forall\, \sigma \in \{ [, ] \}$$

$$\delta([, x, q_2) = (x, q_1, q_2)$$

$$\delta(], q_1, q_2) = (\Lambda, q_2)$$

Si la propriété est vraie pour toute arborescence $\{ B \}$ contenant au plus $(n - 1)$ points alors une arborescence contenant $n$ points est telle que :

— chaque élément d'arborescence contient au plus $(n - 1)$ entiers et $\mathcal{L}(\varphi_{\xi(B)}(B))$ est de la forme

$$\left[\prod_{i=1}^{n-1} W_i \cdot \right], \quad W_i \in L \cup \{ \Lambda \}$$

chaque $W_i$ correspondant à une arborescence d'au plus $(n - 1)$ points.

Soit $T_i$ les automates reconnaissant $W_i$ alors l'automate $T$ est donné par :

$$T_i = (\{ [, ] \}, V_{pi}, Q_i, q_{0i}, \delta_i, F_i)$$
$$T = (\{ [, ] \}, Q, Q, 1, \delta, \{ q_f \})$$
$$Q = \{ (\sigma, q_1, \ldots, q_{n-1}) | q_i \in \mathcal{P}(Q_i), \sigma \in (\{ 1, \ldots, n\}^2)^{n-1}\} \cup \{ q_f \} \cup \{ 1 \}$$
avec : $\sigma = 1 \Leftrightarrow \sigma = (\sigma_1, \ldots, \sigma_{n-1}) \wedge \sigma_i = (\sigma_{i1}, \sigma_{i2}) = (1,1) \; \forall \, i$

$\delta$ : $\delta([, \Lambda, 1) = ((1, \{ q_{0_1} \}, \ldots, \{ q_{0_{n-1}} \}), (1, \{ q_{0_1} \}, \ldots, \{ q_{0_{n-1}} \}))$

$$\delta([, x, (p, \{ q_1 \}, \ldots, \{ q_{n-1} \})) = (x(p, \{ q_1 \}, \ldots, \{ q_n \}), (p, \{ q'_1 \}, \ldots, \{ q'_{n-1} \})$$

où si

$$x = (p'', \{ q''_1 \}, \ldots \{ q''_{n-1} \})$$

on a

$$q \in \{ q'_i \} \Leftrightarrow q \in \delta([, q''_i, q_i), q''_i \in \{ q''_i \} \qquad q_i \in \{ q_i \}$$

$$\delta(], x, (p, \{ q_1 \}, \ldots, \{ q_{n-1} \})) = (\Lambda, (p', \{ q'_1 \}, \ldots, \{ q'_{n-1} \})$$

où

$$p \neq (\sigma_1, \ldots, \sigma_{n-1}) \text{ tel que } \forall j, (\sigma_{j1} \geqslant \sigma_{j2})$$

et si

$$x = (p'', \{ q''_1 \}, \ldots, \{ q''_{n-1} \})$$

on a :

$$\{ q'_i \} = \{ q''_i \} \cup \{ q'''_i \}$$

avec

$$q \in \{ q'''_i \} \Leftrightarrow q \in \delta_i(], q''_i, q_i), q''_i \in \{ q''_i \}, q_i \in \{ q_i \}$$

$$p' = (\sigma'_1, \ldots, \sigma'_{n-1}) \text{ avec}$$

$$\sigma'_i = \begin{cases} (\sigma''_{i1} + 1, \operatorname{card}(\{ q_j | q_j \in F_i \cap \{ q'''_i \}\}) & \text{si } \{ q'''_i \} \cap F_i \neq \phi \\ \sigma''_i & \text{sinon} \end{cases}$$

$$\delta(], x, (p, \{ q_1 \}, \ldots, \{ q_n \})) = (\Lambda, q_f) \quad \text{si } p = (\sigma_1, \ldots, \sigma_{n-1}) \text{ avec } \forall_i \; \sigma_{i1} \geqslant \sigma_{i2}$$

$$\delta(], x, q_f) = (\Lambda, q_f) \quad , \quad x \in Q$$
$$\delta([, x, q_f) = (x\,1, q_f)$$

**11° Théorème**

Étant donné une arborescence $\{ B \}$, il existe une fonction récursive donnant les états de l'automate à pile $T$ du précédent théorème.

*Démonstration*

La démonstration est évidente du fait que la démonstration du précédent théorème peut se traduire comme un schéma de récursion.

**12° Exemple**

Une formule algébrique peut être mise sous forme arborescente. Reconnaître une sous-arborescence est alors équivalent à reconnaître une sous-formule. Dans l'exemple donné les instructions du système sont :

*e* : entrée d'une expression. Elle est effectuée sur la ligne suivante et en minuscule. La réponse du système est la formule prise en compte qui, elle, est donnée avec des majuscules.

*ls* : localisation d'une sous-formule :

— la formule à reconnaître est donnée en minuscule sur la ligne suivante

— réponse en majuscule :

  — « non » si la formule n'existe pas

  — la sous-formule si elle existe. Elle est alors donnée sous la même forme que dans l'expression générale.

1re expression

$a \cdot x \cdot y + b \cdot y \cdot z + c \cdot z \cdot x$

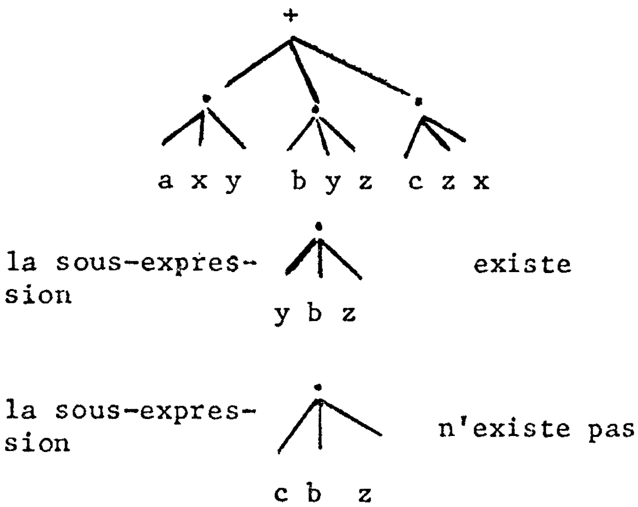


**Figure 1**

```
start phar
EXECUTION BEGINS...
e
a.x.y+b.y.z+c.z.x|
A.X.Y+B.Y.Z+C.Z.X
ls
y.b.z|
B.Y.Z
ls
c.b.z|
NON
e
(x.y.z+y.t.z.a+z.t).(z.r.s+r.y.t.a).(z.x.a.b+r.s.t+t.z.x+a)|
(X.Y.Z+Y.T.Z.A+Z.T).(Z.R.S+R.Y.T.A).(Z.X.A.B+R.S.T+T.Z.X+A)
ls
t.z.x+a.x.z.b+s.r.t+a|
Z.X.A.B+R.S.T+T.Z.X+A
ls
(t.z.x+a.x.z.b+s.r.t+a).(t.z.a.y+t.z+z.x.y)|
(X.Y.Z+Y.T.Z.A+Z.T).(Z.X.A.B+R.S.T+T.Z.X+A)
|
R; T=0.75/1.38 10.02.06
```

**Figure 2**

## BIBLIOGRAPHIE

[1] BARBAULT et DESCLES, *Étude structurelle et catégorielle du système de transition avec application à la science du calcul et à la linguistique mathématique*. Thèse, janvier 1970, Paris.

[2] BARRY K. ROSEN, *Tree manupulating systems and Church-Roser Theorems*. Harvard University, Cambridge Massachussets, 02138.

[3] BRAINERD W. S., *Tree generating regular systems*. I.C. 14 (1969), 217-231.

[4] J. CHAUCHÉ, *Transduction d'arborescence, application aux manipulations de formules sur ordinateur*. Thèse, Grenoble, avril 1971.

[5] W. ROUNDS, *Context-free-Grammars on tree*. ACM Symp on theory of computing (1969), 143-148.

[6] VEILLON, VEYRUNES, VAUQUOIS, *Un métalangage de grammaires transformationnelles*. Document CETA Grenoble.