

ANDRÉ ARNOLD

MAX DAUCHET

**Transductions de forêts reconnaissables  
monadiques, forêts corégulières**

*Revue française d'automatique informatique recherche opérationnelle.  
Informatique théorique*, tome 10, n° R1 (1976), p. 5-28

[http://www.numdam.org/item?id=ITA\\_1976\\_\\_10\\_1\\_5\\_0](http://www.numdam.org/item?id=ITA_1976__10_1_5_0)

© AFCET, 1976, tous droits réservés.

L'accès aux archives de la revue « Revue française d'automatique informatique recherche opérationnelle. Informatique théorique » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

## TRANSDUCTIONS DE FORETS RECONNAISSABLES MONADIQUES FORETS COREGULIERES (\*)

par André ARNOLD<sup>(1)</sup>, Max DAUCHET<sup>(1)</sup>

Communiqué par M. NIVAT

---

Résumé. — On étudie les arbres de dérivation de langages définis par des systèmes de Lindenmayer particuliers en utilisant les transductions d'arbres.

On définit les classes  $M'_1$  et  $M_2$  obtenues par transductions déterministes et non déterministes des forêts reconnaissables monadiques.

Leurs feuillages sont les classes de langages EDTOL-strict et ETOL-strict. La classe  $M'_1$  est engendrée par une classe de grammaires dont la définition est en quelque sorte symétrique de celle des grammaires régulières d'arbres.

### INTRODUCTION

La théorie des forêts [1], [2], [9], [11], qui généralise la théorie des langages, tend à se développer pour des raisons linguistiques et informatiques. Une de ses motivations, essentielle, a été l'étude des arbres de dérivation des mots d'un langage engendré par une grammaire. On peut, par exemple, caractériser les langages algébriques sans mot vide, comme étant les « feuillages » de forêts reconnaissables [2], [9]; la notion de forêt reconnaissable étant une généralisation parfaitement naturelle de la notion de langage reconnaissable ou rationnel. L'étude des forêts permet alors d'obtenir des résultats sur leurs feuillages; ainsi la fermeture d'une classe de forêts par intersection avec une forêt reconnaissable entraîne la fermeture de la classe de ses feuillages par intersection avec un langage rationnel [9]. Il était dès lors tentant de caractériser de manière analogue, comme feuillages de classes de forêts, des classes de langages engendrés par des systèmes de Lindenmayer.

---

(\*) Reçu avril 1975.

(1) Département d'Informatique, Université de Lille I, Villeneuve d'Ascq.

A partir de motivations biologiques, pour formaliser le développement et la croissance d'organismes vivants, A. Lindenmayer a défini en 1968 de nouveaux systèmes de génération (les OL-systèmes) et les langages engendrés (les OL-langages). L'étude mathématique des L-systèmes et des L-langages, généralisations des OL-systèmes et des OL-langages, s'est depuis considérablement développée, comme en témoignent l'ouvrage de synthèse de Herman et Rozenberg [5] et l'abondante bibliographie qui y figure. Parmi les très nombreux L-systèmes, ceux qui nous paraissent le mieux se prêter à l'étude projetée étaient les extensions des OL-systèmes avec tables définies par Rozenberg [10]. Dans ces systèmes – les grammaires ETOL – les dérivations immédiates se font en réécrivant simultanément tous les symboles non-terminaux en utilisant des ensembles de règles d'une grammaire algébrique – les tables. Les arbres de dérivation dans ces grammaires sont déterminés par la séquence des tables utilisées dans les dérivations. Les transformations qui associent les arbres de dérivation aux séquences de tables sont les transductions descendantes d'arbres étudiées par Baker [1], Rounds [9], Thatcher [11].

Le même phénomène apparaît dans l'étude des schémas de programme non récursifs. De façon plus précise, en reprenant les définitions de Luckham, Park et Paterson [7], la valeur d'un calcul d'un schéma non récursif monadique est définie comme l'image par une transduction rationnelle déterministe d'une « séquence d'exécution ».

Dans le cas d'un schéma non récursif non monadique, cette valeur sera alors l'image de la séquence d'exécution par une transduction descendante d'arbres déterministe.

Nous définissons donc les classes de forêts  $M'_1$  et  $M_1$  comme les images par des transductions descendantes déterministes et non déterministes de forêts reconnaissables monadiques (i.e. les forêts reconnaissables dont les arbres se réduisent à une seule branche, et qui peuvent donc s'identifier aux langages rationnels). Les classes  $M'_1$  et  $M_1$  sont évidemment incluses dans les classes  $D'_1$  et  $D_1$ , images par transductions descendantes déterministes et non déterministes de la classe  $D_0$  des forêts reconnaissables [1], [9].

Notre résultat principal est que les classes de feuillages de  $M'_1$  (resp.  $M_1$ ) sont les classes EDTOL-strict (resp. ETOL-strict) dont les langages sont engendrés par des grammaires ETOL déterministes (resp. non déterministes) strictes (i.e. ne contenant pas de règles de la forme  $X \rightarrow \Lambda$ ).

Un autre résultat important, et qui se révèle fécond, est que les forêts de  $M'_1$  peuvent être engendrées par des grammaires que nous appelons corégulières. Ces grammaires corégulières forment une sous-classe de la classe des grammaires algébriques définies par Rounds [9] et leur définition est d'une certaine façon symétrique de celles des grammaires régulières qui engendrent les forêts reconnaissables [2] : avec les grammaires régulières les dérivations

se font par les feuilles, alors qu'avec les grammaires corégulières les dérivations se font par la racine. En restreignant ces notions de grammaire régulière et corégulière au cas des forêts monadiques — i.e. en fait des langages — on retrouve les notions symétriques de grammaire linéaire à droite et à gauche.

Mais, alors que les grammaires linéaires à droite et à gauche engendrent la même classe, celle des langages rationnels, nous montrons que la classe  $D_0$  des forêts reconnaissables engendrées par des grammaires régulières et la classe  $M'_1$  des forêts corégulières engendrées par des grammaires corégulières sont incomparables relativement à l'inclusion.

En restreignant au cas des classes  $M'_1$  et  $M_1$  les propriétés déjà connues pour les classes  $D'_1$  et  $D_1$ , nous obtenons aisément quelques propriétés de fermeture et de décidabilité.

En outre l'utilisation des grammaires corégulières permet de démontrer la fermeture de  $M'_1$  par intersection avec une forêt reconnaissable et par transduction ascendante déterministe.

Bien que le but de ce travail ne soit pas d'étudier les classes ETOL et EDTOL, mais de les caractériser comme classes de feuillages, il est cependant naturel d'examiner les conséquences qu'ont pour leurs feuillages les propriétés des classes  $M'_1$  et  $M_1$ , ne serait-ce que pour vérifier l'intérêt de la caractérisation obtenue. Si, en ce qui concerne la classe ETOL-strict, les résultats que nous avons obtenus sont déjà connus [5], [10], le fait que la classe EDTOL-strict, qui a été moins étudiée, soit fermée par intersection avec un langage rationnel nous paraît être un résultat nouveau.

Depuis que cet article a été rédigé, nous avons pris connaissance des travaux analogues de Downey [3], [4] et d'Engelfriet (cité par Downey dans [4]) qui obtiennent notre théorème 2 ainsi que l'incomparabilité des classes  $D_0$  et  $M'_1$ . Downey donne dans sa thèse [3] une caractérisation grammaticale des classes  $M_1$  et  $M'_1$ , et sa définition des schémas ELBT qui engendrent les forêts de  $M'_1$  est pratiquement identique à celle des grammaires corégulières.

Notre travail comprend six parties. Dans la première nous introduisons les notions préliminaires concernant les arbres et les forêts. Dans la seconde partie nous rappelons la définition des transducteurs descendants d'arbres et nous définissons les classes  $M'_1$  et  $M_1$  que nous situons par rapport aux classes  $D_0$ ,  $D'_1$  et  $D_1$ . La troisième partie est consacrée à la caractérisation des feuillages de ces classes. Nous exhibons dans la quatrième partie quelques propriétés de fermeture et de décidabilité des classes  $M'_1$ ,  $M_1$ , EDTOL-strict, ETOL-strict. Dans la cinquième partie nous rappelons la définition des grammaires d'arbres et dans la sixième partie nous définissons les grammaires corégulières, nous montrons qu'elles engendrent les forêts de  $M'_1$  et en déduisons quelques résultats complémentaires.

## 1. PRELIMINAIRES

Les définitions et résultats de base utilisés ici se trouvent dans divers articles concernant les langages d'arbres tels que : [1], [9], [11]. Nous les rappelons brièvement.

Si  $\Sigma$  est un ensemble fini et  $d$  une application de  $\Sigma$  dans  $N$ ,  $(\Sigma, d)$  est un *ensemble gradué* ou *ensemble d'opérateurs* qu'on notera plus simplement  $\Sigma$ .

On notera  $\Sigma_i$  l'ensemble  $\{\sigma \in \Sigma \mid d(\sigma) = i\}$  appelé ensemble des symboles de  $\Sigma$  de degré  $i$ .

L'ensemble des arbres  $\Sigma^*$  est défini inductivement par

- $\Sigma_0 \subset \Sigma^*$ ,
- $\forall n \forall \sigma \in \Sigma_n, \forall (t_1, \dots, t_n) \in (\Sigma^*)^n, \sigma(t_1, \dots, t_n) \in \Sigma^*$ .

On définit la *profondeur* d'un arbre  $t$ , notée  $|t|$ , par :

- si  $t = \sigma \in \Sigma_0$  alors  $|t| = 1$ ,
- si  $t = \sigma(t_1, \dots, t_n)$  alors  $|t| = 1 + \sup_i |t_i|$ .

Une *forêt* de  $\Sigma^*$  est une partie de  $\Sigma^*$ . Une forêt de  $\Sigma^*$  est *monadique* si  $i \geq 2$  entraîne  $\Sigma_i = \emptyset$ . Un *automate*  $\mathcal{A}$  est un triplet  $(A, \alpha, A_F)$  où

- $A$  est un ensemble fini,
- $A_F$  une partie de  $A$ ,
- $\alpha$  une application qui à tout  $\sigma \in \Sigma$  fait correspondre une application  $\alpha_\sigma : A^{d(\sigma)} \rightarrow A$ .

La *réponse* de l'automate  $\mathcal{A}$  est l'application :

$$\| \cdot \|_{\mathcal{A}} : \Sigma^* \rightarrow A$$

définie inductivement par :

- si  $t = \sigma \in \Sigma_0$  alors  $\|t\|_{\mathcal{A}} = \alpha_\sigma \in A$ ,
- si  $t = \sigma(t_1, \dots, t_n)$  alors  $\|t\|_{\mathcal{A}} = \alpha_\sigma(\|t_1\|_{\mathcal{A}}, \dots, \|t_n\|_{\mathcal{A}})$ .

La forêt *reconnue* par  $\mathcal{A}$  est  $\{t \in \Sigma^* \mid \|t\|_{\mathcal{A}} \in A_F\}$ . Une forêt est dite *reconnaissable* si elle est reconnue par un automate. On notera  $D_0$  la *classe des forêts reconnaissables*. On peut définir, comme dans le cas de la théorie des langages, des automates non déterministes; on démontre qu'ils reconnaissent les mêmes forêts que les automates déterministes [9]. Il est facile de voir comment les notions de forêt, de forêt reconnaissable et d'automate sont des généralisations des notions de langage, de langage reconnaissable ou rationnel, et d'automate fini. On peut même, moyennant quelques précautions d'ordre technique, identifier  $M_0$ , la *classe des forêts reconnaissables monadiques*, et Rat, la classe des langages rationnels.

On appelle *nœud* d'un arbre  $t$  de  $\Sigma^*$ , toute occurrence d'un symbole de  $\Sigma$  dans  $t$ . Si un nœud de  $t$  est l'occurrence d'un symbole  $\sigma$ , on dira que ce nœud est *étiqueté* par  $\sigma$ . Les *feuilles* d'un arbre sont les nœuds étiquetés par des symboles de degré 0. Tout arbre de  $t$  de  $\Sigma^*$  s'écrivant de façon unique  $\sigma(t_1, \dots, t_n)$  [ou  $\sigma$  avec  $\sigma \in \Sigma_0$ ], sa *racine* est l'occurrence de  $\sigma$  qui n'est un nœud d'aucun  $t_i$  [ou l'unique occurrence de  $\sigma$ ].

A tout arbre  $t$  de  $\Sigma^*$  on associe un mot du monoïde libre  $\Sigma_0^*$  appelé le feuillage de  $t$ , noté  $\Phi(t)$ , et obtenu en prenant les étiquettes des feuilles de  $t$  de gauche à droite. Plus précisément  $\Phi: \Sigma^* \rightarrow \Sigma_0^*$  est défini inductivement par :

- si  $t \in \Sigma_0 \subset \Sigma^*$  alors  $\Phi(t) = t$ ,
- si  $t = \sigma(t_1, \dots, t_n)$  alors  $\Phi(t) = \Phi(t_1) \times \dots \times \Phi(t_n)$ ,

où  $\times$  est la concaténation dans le monoïde libre.

Si  $F$  est une forêt,  $\Phi(F) = \{ \Phi(t) \mid t \in F \}$  est un langage de  $\Sigma_0^*$ . Si  $\mathcal{F}$  est une classe de forêts,  $\Phi(\mathcal{F}) = \{ \Phi(F) \mid F \in \mathcal{F} \}$  est une classe de langages.

Le théorème suivant est classique ([2] par exemple). Il est à rapprocher du théorème bien connu de Mezei et Wright [8].

**Théorème 1**

*La classe des feuillages des forêts reconnaissables est identique à celle des langages algébriques ne contenant pas le mot vide.*

Si  $E$  est un ensemble quelconque,  $\Sigma^*(E)$ , l'ensemble des arbres de  $\Sigma^*$  indexés par  $E$ , est l'ensemble des arbres dont les feuilles sont étiquetées par  $\Sigma_0 \cup E$ , c'est-à-dire  $\Sigma^*(E) = \Sigma'^*$  avec  $\Sigma'$  défini par :

$$\begin{aligned} \Sigma'_i &= \Sigma_i \text{ pour } i > 0, \\ \Sigma'_0 &= \Sigma_0 \cup E. \end{aligned}$$

On appellera *ensemble des variables* l'ensemble  $X = \{ x_i \mid i \in \mathbb{N} \}$ . Dans le cas où un arbre de  $\Sigma^*$  est indexé par  $X$  on dira qu'il est indexé, sans autres précisions. En particulier tout arbre de  $\Sigma^*$  est un arbre indexé. Si  $t$  est un arbre indexé, on pourra distinguer certaines occurrences des variables en l'écrivant  $t(x_{i_1}, \dots, x_{i_n})$ . Si  $t_1, \dots, t_n$  sont des arbres indexés, l'arbre  $t(t_1, \dots, t_n)$  obtenu en substituant les arbres  $t_j$  aux feuilles distinguées de  $t(x_{i_1}, \dots, x_{i_n})$  est encore un arbre indexé (des arbres différents peuvent être substitués à des feuilles différentes étiquetées par la même variable). On dira alors que  $t(x_{i_1}, \dots, x_{i_n})$  est un *sous-arbre initial* de  $t(t_1, \dots, t_n)$ . La relation « être un sous-arbre initial » est une relation d'ordre sur les arbres indexés.

On dira que l'arbre indexé  $t$  est un *sous-arbre terminal* de l'arbre indexé  $t'$  (on dira aussi plus simplement *sous-arbre*) si il existe un arbre indexé  $t''(x_0)$  tel que  $t' = t''(t)$ . En toute rigueur cette relation, qui est aussi une relation d'ordre, est une relation entre occurrences de sous-arbres.

## 2. TRANSDUCTIONS DE FORETS RECONNAISSABLES

Les transducteurs d'arbres que nous allons définir ci-dessous et que nous utiliserons sont des transducteurs *descendants*. Nous parlerons donc simplement de transducteur au lieu de transducteur descendant.

Un *transducteur*  $\tau$  est un quintuplet  $\langle \Delta, \Sigma, Q, Q_0, \Pi \rangle$  où

$\Delta$  et  $\Sigma$  sont des alphabets gradués,

$Q$  est un ensemble fini d'états,

$Q_0 \subset Q$  est l'ensemble des états initiaux,

$\Pi$  est un ensemble fini de *productions*.

Une *production* est un triple  $\langle q; \sigma(x_1, \dots, x_n); t \rangle$  où

$$\sigma \in \Delta_n,$$

$$q \in Q,$$

$$t \in \Sigma^*(Q \times \{x_1, \dots, x_n\}).$$

Pour faire apparaître toutes les feuilles de  $t$  qui ne sont pas étiquetées par des symboles de  $\Sigma_0$ , on pourra écrire  $t$  sous la forme :  $t(q_{i_1}, x_{i_1}), \dots, (q_{i_p}, x_{i_p})$  de telle sorte que  $\Phi(t) = w_0(q_{i_1}, x_{i_1})w_1 \dots w_{p-1}(q_{i_p}, x_{i_p}) w_p$  où tous les  $w_i$  sont dans  $\Sigma_0^*$ . Cette écriture détaillée de  $t$  est unique.

Un transducteur est *déterministe* si :

$$\forall q \in Q, \forall \sigma \in \Delta \langle q; \sigma(x_1, \dots, x_n); t \rangle \in \Pi \quad \text{et} \quad \langle q; \sigma(x_1, \dots, x_n); t' \rangle \in \Pi$$

entraîne  $t = t'$ .

Un transducteur est *linéaire* si quelle que soit la production  $\langle q; \sigma(x_1, \dots, x_n); t \rangle$  de  $\Pi$ , la même variable  $x_i$  n'apparaît pas deux fois dans  $t$ .

Si  $\tau$  est un transducteur défini sur une forêt monadique, la partie droite de toute production de  $\tau$  est un arbre de  $\Sigma^*(Q \times X)$  où  $X$ , l'ensemble des variables est un singleton.  $Q \times X$  est alors isomorphe à  $Q$ . On pourra donc se contenter d'étiqueter les feuilles de tels arbres par des éléments de  $Q$ . D'autre part on n'est pas obligé non plus de faire apparaître la variable en partie gauche de la production. La règle  $\langle q; \alpha(x); t((q_{i_1}, x), \dots, (q_{i_n}, x)) \rangle$  s'écrira alors  $\langle q; \alpha; t(q_{i_1}, \dots, q_{i_n}) \rangle$ .

A tout transducteur  $\tau$  on associe une relation, notée  $\vdash_\tau$ , sur  $[\Delta^*(Q \times \Delta^*)] \times [\Sigma^*(Q \times \Sigma^*)]$  définie par :

–  $(q, \sigma(t_1, \dots, t_n)) \vdash_\tau t'$  ssi il existe

$$t = t((q_{i_1}, x_{i_1}), \dots, (q_{i_p}, x_{i_p})) \text{ tel que } \langle q; \sigma(x_1, \dots, x_n); t \rangle \in \Pi$$

$$\text{et } t' = t((q_{i_1}, t_{i_1}), \dots, (q_{i_p}, t_{i_p})),$$

–  $\sigma((q_1, t_1), \dots, (q_i, t_i), \dots, (q_n, t_n)) \vdash_\tau \sigma((q_1, t_1), \dots, t'_i, \dots, (q_n, t_n))$

$$\text{ssi } (q_i, t_i) \vdash_\tau t'_i.$$

En notant  $\xrightarrow[\tau]^*$  la fermeture transitive de  $\xrightarrow[\tau]$ , on peut définir sur  $\Delta^* \times \Sigma^*$  une relation notée  $\xrightarrow[\tau]$  par :

$$t_1 \xrightarrow[\tau] t_2 \text{ ssi } \exists q_0 \in Q_0 \text{ tel que } (q_0, t_1) \xrightarrow[\tau]^* t_2.$$

Si  $F$  est une partie de  $\Delta^*$  on notera  $\tau(F)$  la forêt  $\{ t \in \Sigma^* \mid \exists t' \in F; t' \xrightarrow[\tau] t \}$  qu'on appellera la *transductée* de  $F$ .

Si le transducteur  $\tau$  est déterministe, la relation  $\xrightarrow[\tau]$  est fonctionnelle, pas forcément totale, (i.e.  $\forall t \text{ Card } \{ t' \mid t \xrightarrow[\tau] t' \} \leq 1$ ).

- On note  $NT$  la classe des transducteurs,
- $DT$  la classe des transducteurs déterministes,
- $NLT$  la classe des transducteurs linéaires,
- $DO$  la classe des transducteurs déterministe à un seul état.

On note  $D'_1 = DT(D_0)$ , la classe des transductées déterministes de forêts reconnaissables,

$D_1 = NT(D_0)$ , la classe des transductées de forêts reconnaissables,

$M'_1 = DT(M_0)$ , la classe des transductées déterministes de forêts reconnaissables monadiques,

$M_1 = NT(M_0)$ , la classe des transductées de forêts reconnaissables monadiques.

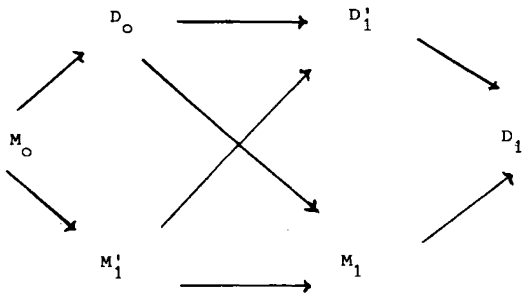
En tenant compte de l'identité entre  $M_0$  et  $\text{Rat}$ , la classe  $M_1$  [resp.  $M'_1$ ] sera aussi appelée la classe des transductées [déterministes] de langages rationnels.

On a manifestement les inclusions suivantes :

$$\begin{aligned} M_0 &\subsetneq D_0 \subset D'_1 \subset D_1, \\ M_0 &\subsetneq M'_1 \subset M_1, \\ M'_1 &\subset D'_1, \\ M_1 &\subset D_1. \end{aligned}$$

On a aussi [11]

$$D_0 \subset M_1.$$



Ces relations sont rassemblées dans la figure 1 où la flèche représente la relation d'inclusion.

**Proposition 1**

*La classe des transductées déterministes de langages rationnels n'est pas incluse dans la classe des forêts reconnaissables.*



**Corollaire 1**

*La classe des forêts reconnaissables est strictement incluse dans la classe des transductées déterministes de forêts reconnaissables et dans celle des transductées de langages rationnels.*

*En effet.*

On considère le langage rationnel  $a^*b$  et le transducteur déterministe à un seul état  $q$  dont les productions sont :

$$\begin{aligned} &\langle q; b; b \rangle, \\ &\langle q; a; c(q, q) \rangle. \end{aligned}$$

On a  $\tau(b) = b$  et  $\tau(a^{n+1}b) = c(\tau(a^n b), \tau(a^n b))$ . On en déduit que  $\Phi(\tau(b)) = b$  et  $\Phi(\tau(a^{n+1}b)) = (\Phi(\tau(a^n b)))^2$ , d'où  $\Phi(\tau(a^n b)) = b^{(2^n)}$  et donc

$$\Phi(\tau(a^*b)) = \{ b^{(2^n)} \mid n \geq 0 \}.$$

Si  $M'_1$  était inclus dans  $D_0$ ,  $\Phi(M'_1)$  serait inclus dans  $\Phi(D_0)$  et donc dans Alg; or  $\{ b^{(2^n)} \mid n \geq 0 \}$  n'est pas un langage algébrique, d'où une contradiction.

Si  $D_0 = D'_1$  alors  $M'_1 \subset D'_1 = D_0$ . Si  $D_0 = M_1$  alors  $M'_1 \subset M_1 = D_0$ . Dans les deux cas on aboutit à une contradiction.  
*cqfd.*

**Proposition 2**

*La classe des forêts reconnaissables n'est pas incluse dans la classe des transductées déterministes de langages rationnels.*

**Corollaire 2**

*La classe des transductées déterministes de langages rationnels est strictement incluse dans la classe des transductées de langages rationnels et dans celle des transductées déterministes de forêts reconnaissables.*

*En effet.*

Soit  $\Sigma = \Sigma_0 \cup \Sigma_2$  avec  $\Sigma_0 = \{ \# \}$  et  $\Sigma_2 = \{ \sigma \}$ .

Supposons que la forêt reconnaissable  $\Sigma^*$  soit la transductée déterministe d'un rationnel  $R$  inclus dans  $\Omega^*$ , c'est-à-dire  $\Sigma^* = \tau(R)$ . A fortiori on aura  $\Sigma^* = \tau(\Omega^*)$ . A chaque état  $q$  du transducteur  $\tau = \langle \Omega, \Sigma, Q, Q_0, \Pi, \rangle$ , on

associe la forêt  $F_q = \{ t \in \Sigma^* \mid \exists w \in \Omega^* tq(q, w) \vdash^* t \}$ . On pose

$Q_\infty = \{ q \mid F_q \text{ est une forêt infinie} \}$  et  $p = \sup \{ |t| \mid t \in F_q, q \in Q - Q_\infty \}$ ;

d'où si  $(q, w) \vdash^* t$  avec  $|t| > p$  alors  $q \in Q_\infty$ .

A chaque production  $p = \langle q; \alpha; t \rangle$  du transducteur, on associe le nombre  $r_p$  de nœuds de  $t$  étiquetés par  $\sigma$  et on pose  $r = \sup_{p \in \pi} r_p$ . Il est évident que  $r \neq 0$ .

Considérons maintenant un arbre  $t$  de  $\Sigma^*$  de la forme  $u_n(v_1, v_2, \dots, v_{2^n})$  où  $\forall i \in [1, 2^n] |v_i| > p$  et où  $u_n$  est l'arbre « équilibré » de profondeur  $n$  défini par

$$\begin{cases} u_0 = x \\ u_{n+1} = \sigma(u_n, u_n) \end{cases}$$

dont le feuillage est bien  $x^{(2^n)}$ .

Par hypothèse, il existe  $w \in \Omega^*$  et  $q_0 \in Q_0$  tel que  $(q_0, w) \stackrel{*}{\vdash}_\tau t$ . En notant  $w_i$  le mot formé des  $i$  premières lettres de  $w$ , on a :

$$(q_0, w_i) \stackrel{*}{\vdash}_\tau t_i(q_{i_1}, \dots, q_{i_k})$$

où  $t_i$  est un sous-arbre initial de  $t$ . La suite  $t_i$  est une suite croissante de sous-arbres initiaux de  $t$  qui converge vers  $t$ . Pour chaque arbre  $t_i$ ,  $n_i$  est le nombre de nœuds de  $t_i$  étiquetés par  $\sigma$  et  $k_i$  le nombre de feuilles de  $t_i$  étiquetées par des états de  $Q_\infty$ . Les suites  $n_i$  et  $k_i$  sont croissantes et  $\forall i k_i \leq n_i + 1$ .

Si  $n_i \leq n$  alors  $t_i$  est un sous-arbre initial de  $u_n$ . D'où si  $n \geq r$ ,  $t_1$  est un sous-arbre initial de  $u_n$ . Comme  $t$  n'est pas un sous-arbre initial de  $u_n$ , il existe  $i_0$  tel que  $t_{i_0}$  est un sous-arbre initial de  $u_n$  et  $t_{i_0+1}$  n'est pas un sous-arbre initial de  $u_n$ .

Puisque  $t_{i_0}$  est un sous-arbre initial de  $u_n$ , tout sous-arbre terminal de  $t$  dont la racine est une feuille de  $t_{i_0}$  est de profondeur supérieure à  $p$ . Toutes les feuilles de  $t_{i_0}$  sont donc étiquetées par des états de  $Q_\infty$  et donc  $k_{i_0} = n_{i_0} + 1$ .

Pour passer de  $t_{i_0}$  à  $t_{i_0+1}$  on va donc appliquer simultanément  $k_{i_0}$  productions du transducteur, d'où :

$$n_{i_0+1} \leq n_{i_0} + k_{i_0} \cdot r = (r + 1)k_{i_0} - 1$$

Comme  $t_{i_0+1}$  n'est pas un sous-arbre initial de  $u_n$ ,  $n_{i_0+1} > n$  d'où  $(r + 1)k_{i_0} > n + 1$ .

Soit  $m$  le nombre d'éléments de  $Q_\infty$ . Si  $n$  est supérieur à  $m(r + 1)$  alors  $(r + 1)k_{i_0} > m(r + 1) + 1$  et  $k_{i_0} > m + \frac{1}{r + 1}$  d'où  $k_{i_0} > m + 1$ .

Le sous-arbre  $t_{i_0}$  contient donc deux feuilles étiquetées par le même état de  $Q_\infty$  qui donneront donc le même sous-arbre dans la transduction puisque le transducteur est déterministe. Il s'ensuit que  $t$  contient au moins deux sous-arbres identiques de profondeur supérieure à  $p$ .

On a donc démontré que si  $\Sigma^*$  est la transductée déterministe d'un rationnel, il existe deux entiers  $n_0$  et  $p_0$  tels que tout arbre  $t$  de  $\Sigma^*$  de la forme :

$$u_n(v_1, \dots, v_{2^n}) \quad \text{où } n \geq n_0 \quad \text{et} \quad |v_i| > p_0$$

a au moins deux sous-arbres identiques de profondeur supérieure à  $p_0$  dont la racine est un nœud de  $u_n$ .

Or  $\Sigma^*$  est tel que pour tout  $n$  et tout  $p$  il existe un arbre  $t$  de  $\Sigma^*$  de la forme :

$$u_n(v_1 \dots v_{2^n}) \quad \text{avec} \quad |v_i| > p$$

tel que tous les sous-arbres de profondeur supérieure à  $p$  dont la racine est un nœud de  $u_n$  sont différents : il suffit que  $v_i$  soit de profondeur égale à  $p + 2n_i$ .

Donc  $\Sigma^* \notin M'_1$  et  $D_0 \notin M'_1$ .

Si  $M'_1 = M_1$  alors  $D_0 \subset M_1 = M'_1$ . Si  $M'_1 = D'_1$  alors  $D_0 \subset D'_1 = M'_1$ . Dans les deux cas on aboutit à une contradiction.

*cqfd.*

### Proposition 3

*La classe des transductées déterministes de forêts reconnaissables est strictement incluse dans celle des transductées de forêts reconnaissables.*

*En effet.*

En utilisant la décomposition des transducteurs  $NT = NLT \circ DO$  donnée par B. Baker [1], on montre que  $D_1 = D'_1$  entraîne  $NT(D_1) = NT(D'_1) = D_1$ . Or on sait que  $D_1$  est strictement incluse dans  $NT(D_1)$ .

*cqfd.*

Pour que toutes les inclusions de la figure 1 soient strictes, il reste à montrer que  $M_1 \subset D_1$  ce que nous conjecturons être vrai. Nous conjecturons aussi que, comme les classes  $D_0$  et  $M'_1$  (propositions 1 et 2), les classes  $D'_1$  et  $M_1$  sont incomparables pour l'inclusion.

### 3. LES FEUILLAGES DE $M'_1$ ET $M_1$

Dans cette partie nous montrons que les classes de feuillages de  $M'_1$  et  $M_1$  sont exactement les classes de langages EDTOL-strict et ETOL-strict dont nous rappelons une définition :

Une grammaire ETOL est un quadruplet  $\langle V, T, \mathcal{P}, \sigma \rangle$  avec

- $V$  vocabulaire fini,
- $T \subset V$  vocabulaire terminal,
- $\sigma \in V - T$ .

$\mathcal{P}$  est un ensemble fini de *tables* :  $\{ P_1, \dots, P_n \}$  où chaque  $P_i$  est un ensemble fini de règles d'une grammaire algébrique (i.e.  $P_i \subset (V - T) \times V^*$ ).

Une grammaire ETOL est *déterministe* si toutes les règles d'une même table ont le symbole gauche différent. Elle est *stricte* si toutes ses tables sont des parties de  $(V - T) \times V^+$ .

Appliquer une table  $P_i$  de  $\mathcal{P}$  à un mot  $w$  de  $V^*$  consiste à réécrire *tous* les non-terminaux de  $w$  qui se trouvent en partie gauche d'une règle de  $P_i$  en utilisant une règle de  $P_i$ . On peut définir alors le langage engendré par une grammaire ETOL comme l'ensemble des mots de  $T^*$  obtenus à partir de  $\sigma$  par des applications successives de tables.

Ces définitions sont légèrement différentes de celles données dans [10]. Il est cependant facile de démontrer qu'elles sont équivalentes.

Les classes de langages ETOL [-strict] et EDTOL [-strict] seront les classes de langages engendrés par des grammaires ETOL [strictes] et des grammaires ETOL déterministes [strictes].

On sait [10] que la classe ETOL-strict est celle des langages de ETOL ne contenant pas le mot vide. Après Latteux [6], nous redémontrons un résultat analogue pour la classe EDTOL (proposition 12).

## Théorème 2

*La classe de langages ETOL-strict [EDTOL-strict] est celle des feuillages des transductées [déterministes] de langages rationnels.*

*En effet.*

A)  $EDTOL\text{-strict} \subset \Phi(M_1)$ ;  $ETOL\text{-strict} \subset \Phi(M_1)$

Soit une grammaire ETOL stricte  $G = \langle V, T, \mathcal{P}, \sigma \rangle$ . On considère le langage rationnel  $\mathcal{P}^*$  et le transducteur  $\tau$  défini par :

- ensemble des états :  $\{ q_A \mid A \in V - T \}$ ,
- état initial :  $q_\sigma$ ,
- ensemble des productions (noté  $\Pi_\tau$ ) :

pour tout  $A \in V - T$  et  $P_i \in \mathcal{P}$ .

- Si  $A \rightarrow z_1 \dots z_n \in P_i$ , où  $\forall i z_i \in V$ , alors  $\langle q_A; P_i; A(z'_1, \dots, z'_n) \rangle \in \Pi_\tau$  où

$$z'_i = \begin{cases} z_i & \text{si } z_i \in T \\ q_{z_i} & \text{si } z_i \in V - T \end{cases}$$

— Si  $A$  n'est pas partie gauche d'aucune règle de  $P_i, \langle q_A; P_i; q_A \rangle \in \Pi_\tau$ .

Par construction  $\tau$  est déterministe si et seulement si la grammaire  $G$  est déterministe.

La démonstration de  $L(G) = \Phi(\tau(\mathcal{F}^*))$  se base sur le fait qu'à chaque dérivation  $D: w_1 \xrightarrow{*} w_2$  de  $G$  on peut associer un mot  $\gamma_D$  de  $\mathcal{F}^*$  formé des tables de  $\mathcal{F}$  dans l'ordre dans lequel elles ont été utilisées; la dérivation  $D: w_1 \xrightarrow{*} w_2$  se notera aussi  $w_1 \xrightarrow[\gamma_D]{} w_2$  et on a  $w_1 \xrightarrow[\gamma]{} w_2 \xrightarrow[\gamma']{} w_3 = w_1 \xrightarrow[\gamma\gamma']{} w_3$ .

On démontre par induction sur la longueur de  $\gamma$  la proposition

(P1):  $\forall A \in V - T, \forall w \in T^*, \quad A \xrightarrow[\gamma]{} w \text{ ssi } \exists t \text{ tq. } (q_A, \gamma) \vdash_\tau^* t \text{ et } \Phi(t) = w$

(voir la démonstration en annexe).

On en déduit que  $w \in L(G)$  ssi  $w \in \Phi(\tau(\mathcal{F}^*))$ . En effet, si  $w \in L(G)$ , il existe  $\gamma \in \mathcal{F}^*$  tel que  $\sigma \xrightarrow[\gamma]{} w$  et donc d'après (P1)  $\exists t$  tq.  $(q_\sigma, \gamma) \vdash_\tau^* t$  avec  $\Phi(t) = w$  et comme  $\gamma \rightarrow_\tau t \in \tau(\mathcal{F}^*)$ ,  $w \in \Phi(\tau(\mathcal{F}^*))$ .

Réciproquement si  $w \in \Phi(\tau(\mathcal{F}^*))$ , il existe  $\gamma \in \mathcal{F}^*$  et  $t$  tel que  $\gamma \rightarrow_\tau t$  et  $w = \Phi(t)$  d'où  $(q_\sigma, \gamma) \vdash_\tau^* t$  et donc  $\sigma \xrightarrow[\gamma]{} \Phi(t) = w$ .

On a donc  $\text{ETOL-strict} \subset \Phi(M_1)$  et  $\text{EDTOL-strict} \subset \Phi(M'_1)$ .

B)  $\Phi(M'_1) \subset \text{EDTOL-strict}; \Phi(M_1) \subset \text{ETOL-strict}$

On sait que tout langage rationnel est l'image par un homomorphisme alphabétique strict d'un langage local. Il est facile d'en déduire que si une forêt  $F$  est la transductée [déterministe] d'un langage rationnel, c'est aussi la transductée [déterministe] d'un langage local. En effet, on sait [11] que le composé d'un homomorphisme strict et d'une transduction [déterministe] est encore une transduction [déterministe].

Soit donc  $K$  un langage local défini par :

$$K = \Sigma^{(0)}\Sigma^* \cap \Sigma^*\Sigma^{(1)} - \Sigma^*\Sigma^{(2)}\Sigma^* \quad \text{où } \Sigma^{(0)} \subset \Sigma, \Sigma^{(1)} \subset \Sigma, \Sigma^{(2)} \subset \Sigma^2$$

et  $\tau = \langle \Sigma, \Omega, Q, Q_0, \Pi \rangle$  un transducteur de  $\Sigma^*$  dans  $\Omega^*$ .

On construit une grammaire ETOL stricte  $G = \langle V, T, \mathcal{F}, \sigma_0 \rangle$  avec  $V = \Sigma \times Q \cup \{ \sigma_0 \}$  où  $\sigma_0 \notin \Sigma \times Q$ , et avec  $T = \Omega_0$ .

Les tables de cette grammaire sont :

— si  $a \in \Sigma^{(0)}$

$$Pa = \{ \sigma_0 \rightarrow (a, q_0) \mid q_0 \in Q_0 \}$$

- si  $(a, a') \in \Sigma^2 - \Sigma^{(2)}$

$$Paa' = \{ (a, q) \rightarrow w_{i,a'} \mid \langle q; a; t \rangle \in \Pi \}$$

où  $w_{i,a'}$  est le mot obtenu en remplaçant dans le feuillage de  $t$ ,  $q_i$  par  $(a', q_i)$ , c'est-à-dire que si  $t = t(q_{i_1}, \dots, q_{i_p})$  alors  $w_{i,a'} = \Phi(t((a', q_{i_1}), \dots, (a', q_{i_p})))$

- si  $a \in \Sigma^{(1)}$

$$P'a = \{ (a, q) \rightarrow \Phi(t) \mid \langle q; a; t \rangle \in \Pi \}$$

(puisque  $a \in \Sigma^{(1)}$ , on aura  $\Phi(t) \in \Omega_0^*$ ).

Par construction, cette grammaire est déterministe si et seulement si le transducteur  $\tau$  est déterministe.

Étant donné qu'un symbole non terminal de  $G$  de la forme  $(a, q)$  est uniquement en partie gauche des tables  $Paa'$  ou  $P'a$ , et qu'une table  $Pa'$  ou  $Paa'$  ne permet d'obtenir que des symboles non-terminaux de la forme  $(a', q)$ , si  $(a, q) \Rightarrow w$  alors  $\gamma$  est un sous-mot initial d'un mot de la forme

$$Paa_{i_1} Pa_{i_1} a_{i_2} \dots Pa_{i_n} a' P'a' = \bar{P}_\alpha, \text{ où } \alpha = aa_{i_1} a_{i_2} \dots a_{i_n} a'$$

est un sous-mot terminal d'un mot de  $K$ .  $\gamma$  n'est pas forcément  $\bar{P}_\alpha$  tout entier car il se peut qu'on obtienne un mot  $w$  de  $T^*$  en terminant la dérivation de  $w$  par une table qui ne contient pas que des règles terminales. Toutefois, si on applique à  $w$  n'importe quelle table, on obtient encore  $w$ . Aussi on pourra considérer que si  $w$  se dérive de  $(a, q)$  dans la grammaire ETOL  $G$ , il se dérive par  $\bar{P}_\alpha$  où  $\alpha \in K_* = \Sigma^* \Sigma^{(1)} - \Sigma^* \Sigma^{(2)} \Sigma^*$ . ( $K_*$  est l'ensemble des sous-mots terminaux de mots de  $K$ ).

On démontre par induction sur la longueur de  $\alpha \in K_*$

$$(P2) : \forall q \in Q, (\alpha_1, q) \xRightarrow{\bar{P}_\alpha} w \text{ ssi } \exists t \text{ tel que } (q, \alpha) \vdash_\tau^* t \text{ et } \Phi(t) = w$$

( $\alpha_1$  étant la première lettre de  $\alpha$ ).

(La démonstration de P2 est pratiquement identique à celle de P1.)

On en déduit alors que  $w \in \Phi(\tau(K))$  ssi  $w \in L(G)$ .

En effet si  $w \in L(G)$ , la première table appliquée dans la dérivation de  $w$  est une table de la forme  $Pa$ . Il existe donc  $(a, q) \in \Sigma^{(0)} \times Q_0$  et  $\alpha$  tel que  $(a, q) \xRightarrow{\bar{P}_\alpha} w$  où  $a$  est la première lettre de  $\alpha$ .

Comme  $a \in \Sigma^{(0)}$ ,  $\alpha$  est dans  $K$ . D'après (P2) il existe  $t$  tel que  $(q, \alpha) \vdash_\tau^* t$  avec  $w = \Phi(t)$  et donc  $\Phi(t) \in \Phi(\tau(\alpha)) \subset \Phi(\tau(K))$ .

Réciproquement, si  $t \in \tau(\alpha)$  avec  $\alpha \in K$ , il existe  $q_0 \in Q_0$  tel que  $(q_0, \alpha) \vdash_\tau^* t$ .

D'après P2  $(\alpha_1, q_0) \xRightarrow{\bar{P}_\alpha} \Phi(t)$ . Comme  $\alpha \in K$ , sa première lettre  $\alpha_1$  appartient à  $\Sigma^{(0)}$  et  $\sigma_0 \rightarrow (\alpha_1, q_0) \xRightarrow{\bar{P}_\alpha} \Phi(t)$  est une règle de la table  $P\alpha_1$  d'où  $\sigma_0 \xRightarrow{P\alpha_1, \bar{P}_\alpha} \Phi(t)$  et donc  $\Phi(t) \in L(G)$

*cqfd.*

#### 4. QUELQUES PROPRIETES DES CLASSES $M_1, M'_1$ , ETOL-STRICT, EDTOL-STRICT

Les résultats figurant dans cette partie et concernant les classes ETOL-strict et EDTOL-strict sont déjà connus ou se déduisent aisément de résultats déjà connus, essentiellement dus à Rozenberg [5], [10]. Toutefois leurs démonstrations utilisent les propriétés des classes de forêts  $M_1$  et  $M'_1$  dont nous venons de montrer qu'elles ont pour feuillage les classes ETOL-strict et EDTOL-strict.

##### Proposition 4

*Les classes  $M_1, M'_1$ , ETOL-strict, EDTOL-strict, sont fermées par union.*

*En effet :*

La réunion de deux langages rationnels est encore un langage rationnel. On peut toujours se ramener au cas où ces deux langages n'ont aucun symbole commun.

De même on peut faire en sorte que deux transducteurs n'aient pas de symbole d'état commun. La réunion de ces deux transducteurs (définie trivialement) est encore un transducteur et on a bien :

$$\tau_1 \cup \tau_2(R_1 \cup R_2) = \tau_1(R_1) \cup \tau_2(R_2)$$

*cqfd.*

##### Proposition 5

- (i)  $H_1$  est fermée par transduction linéaire
- (ii)  $M'_1$  est fermée par transduction déterministe
- (iii) ETOL-strict et EDTOL-strict sont fermées par homomorphisme strict
- (iv)  $H_1$  est fermée par intersection avec une forêt reconnaissable
- (v) ETOL-strict est fermée par intersection avec un langage rationnel.

*En effet.*

Pour démontrer que  $D_1$  est fermée par transduction linéaire et  $D'_1$  par transduction déterministe, Rounds procède ainsi [9] : étant donné une forêt reconnaissable  $F$  et deux transducteurs  $\tau_1$  et  $\tau_2$  qui vérifient certaines propriétés il construit une forêt reconnaissable  $F'$  et un transducteur  $\tau_3$  tels que

$\tau_3(F') = \tau_2(\tau_1(F))$ . Il suffit donc de vérifier que si  $F$  est monadique,  $F'$  reste monadique, ce qui est trivial.

Par ailleurs, il est clair que pour tout homomorphisme strict  $\phi$ , il existe un transducteur d'arbres  $\tau$  tel que  $\phi(\Phi(t)) = \Phi(\tau(t))$  pour tout arbre  $t$ ; ce transducteur  $\tau$  ne modifie que les feuilles d'un arbre; il est déterministe et linéaire d'où (iii) est une conséquence de (i), (ii) et du théorème 2.

On sait [9] que toute forêt reconnaissable  $F$  est le domaine d'un transducteur linéaire non déterministe  $\tau$  qui réalise l'application identique sur son domaine. En appliquant  $\tau$  à une forêt quelconque on obtient bien l'intersection de cette forêt avec  $F$ . Comme  $M_1$  est fermée par transduction linéaire,  $M_1$  est fermée par intersection avec une forêt reconnaissable.

De plus si  $R$  est un langage rationnel quelconque  $\{t \in \Sigma^* \mid \Phi(t) \in R\}$  est une forêt reconnaissable [9]. D'où si  $F \in M_1$ ,  $\Phi(F) \cap R = \Phi(F \cap \Phi^{-1}(R))$  et  $F \cap \Phi^{-1}(R)$  appartenant encore à  $M_1$ ,  $\Phi(F) \cap R$  appartient à ETOL-strict. *cqfd.*

Si on applique une transduction linéaire à une forêt de  $M'_1$ , on obtient une forêt de  $M_1$  qui n'est pas nécessairement dans  $M'_1$ . La méthode précédente ne permet donc pas de montrer que  $M'_1$  est fermée par intersection avec une forêt reconnaissable. Cependant nous démontrerons ultérieurement cette propriété (proposition 9). Nous l'affirmons donc dès maintenant et en déduisons :

### Proposition 6

*EDTOL-strict est fermée par intersection avec un langage rationnel.*

### Proposition 7

*La classe de langages ETOL-strict est strictement incluse dans la classe des langages à contexte lié.*

Cette propriété vient de ce que  $\text{ETOL-strict} \subset \Phi(D_1)$  et que  $\Phi(D_1)$  est strictement incluse dans la classe des langages à contexte lié [1].

### Proposition 8

- (i) *Pour toute forêt  $F$  de  $M_1$  ou de  $M'_1$ ,  $F = \emptyset$  et  $\text{Card}(F) = \infty$  sont des propriétés décidables.*
- (ii) *Pour tout langage  $L$  de ETOL-strict ou de EDTOL-strict,  $L = \emptyset$  et  $\text{Card}(L) = \infty$  sont des propriétés décidables.*
- (iii) *Pour toutes forêts  $F_1, F_2$  de  $M_1$  [ou de  $M'_1$ ],  $F_1 \cap F_2 = \emptyset$  et  $F_1 = F_2$  sont des propriétés indécidables.*

*En effet.*

Les propriétés (i) et (ii) ont été démontrées par Rounds [9] dans le cas plus général des forêts de  $D_1$  et des langages de  $\Phi(D_1)$ .

Pour montrer la propriété (iii) dans le cas de la classe  $D_1$ , Rounds [9] associe à tout problème de Post deux forêts  $F_1$  et  $F_2$  telles que  $F_1 \cap F_2 \neq \emptyset$  ssi le problème de Post a une solution. En fait ces deux forêts sont dans  $M'_1$ . Il en est de même pour l'égalité de deux forêts.

*cqfd.*



## 5. GRAMMAIRES D'ARBRES

Soient deux alphabets gradués disjoints  $\Sigma_N$  et  $\Sigma_T$  dont les éléments sont appelés respectivement symboles non-terminaux et symboles terminaux. Une grammaire sur  $\Sigma = \Sigma_N \cup \Sigma_T$  est un couple  $(X_0, \mathcal{R})$

où  $\left\{ \begin{array}{l} X_0 \text{ est un symbole de } \Sigma_N \text{ de degré } 0 \text{ (} X_0 \text{ est appelé l'axiome)} \\ \mathcal{R} \text{ est un ensemble fini de règles.} \end{array} \right.$

Une règle est un couple  $(\sigma(x_1, \dots, x_n); t)$

où  $\left\{ \begin{array}{l} \sigma \text{ est un symbole non-terminal de degré } n, \\ x_1, \dots, x_n \text{ sont des variables,} \\ t \text{ un arbre de } \Sigma^*(x_1, \dots, x_n). \end{array} \right.$

Comme pour les transducteurs on pourra écrire  $t(x_{i_1}, \dots, x_{i_p})$  au lieu de  $t$ . On dira que l'arbre  $u$  se dérive immédiatement à partir de  $u'$  en utilisant la règle  $r = (\sigma(x_1, \dots, x_n); t)$  (ce qui se notera  $u' \xrightarrow{r} u$ ) si

- a)  $u' = \sigma(t_1, \dots, t_n)$  et  $u$  est obtenu en remplaçant  $x_i$  par  $t_i$  dans  $t$  c'est-à-dire si  $t = t(x_{i_1}, \dots, x_{i_p})$  alors  $u = t(t_{i_1}, \dots, t_{i_p})$
- b)  $u' = \gamma(u'_1, \dots, u'_{i-1}, u'_i, u'_{i+1}, \dots, u'_p)$   
 $u = \gamma(u'_1, \dots, u'_{i-1}, u_i, u'_{i+1}, \dots, u'_p)$   
 et  $u'_i \xrightarrow{r} u_i$

On dira que  $u' \Rightarrow u$  si il existe une règle  $r$  telle que  $u' \xrightarrow{r} u$ . On définit alors la relation de dérivation  $\Rightarrow$  par  $(\Rightarrow)^*$  ce qui permet de définir la forêt engendré par une grammaire comme l'ensemble des arbres de  $\Sigma_T^*$  dérivables à partir de l'axiome.

Les grammaires et les forêts qu'elles engendrent (définies par Rounds [9] qui les appelle créatives) généralisent les grammaires et langages algébriques et seront donc appelées *algébriques*.

Une grammaire *régulière* est une grammaire où  $\Sigma_N$  ne contient que des symboles de degré 0. Les règles d'une grammaire régulière sont donc de la forme  $\sigma \rightarrow t$  où  $t$  est un arbre dont seules les feuilles peuvent être étiquetées par des non-terminaux. Les grammaires régulières généralisent les grammaires linéaires à droite <sup>(1)</sup>. Brainerd a montré [2] que les grammaires régulières engendrent exactement la classe des forêts reconnaissables.

(1) Si l'on prend la convention déjà utilisée implicitement, que si un mot est représenté sous forme d'arbre monadique, la racine de l'arbre est la première lettre du mot.

6. GRAMMAIRES ET FORETS COREGULIÈRES

Pour généraliser la notion de règle linéaire à gauche nous définissons les règles corégulières.

Une règle  $(\sigma(x_1, \dots, x_n); t)$  est *corégulière* si

- $t$  ne contient aucun symbole non-terminal; la règle est alors dite terminale,
- $t$  ne contient qu'un seul symbole non-terminal qui est sa racine; la règle est alors dite non-terminale.

Une *grammaire corégulière* est une grammaire dont toutes les règles sont corégulières.

Une *forêt* est *corégulière* si elle est engendrée par une grammaire corégulière.

**Théorème 3**

*La classe des forêts corégulières est exactement celle des transductées déterministes de langages rationnels.*

*En effet.*

A) Soit une grammaire corégulière  $G = (X_0, \mathcal{R})$  sur  $\Sigma = \Sigma_N \cup \Sigma_T$ . On lui associe le langage des dérivations  $L_D(G) \subset \mathcal{R}^*$  défini par :

$$r_{i_n} \dots r_{i_2} r_{i_1} \in L_D(G) \quad \text{ssi} \quad \exists u_1, \dots, u_{n-1} \in \Sigma^*, \quad u_n \in \Sigma_T^*$$

tel que

$$X_0 \xrightarrow{r_{i_1}} u_1 \xrightarrow{r_{i_2}} u_2 \dots \xrightarrow{r_{i_{n-1}}} u_{n-1} \xrightarrow{r_{i_n}} u_n$$

Il est clair que  $L_D(G)$  est un langage local.

Construisons alors un transducteur  $\tau$  de  $\mathcal{R}^*$  dans  $\Sigma_T^*$  dont l'état initial est  $q_0$  et l'ensemble des autres états est

$$\{ q_{\sigma, i} \mid \sigma \in \Sigma_N, d(\sigma) > 0, 1 \leq i \leq d(\sigma) \}$$

L'ensemble  $\Pi_\tau$  des productions du transducteur est défini par :

- si  $r = \langle \sigma(x_1, \dots, x_n); t \rangle$  est une règle terminale de  $\mathcal{R}$  alors  $\langle q_0; r; t' \rangle \in \Pi_\tau$

où  $t'$  est obtenu en remplaçant chaque variable  $x_k$  par  $q_{\sigma, k}$  dans  $t$ .

- si  $r = \langle \sigma(x_1, \dots, x_n); \sigma'(t_1, \dots, t_p) \rangle$  est une règle non-terminale de  $\mathcal{R}$  pour chaque  $i \in [1, p]$

$\langle q_{\sigma, i}; r; t'_i \rangle \in \Pi_\tau$  où  $t'_i$  est obtenu en remplaçant chaque variable  $x_k$  par  $q_{\sigma, k}$  dans  $t_i$ .

- si  $r = \langle X_0; \sigma'(t_1, \dots, t_p) \rangle$  est une règle non-terminale de  $\mathcal{R}$ , (les  $t_i$  sont donc tous dans  $\Sigma_T^*$ ) alors pour chaque  $i \in [1, p]$   $\langle q_{\sigma',i}; r; t_i \rangle \in \Pi_r$ .
- si  $r = \langle X_0; t \rangle$  est une règle terminale de  $\mathcal{R}$  alors  $\langle q_0; r; t \rangle \in \Pi_r$ .

## REMARQUE

On peut toujours supposer que  $X_0$  est la seule variable non-terminale de degré 0 et qu'elle ne figure jamais en partie droite d'une règle (et donc qu'il n'y a pas d'autres types de règles de  $R$  que les quatre indiqués ci-dessus). En effet si ce n'était pas le cas on aurait alors des dérivations :

$$X_0 \xrightarrow{*} X \xRightarrow[r]{} t$$

et on pourrait remplacer  $r$  par  $r' = \langle X_0; t \rangle$ .

Le transducteur ainsi construit est évidemment déterministe puisque la partie droite de la production  $\langle q_{\sigma,i}; r; t \rangle$  est totalement déterminée par  $r$  et  $i$ .

Le transducteur vérifie aussi la propriété :

(P3) si  $\gamma$  est un sous-mot terminal d'un mot de  $L_D(G)$  dont la première lettre est une règle  $\langle \sigma'(x_1, \dots, x_n); \sigma(t_1, \dots, t_p) \rangle$  alors  $\forall i \in [1, p] \exists t_i$  tel que

$$(q_{\sigma,i} \gamma) \vdash_{\tau}^* t_i.$$

En effet.

Si  $\gamma$  est de longueur 1, alors  $\gamma$  est nécessairement  $\langle X_0; \sigma(t_1, \dots, t_p) \rangle$  avec  $t_i \in \Sigma_T^*$ . Dans ce cas pour chaque  $i$ ,  $\langle q_{\sigma,i}; \gamma; t_i \rangle$  est une règle du transducteur et donc  $(q_{\sigma,i} \gamma) \vdash_{\tau}^* t_i$ .

Si  $\gamma = r\gamma'$  est de longueur  $n + 1$ , alors  $r$  est une règle  $\langle \sigma'(x_1, \dots, x_n); \sigma(t_1, \dots, t_p) \rangle$  et la première règle de  $\gamma'$  est de la forme :

$$\langle \sigma''(x_1, \dots, x_n); \sigma'(t'_1, \dots, t'_n) \rangle.$$

Par l'hypothèse de récurrence on a :  $\forall i \in [1, n] (q_{\sigma',i} \gamma') \vdash_{\tau}^* t'_i$ .

D'autre part le transducteur contient les productions  $\langle q_{\sigma,i}; r; \bar{t}_i \rangle$  pour  $i \in [1, p]$  où  $\bar{t}_i$  est obtenu en remplaçant  $x_j$  par  $q_{\sigma',j}$  dans  $t_i$ .

Donc  $(q_{\sigma,i} r\gamma') \vdash_{\tau}^* \hat{t}_i$  où  $\hat{t}_i$  est obtenu en remplaçant  $x_j$  par  $t'_j$  dans  $t_i$ .  
cqfd.

On démontre par induction sur la longueur de  $\gamma$  la propriété :

(P4) si  $\sigma \in \Sigma_N$ ;  $t_1, \dots, t_n \in \Sigma_T^*$  avec  $n = d(\sigma)$  et si  $\gamma$  est un sous-mot terminal d'un mot de  $L_D(G)$  alors

$$X_0 \xRightarrow[\gamma]{} \sigma(t_1, \dots, t_n) \quad \text{ssi} \quad \forall i \in [1, n] (q_{\sigma,i} \gamma) \vdash_{\tau}^* t_i$$

(La démonstration est analogue à celle de P3)

On en déduit que  $t \in F(G)$  ssi  $t \in \tau(L_D(G))$  où  $F(G)$  est la forêt corégulière engendrée par  $G$ .

En effet.

Si  $t \in F(G)$  il existe  $\gamma \in L_D(G)$  tel que  $X_0 \xrightarrow{\gamma} t$ .

$\gamma = r\gamma'$  avec  $r = \langle \sigma(x_1, \dots, x_n); t'(x_{i_1}, \dots, x_{i_p}) \rangle$  et  $X_0 \xrightarrow{\gamma'} \sigma(t_1, \dots, t_n)$ .

De plus  $t = t'(t_{i_1}, \dots, t_{i_p})$ . D'après P3  $(q_{\sigma, i_p}, \gamma') \vdash_{\tau}^* t_i$  pour tout  $i$ .

Par ailleurs  $\langle q_0; r; t'(q_{\sigma, i_1}, \dots, q_{\sigma, i_p}) \rangle$  est une production du transducteur  $\tau$ .

Donc  $(q_0, r\gamma') \vdash_{\tau}^* t'(t_{i_1}, \dots, t_{i_p}) = t$ ; et donc  $t \in F(G)$  entraîne  $t \in \tau(L_D(G))$ .

Réciproquement si  $(q_0, \gamma) \vdash_{\tau}^* t$  alors  $\gamma = r\gamma'$  avec

$$r = \langle \sigma(x_1, \dots, x_n); t'(x_{i_1}, \dots, x_{i_p}) \rangle.$$

On en déduit en appliquant P3 et P4 l'existence de  $t_1, \dots, t_n$  tels que :  $t = t'(t_{i_1}, \dots, t_{i_p})$  et  $X_0 \xrightarrow{\gamma'} \sigma(t_1, \dots, t_n)$ .

Par ailleurs :  $\sigma(t_1, \dots, t_n) \xrightarrow{r} t'(t_{i_1}, \dots, t_{i_p})$  d'où  $X_0 \xrightarrow{\gamma} t$  et donc  $t \in \tau(L_D(G))$  entraîne  $t \in F(G)$ .

cqfd.

Les forêts corégulières sont donc toutes dans  $M'_1$ .

B) Soit  $K \subset \Sigma^*$  un langage rationnel qu'on peut supposer local (cf. B de la démonstration du théorème 2) et  $\tau$  un transducteur déterministe de  $\Sigma^*$  dans  $\Omega^*$ , dont l'ensemble des états est  $Q = \{q_1, \dots, q_n\}$ . Puisque  $\tau$  est déterministe, pour tout  $a \in \Sigma$  et  $q \in Q$ , il existe au plus un arbre  $t$  de  $\Omega^*(Q)$  tel que  $\langle q; a; t \rangle$  est une production de  $\tau$ . On posera donc

$$t_{q,a} = \begin{cases} t & \text{si } \langle q; a; t \rangle \text{ est une production de } \tau, \\ \# & \text{sinon, où } \# \text{ est un symbole nouveau n'appartenant pas à } \\ & \Omega \cup \Sigma \cup Q. \end{cases}$$

On construit une grammaire corégulière dont l'ensemble des symboles terminaux est  $\Omega \cup \{\#\}$ , en attribuant à  $\#$  le degré 0, et l'ensemble des symboles non-terminaux est  $\{X_0\} \cup \{\bar{a} \mid a \in \Sigma\}$ . Le degré de  $\bar{a}$  pour tout  $a$ , est  $n = \text{Card}(Q)$ . On pourra donc prendre  $Q$  comme ensemble de variables.

Les règles de cette grammaire sont :

– si  $a \in \Sigma^{(1)}$

$$r_a = \langle X_0; \bar{a}(t_{q_1,a}, \dots, t_{q_n,a}) \rangle,$$

– si  $(a, a') \in \Sigma^2 - \Sigma^{(2)}$

$$r_{a'a} = \langle \bar{a}'(q_1, \dots, q_n); \bar{a}(t_{q_1, a'}, \dots, t_{q_n, a}) \rangle,$$

– si  $a \in \Sigma^{(0)}$

pour tout  $q_i$  état initial du transducteur

$$r_{a q_i} = \langle \bar{a}(q_1, \dots, q_n); q_i \rangle.$$

Si  $\alpha = a_{i_1} \dots a_{i_n}$  est un sous-mot terminal d'un mot de  $K$  on pose

$$r_\alpha = r_{a_{i_n}} r_{a_{i_n} a_{i_{n-1}}} \dots r_{a_{i_2} a_{i_1}}.$$

On démontre par les mêmes méthodes que précédemment :

$$X_0 \xRightarrow[r_\alpha]{} \bar{a}_{i_1}(t_1, \dots, t_n) \text{ ssi } \forall i \in [1, n] (q_i, \alpha) \xrightarrow[\tau]{} t_i$$

où  $\bar{\tau}$  est le transducteur  $\tau$  auquel on a rajouté les règles manquantes  $\langle q; a; \# \rangle$ .

On en déduit, par des méthodes déjà utilisées, que  $F(G) = \bar{\tau}(K) \cdot \bar{\tau}(K)$  est donc une forêt corégulière.

Il reste à démontrer, pour obtenir le résultat voulu, que  $\tau(K)$  est aussi une forêt corégulière. Or, par construction de  $\bar{\tau}$  on a :

$$\text{si } t \in \Omega^* - \Omega^*(\#) \text{ alors } (q, \alpha) \xrightarrow[\tau]{} t \text{ ssi } (q, \alpha) \xrightarrow[\bar{\tau}]{} t.$$

On a donc

$$\tau(K) \cap (\Omega^* - \Omega^*(\#)) = \bar{\tau}(K) \cap (\Omega^* - \Omega^*(\#))$$

or  $\tau(K) \subset \Omega^* - \Omega^*(\#)$  d'où  $\tau(K) = \bar{\tau}(K) \cap (\Omega^* - \Omega^*(\#))$ .

Or  $\Omega^* - \Omega^*(\#)$  est évidemment une forêt reconnaissable;  $\bar{\tau}(K)$  est une forêt corégulière. On va démontrer ci-dessous (Proposition 9) que l'intersection d'une forêt reconnaissable et d'une forêt corégulière est encore une forêt corégulière, d'où  $\tau(K)$  est une forêt corégulière.

*cqfd.*

### Proposition 9

*La classe des forêts corégulières est fermée par intersection avec une forêt reconnaissable.*

*En effet.*

Soit  $G$  une grammaire corégulière qui engendre  $F(G) \subset \Sigma_T^*$  et  $H \subset \Sigma_T^*$  une forêt reconnue par l'automate déterministe  $\langle A, A_F, \alpha \rangle$ .

On peut étendre homomorphiquement  $\alpha$  en une application qui à

$$t \in \Sigma_T^*(x_1, \dots, x_n)$$

fait correspondre une application  $\alpha_t : A^n \rightarrow A$  qui vérifie

$$\forall t_1, \dots, t_n \in \Sigma_T^*, \|t(t_1, \dots, t_n)\| = \alpha_t(\|t_1\|, \dots, \|t_n\|)$$

(Il n'est pas nécessaire que toutes les variables  $x_i$  figurent dans  $t$ .)

On va construire une grammaire corégulière  $G'$  qui engendre  $F(G) \cap H$ .

Le vocabulaire terminal de  $G'$  est celui de  $G : \Sigma_T$ .

Le vocabulaire non-terminal de  $G'$  est :  $\bigcup_{\sigma \in \Sigma_N} \{\sigma\} \times A^{d(\sigma)}$  où  $\Sigma_N$  est le vocabulaire non-terminal de  $G$  (on peut supposer que  $X_0$ , l'axiome de  $G$ , est le seul symbole de degré 0 de  $\Sigma_N$ ).

L'axiome de  $G'$  est  $(X_0, A^0) = X_0$ .

Les règles de  $G'$  sont définies par :

– si  $\langle X_0; \sigma(t_1, \dots, t_n) \rangle$  est une règle non-terminale de  $G$  alors

$$\langle X_0; (\sigma, a_{i_1}, \dots, a_{i_n})(t_1, \dots, t_n) \rangle$$

est une règle de  $G'$  avec  $a_{i_j} = \alpha_{t_j} = \|t_j\|$

– si  $\langle \sigma'(x_1, \dots, x_p); \sigma(t_1, \dots, t_n) \rangle$  est une règle non-terminale de  $G$  alors pour tout  $(a_{i_1}, \dots, a_{i_p}) \in A^p$

$$\langle (\sigma', a_{i_1}, \dots, a_{i_p})(x_1, \dots, x_p); (\sigma, a_{j_1}, \dots, a_{j_n})(t_1, \dots, t_n) \rangle$$

est une règle de  $G'$  où  $a_{j_k} = \alpha_{t_k}(a_{i_1}, \dots, a_{i_p})$

– si  $\langle X_0; t \rangle$  est une règle terminale de  $G$  alors  $\langle X_0; t \rangle$  est une règle de  $G'$  si et seulement si  $\alpha_t = \|t\| \in A_F$

– si  $\langle \sigma'(x_1, \dots, x_p); t \rangle$  est une règle terminale de  $G$  alors pour tout  $(a_{i_1}, \dots, a_{i_p}) \in A^p$ ,

$\langle (\sigma', a_{i_1}, \dots, a_{i_p})(x_1, \dots, x_p); t \rangle$  est une règle de  $G'$  si et seulement si  $\alpha_t(a_{i_1}, \dots, a_{i_p}) \in A_F$ .

Il est immédiat par construction que :

a)  $X_0 \xrightarrow{*}_G \sigma(t_1, \dots, t_n)$  entraîne  $X_0 \xrightarrow{*}_{G'} (\sigma, \|t_1\|, \dots, \|t_n\|)(t_1, \dots, t_n)$

b)  $(\sigma', a_{i_1}, \dots, a_{i_p})(t_1, \dots, t_p) \xrightarrow{*}_{G'} (\sigma, a_{j_1}, \dots, a_{j_n})(t'_1, \dots, t'_n)$

entraîne  $\sigma'(t_1, \dots, t_p) \xrightarrow{*}_G \sigma(t'_1, \dots, t'_n)$

et  $X_0 \xrightarrow{*}_{G'} (\sigma, a_{i_1}, \dots, a_{i_n})(t_1, \dots, t_n)$  entraîne  $X_0 \xrightarrow{*}_G \sigma(t_1, \dots, t_n)$

d'où  $X_0 \xrightarrow{*}_G \sigma(t_1, \dots, t_n)$  ssi

$$X_0 \xrightarrow{*}_{G'} (\sigma, \|t_1\|, \dots, \|t_n\|)(t_1, \dots, t_n)$$

Par ailleurs d'après la définition des règles terminales de  $G'$

$$(\sigma, \|t_1\|, \dots, \|t_n\|)(t_1, \dots, t_n) \xRightarrow{G'} t \quad \text{ssi} \quad \sigma(t_1, \dots, t_n) \xRightarrow{G} t$$

et  $\|t\| \in A_F$ .

On a donc bien  $F(G') = F(G) \cap H$ .

*cqfd.*

Dans [1] sont définis des transducteurs « ascendants » qui, à l'opposé des transducteurs « descendants » définis ci-dessus, transduisent les arbres en les parcourant des feuilles à la racine (1).

En utilisant une construction analogue à celle utilisée pour démontrer la proposition 9, on démontre :

**Proposition 10**

$M'_1$  est fermée par transduction ascendante déterministe.

Par contre  $M'_1$  n'est pas fermée par transduction descendante quelconque. Sinon on aurait, puisque  $M_0$  est incluse dans  $M'_1$  :  $NT(M_0) \subset NT(M'_1) \subset M'_1$ . Or  $NT(M_0) = M_1$  qui n'est pas incluse dans  $M'_1$ .

$M'_1$  n'est pas non plus fermée par transduction ascendante quelconque car alors, puisqu'une transduction descendante est le composé de deux transductions ascendantes [1],  $M'_1$  serait fermée par transduction descendante.

**Proposition 11**

Si  $\varphi$  est un homomorphisme de monoïde et si  $L$  appartient à la classe EDTOL-strict,  $\varphi(L) - \{\Lambda\}$  appartient à la classe EDTOL-strict.

*En effet.*

On sait [1] que si  $F$  est une forêt de  $\Sigma^*$  et  $\varphi$  un homomorphisme de monoïde défini sur  $\Sigma_0^*$ , il existe un transducteur ascendant déterministe  $\tau$  tel que

$\Phi(\tau(F)) = \varphi(\Phi(F)) - \{\Lambda\}$ . Le résultat découle alors des théorèmes 2 et 3 et de la proposition 10.

*cqfd.*

**Proposition 12**

La classe EDTOL-strict est identique à la classe des langages de EDTOL ne contenant pas le mot vide.

*En effet.*

Il est clair que tout langage de EDTOL-strict est un langage de EDTOL ne contenant pas le mot vide. Réciproquement si le langage  $L$  est engendré

---

(1) Dans l'univers des objets mathématiques, les arbres ont les feuilles en bas et la racine en haut!

par une grammaire ETOL déterministe  $G = \langle V, T, \mathcal{F}, \sigma \rangle$ , on construit une grammaire ETOL déterministe stricte  $G'$  en remplaçant toute règle  $X \rightarrow \Lambda$  figurant dans une table de  $\mathcal{F}$  par la règle  $X \rightarrow a$  où  $a$  n'appartient pas à  $T$ .

Le langage  $L' = L(G')$  appartient donc à EDTOL-strict. Soit  $\varphi$  l'homomorphisme défini par  $\varphi(x) = x$ , si  $x \in T$  et  $\varphi(a) = \Lambda$ . On a bien  $L = \varphi(L')$ . Si le mot vide n'appartient pas à  $L$ ,  $L = \varphi(L') - \{ \Lambda \}$  qui appartient à EDTOL-strict d'après la proposition 11.

*cqfd.*

Cette proposition a également été démontrée par M. Latteux [6].

### ANNEXE

*Démonstration de P 1 :*

– Si  $\gamma$  est de longueur 1 alors  $\gamma$  est formé d'une seule table  $P_i$

\* Si  $A \xRightarrow{\gamma} w = z_1 \dots z_n$  c'est que la règle terminale  $A \rightarrow w$  appartient à  $P_i$ , et donc que le transducteur contient la production  $\langle q_A; P_i; A(z_1, \dots, z_n) \rangle$  d'où  $(q_A, P_i) \vdash_{\tau}^* A(z_1, \dots, z_n)$  et  $\Phi(A(z_1, \dots, z_n)) = z_1 \dots z_n = w$ .

\* Réciproquement si  $(q_A, P_i) \vdash_{\tau}^* t$  avec  $\Phi(t) = z_1 \dots z_n \in T^*$ , alors, par construction de  $\tau$ ,  $t$  est nécessairement égal à  $A(z_1, \dots, z_n)$  avec  $A \rightarrow z_1 \dots z_n \in P_i$  d'où  $A \xRightarrow{P_i} z_1 \dots z_n$ .

– Si  $\gamma = P_i \gamma'$  et si P1 est vraie pour  $\gamma'$

Si  $A \xRightarrow{\gamma} w$  alors il existe  $A \rightarrow u_0 A_{i_1} \dots A_{i_n} u_n \in P_i$  et  $w_1, \dots, w_n \in T^*$  tels que pour tout  $j$ ,  $A_{i_j} \xRightarrow{\gamma'} w_j$  et tels que  $w = u_0 w_1 \dots w_n u_n$ .

D'après (P1), pour tout  $j$ , il existe  $t_j$  tel que  $(q_{A_{i_j}}, \gamma') \vdash_{\tau}^* t_j$  et  $\Phi(t_j) = w_j$ .

D'autre part le transducteur contient une production  $\langle q_A; P_i; t \rangle$  avec  $t = t(q_{A_{i_1}}, \dots, q_{A_{i_n}})$  et  $\Phi(t) = u_0 q_{A_{i_1}} \dots q_{A_{i_n}} u_n$ .

Donc  $(q_A, P_i \gamma') \vdash_{\tau}^* t(t_1, \dots, t_n)$  et  $\Phi(t(t_1, \dots, t_n)) = u_0 \Phi(t_1) \dots \Phi(t_n) u_n = w$

\* Réciproquement si  $(q_A, P_i \gamma') \vdash_{\tau}^* t$  alors il existe une production

$$\langle q_A; P_i; t'(q_{A_{i_1}}, \dots, q_{A_{i_n}}) \rangle$$

du transducteur et des arbres  $t_1, \dots, t_n$  tels que  $\forall j \in [1, n] (q_{A_{i_j}}, \gamma') \vdash_{\tau}^* t_j$  et  $t = t'(t_1, \dots, t_n)$ .

De plus si  $\Phi(t') = u_0 q_{A_{i_1}} \dots q_{A_{i_n}} u_n$  alors  $A \rightarrow u_0 A_{i_1} \dots A_{i_n} u_n \in P_i$ .



En appliquant P1 on obtient que  $A_{ij} \xRightarrow{\gamma'} \Phi(t_j)$  pour tout  $j$  d'où  
 $A \xRightarrow{p_{i\gamma'}} u_0 \Phi(t_1) \dots \Phi(t_n) u_n$ .

Or  $\Phi(t) = u_0 \Phi(t_1) \dots \Phi(t_n) u_n$ .

*cqfd.*

#### REMERCIEMENTS

Nous tenons à remercier notre collègue Michél Latteux, dont les recherches sur les systèmes de Lindenmayer sont à l'origine de ce travail, pour sa précieuse contribution.

#### REFERENCES

- [1] BAKER, B. S., *Tree transductions and families of tree languages*. 5<sup>th</sup> ACM Symposium on theory of Computing (1973).
- [2] BRAINERD, W. S., *Tree-generating regular system*. Information and Control, 14 (1969), 218-231.
- [3] DOWNEY, P., *Formal languages and recursion schemas*, ph. D. dissertation Harvard University (1974).
- [4] DOWNEY, P., *Tree transducers and ETOL Tree Systems*. Conference on formal languages, automata and development. Noordwijkerhout, The Netherlands (avril 1975).
- [5] HERMAN, G. T. et ROZENBERG, G., *Developmental Systems and Languages*. North-Holland P. C. (1975).
- [6] LATTEUX, M., *Langages simultanés*. Publication du laboratoire de calcul de Lille, n° 46 (1974).
- [7] LUCKHAM, D. C., PARK, D. M. et PATERSON, M. S. *On formalized computer programs*. J. Comput. System Sci., 4 (1970), 220-249.
- [8] MEZEI, J. et WRIGHT, J. B., *Algebraic automata and context-free sets*. Math Systems Theory, 4 (1970), 257-287.
- [9] ROUNDS, W. C., *Mappings and grammars on trees*. Math Systems Theory, 4 (1970), 257-287.
- [10] ROZENBERG, G., *Extensions of tabled OL-Systems and Languages*. Internat. J. Comput. Information Sci, 2 (1973), 311-336.
- [11] THATCHER, J. W., *Generalized<sup>2</sup> sequential machine maps*. J. Comput. System Sci., 4 (1970), 339-367.