

50376
1981
15
N° d'ordre : 847

50376
1981
15

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

**DOCTEUR DE 3^{ème} CYCLE
INFORMATIQUE**

par

Jocelyne MONGY-STEEN



**TRANSFORMATIONS DE NOYAUX
RECONNAISSABLES D'ARBRES
FORETS RATEG**

Soutenue le 19 février 1981 devant la Commission d'Examen

Membres du Jury	MM. G. JACOB	Président
	M. DAUCHET	Rapporteur
	L. BOASSON	Examineur
	C. CARREZ	Examineur
	M. GROSS	Examineur

Gérard JACOB, Professeur à l'Université de Lille I, a bien voulu témoigner de l'intérêt qu'il accorde à cette thèse en acceptant la présidence du Jury. Je tiens à lui exprimer toute ma reconnaissance.

Luc BOASSON, Professeur à l'Université de Paris VII, me fait l'honneur de bien vouloir juger les résultats d'un travail dont il a pris intérêt à suivre les différentes étapes. J'ai le plaisir de l'assurer ici de mes plus vifs remerciements.

Christian CARREZ, Professeur à l'Université de Lille I, me fait l'honneur de faire partie de ce jury. Je l'en remercie très vivement.

Monsieur GROSS, Professeur à l'Université de Paris VII, me fait l'honneur de bien vouloir s'intéresser à ma recherche et de participer au jury de cette thèse. Qu'il veuille croire en l'expression de mes plus profonds remerciements.

Max DAUCHET a été à l'origine de ce travail. Ses nombreux conseils, ses fréquentes interventions et ses remarques toujours fécondes, ont été les guides indispensables à l'orientation et à l'aboutissement de ma recherche. Il a toujours su me témoigner disponibilité et enthousiasme et a bien voulu être le rapporteur de cette thèse. Qu'il soit ici assuré de ma plus profonde gratitude.

Je remercie tous les membres de l'équipe d'informatique théorique pour l'ambiance sympathique d'un cadre que je n'ai pas toujours pu fréquemment cotoyer.

Je remercie Madame Patricia CARON, qui s'est chargée de la mise en pages de cette thèse avec gentillesse, diligence et adresse.

Je remercie Madame et Monsieur DEBOCK qui en ont assuré la réalisation matérielle avec beaucoup de compétence et de rapidité.

A mes parents

A François

A Denis pour son rôle de "Candide".

TABLE DES MATIÈRES

INTRODUCTION

page n°

* * *

CHAPITRE 0 : PRÉLIMINAIRES

18

A - Les arbres

26

B - Les magmoïdes

30

1) Définitions

30

2) Le magmoïde des torsions

31

3) Le magmoïde $T(\Sigma)$

32

4) Le magmoïde $\hat{T}(\Sigma)$

34

C - Les forêts

36

1) Définitions

36

2) Grammaires

38

D - Les morphismes

40

1) Morphismes de magmoïde

40

2) Morphismes d'arbres

40

E - Les bimorphismes

44

Définition (restreinte)

44

* * *

CHAPITRE I : GÉNÉRATION DE FORÊTS RÉCURSIVEMENT ÉNUMÉRABLES À PARTIR DE FORÊTS RECONNAISSABLES

45

Énoncé des résultats et démonstration

* * *

CHAPITRE II : LES FORÊTS RATEG

72

A - Préliminaires algébriques

73

I - L'ensemble ordonné $T(\Sigma)_{/\equiv}$

74

1) Le préordre $<$ dans $T(\Sigma)$

2) L'équivalence \equiv . L'ordre \leq dans $T(\Sigma)_{/\equiv}$

75

3) Propriétés de \leq et \equiv

80

4) Bornes dans $T(\Sigma)_{/\equiv}$

83

II - Algorithme d'unification	page 88
1) Forme générale de l'algorithme	88
2) Définition des opérateurs b f et γ	90
3) Propriétés de b f et γ	92
4) Exemples d'unification	96
B - Les Forêts RATEG	102
I - Introduction aux RATEG	102
II - Définitions formelles	105
1) Les $\bar{\text{RATEG}}$	105
2) Les RATEG	107
3) Les RATEG généralisés	111
III - Deux exemples complets de forêts RATEG	112
IV - Propriétés des RATEG	117
Enoncés et démonstrations	
* * *	
CHAPITRE III : LE CAS DES RATEG HOMOGÈNES	148
1) Définitions	151
2) Propriétés	152
Enoncés et démonstrations	
3) Exemple de forêt RATEG homogène	164
4) Cas des morphismes complets homogènes	168
* * *	

BIBLIOGRAPHIE

* * *

*

INTRODUCTION

Tout comme les langages reconnaissables, les forêts reconnaissables jouent un rôle fondamental en informatique.

Nous étudions ici l'effet d'opérations simples telles que homomorphismes, union ou intersection sur la classe des forêts reconnaissables, ce qui nous amène entre autres à définir une nouvelle famille d'automates : les Reconnaisseurs Avec Test d'EGalité (en abrégé RATEG).

Nous prenons pour cadre à notre étude la théorie des magmoïdes (Arnold et Dauchet [AD1, AD2]).

Pour mieux situer notre travail, présentons d'abord brièvement les notions de base que nous utilisons en en mentionnant les principales motivations, puis l'essentiel de nos résultats.

Un arbre peut être considéré comme une description d'une composition formelle d'opérateurs. Si on interprète ces opérateurs comme :

- opérateurs arithmétiques et de contrôle, on obtient les schémas de calculs et les arbres programmatiques,
- opérateurs grammaticaux, on obtient les arbres linguistiques,
- "constructeurs de types", on obtient des structures d'objets.

Une forêt étant un ensemble d'arbres, l'étude des propriétés des arbres mène à l'étude des classes de forêts.

La classe des forêts reconnaissables a été la plus étudiée.

De fait elle est essentielle dans bien des domaines :

- tout schéma de calcul itératif peut être développé en une forêt reconnaissable (Cousineau [COUS]),
- l'ensemble des arbres de dérivation d'une grammaire algébrique est une forêt reconnaissable (Pair et Quéré [PAQ], Rounds [RL], Thatcher [TH1]).
- l'ensemble des composés d'opérateurs typés est une forêt reconnaissable.

La notion de transformation d'arbres est apparue dès lors qu'il s'agissait de transformer des phrases - naturelles ou informatiques (des programmes) - en respectant leur structure syntaxique :

- c'est le cas en linguistique transformationnelle (Gross et Lentin [GL]), où l'on étend un noyau de phrases (en général un langage algébrique) par composition de transformations simples guidées par la syntaxe (Ginsburg et Partee [GP], Peters et Richie [PR]). En termes d'arbres cela revient donc à transformer des forêts reconnaissables ;

- de même en compilation sont nés les "syntax-directed translators" (Ahd et Ullman [AU1, AU2], Lohro [LO], Schreiber [SC]) puis les transducteurs d'arbres. Ces transducteurs sont d'abord apparus du point de vue algorithmique comme des reconnaisseurs d'états finis avec sortie (Bjorner [BJ], Engelfriet [EN1], Lilin [LI], Rounds [R1], Thatcher [TH3]) puis leur formalisation algébrique s'est effectuée en termes d'homomorphismes et de bimorphismes. (Arnold et Dauchet [AD5], [AD6], Baker [BA], Dauchet [D], Engelfriet [EN2], Takahashi [TA]).

Rappelons qu'un homomorphisme d'arbres (ou morphisme) est une application compatible avec les opérations sur les arbres c'est à dire, pour nous, compatible avec la structure de magmaïde. Si Σ et Δ sont deux alphabets gradués nous appelons $T^1(\Sigma)$ et $T^1(\Delta)$ les ensembles des arbres construits sur Σ et Δ .

Un homomorphisme ϕ transformant des arbres de $T^1(\Sigma)$ en arbres de $T^1(\Delta)$ est alors totalement défini par sa donnée sur Σ et s'étend de façon compatible à $T^1(\Sigma)$. Soit par exemple les deux alphabets gradués

$$\Sigma = \{\bar{f}(x_1, x_2, x_3), a(x_1), b(x_1), \bar{a}\}$$

$$\text{et } \Delta = \{+(x_1, x_2), \times(x_1, x_2), \alpha(x_1), \bar{\alpha}\},$$

où les variables x_1, x_2, x_3 seront dans un arbre remplacées en chacune de leurs occurrences par t_1, t_2, t_3 .

Et soit ϕ tel que :

$$\phi\left(\begin{array}{c} \bar{x} \\ / \quad | \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array}\right) = \begin{array}{c} + \\ / \quad \backslash \\ x \quad x \\ / \quad \backslash \quad / \quad \backslash \\ x_1 \quad x_2 \quad x_1 \quad x_3 \end{array}$$

$$\phi\left(\begin{array}{c} a \\ | \\ x_1 \end{array}\right) = \begin{array}{c} \alpha \\ | \\ x_1 \end{array}$$

$$\alpha\left(\begin{array}{c} b \\ | \\ x_1 \end{array}\right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ x_1 \quad \alpha \\ | \\ x_1 \end{array}$$

$$\phi(\bar{a}) = \bar{\alpha}$$

Pour $t = \begin{array}{c} \bar{x} \\ / \quad | \quad \backslash \\ a \quad a \quad b \\ | \quad | \\ a \quad a \end{array}$

on obtient alors

$$\phi(t) = \begin{array}{c} + \\ / \quad \backslash \\ x \quad x \\ / \quad \backslash \quad / \quad \backslash \\ \alpha \quad \bar{\alpha} \quad \alpha \quad \alpha \\ | \quad | \quad | \quad \backslash \\ \alpha \quad \alpha \quad \alpha \quad \alpha \\ | \\ \alpha \end{array}$$

Un morphisme ϕ est

- linéaire ssi pour toute lettre a de Σ , chaque variable de a apparaît au plus une fois dans $\phi(a)$;
- complet ssi, pour tout $a \in \Sigma$, chaque variable de a apparaît au moins une fois dans $\phi(a)$;
- strict ssi, pour tout $a \in \Sigma$, $\phi(a)$ contient au moins une lettre de Δ ;
- un démarquage ssi ϕ est linéaire et tel que $\phi(a)$ au plus une lettre de Δ ;
- un démarquage propre ssi ϕ ne transforme que les labels.

Illustrons ces différents cas en prenant ϕ défini par l'identité sur

$\Sigma - \{a\}$ et sur a par :

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ x_1 \quad x_2 \quad \beta \\ | \\ x_3 \end{array} ; \phi \text{ est linéaire, complet, strict.}$$

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ x_1 \quad \beta \\ / \quad \backslash \\ x_1 \quad x_2 \quad \alpha \\ | \\ x_3 \end{array} ; \phi \text{ est complet, strict, non linéaire.}$$

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ x_1 \quad \beta \\ / \quad \backslash \\ x_1 \quad x_2 \end{array} ; \phi \text{ est strict, non linéaire, non complet.}$$

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = \begin{array}{c} \alpha \\ | \\ x_1 \end{array} ; \phi \text{ est un démarquage non complet.}$$

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} ; \phi \text{ est un démarquage propre.}$$

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ x_2 \quad x_1 \quad x_3 \end{array} ; \phi \text{ est un démarquage non propre.}$$

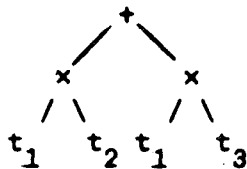
$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array} \right) = x_3 ; \phi \text{ est non strict.}$$

Un bimorphisme $B = (\phi, K, \Psi)$ est le composé d'un morphisme inverse ϕ , d'une intersection avec une forêt reconnaissable K et d'un morphisme Ψ ; pour tout arbre t la transformation associée est définie par $B(t) = \Psi[\phi^{-1}(t) \cap K]$.
 Tout bimorphisme s'interprète comme la succession d'une phase d'analyse (l'appartenance à $\phi^{-1}(t) \cap K$ revient à reconnaître à t certaines propriétés)

et d'une phase de synthèse (le choix de Ψ peut être un choix d'implémentation de l'image, l'application de Ψ synthétisant en ce sens les propriétés analysées précédemment).

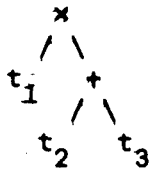
Nous verrons par exemple dans le chapitre 0 le cas de la distributivité de \times par rapport à $+$. Pouvoir appliquer la distributivité à un arbre t revient à vérifier l'existence d'un triplet d'arbres (t_1, t_2, t_3)

tels que $t =$



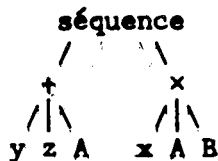
ce qui se traduit à l'analyse par l'existence d'un antécédent dans $\phi^{-1}(t) \cap K$ de la forme $t' = \alpha(t_1, t_2, t_3)$.

Par un certain Ψ on transforme alors t' en



Si on veut traduire les opérations $+$ et \times en termes de procédures (le résultat de l'addition est la valeur de A dans la procédure $+(x, y, A)$) cela revient à choisir Ψ tel que

$\Psi(x, y, z) =$



Rappelons qu'en théorie des langages la même démarche a été faite : l'étude des langages reconnaissables (ou rationnels, ou réguliers selon le point de vue adopté) a mené des transductions aux bimorphismes (Nivat [NI]). En fait les notions de reconnaissabilité rationalité et de bimorphisme peuvent être définis dans des structures tout à fait générales (Eilenberg et Wright [EW]). Dans le cas des langages, on a obtenu de bonnes propriétés

de clôture : clôture des langages reconnaissables par bimorphismes, clôture des bimorphismes par composition, d'où l'étude des cônes de langages (Boasson - Nivat [BN], Latteux [L]). Mais ces résultats ne s'étendent pas au cas des arbres : il est connu que la classe des forêts reconnaissables n'est pas close par morphisme non linéaire.

Etudier la classe de forêts obtenue par clôture de la classe des forêts reconnaissables par union, intersection et morphisme s'impose alors : c'est le centre de notre étude et nous avons entre autres, établi les résultats suivants :

- la clôture par homomorphisme, réunion et intersection de la classe des forêts reconnaissables est la classe des forêts récursivement énumérables. (Conséquence du corollaire I.4 établi au chapitre I).

Nous définissons une nouvelle classe d'automates permettant de tester des égalités de sous-arbres (voir fin de l'introduction et chapitre II).

Nous montrons que

- la classe des forêts RATEG contient la classe des forêts reconnaissables
- la classe des forêts RATEG est incluse dans la classe des forêts récursives
- la classe des forêts RATEG est close par union intersection et morphisme complet.

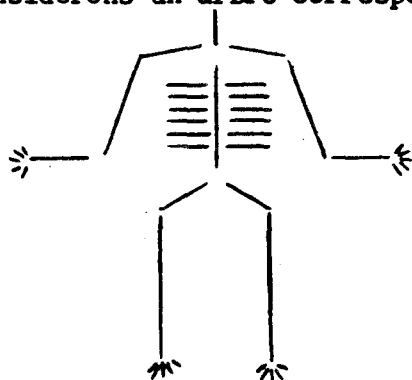
La classe des forêts RATEG contient donc la clôture par union intersection et morphisme complet des forêts reconnaissables.

L'importance de la "complétude" des morphismes, que nous avons ici mise en évidence peut faire l'objet d'un parallèle avec la théorie des langages : en se limitant à des morphismes non effaçants (Book et Greibach [BG] ont défini la classe des "Quasi-Realtime Languages" comme clôture de Alg par

union intersection et morphisme non effaçant. De plus chaque Quasi Realtime Language est l'image par morphisme non effaçant de l'intersection de trois langages algébriques.

Bien que les manipulations et analyses d'images ne soient pas le but de notre travail, nous allons maintenant illustrer de trois exemples figuratifs les différentes classes de forêts que nous introduisons dans la suite, et tout d'abord la classe des images par morphismes de forêts reconnaissables - que nous noterons $\phi(\text{Rec})$ -

Considérons un arbre correspondant à un schéma (très simpliste) du squelette



humain. (On conviendra qu'un os long de n centimètres est représenté par n noeuds d'arbres, ainsi " ——— " de longueur n est approximé graphiquement par " - - - - - " où les tirés représentent les arcs joignant les n noeuds. Ceci permet de représenter

avec un alphabet fini l'infinie diversité des mensurations ...)

Vérifier que le fémur précède l'ensemble tibia-péroné est du domaine du reconnaissable : il suffit d'imposer le bon enchaînement des états correspondants à la "lecture" de chaque membre, c'est le principe de reconnaissance par mémoire finie des automates d'états finis d'arbres. Par contre vérifier l'égalité des deux jambes ou des côtes n'est plus de ce domaine - car les membres peuvent être arbitrairement longs - mais du domaine de $\phi(\text{Rec})$: il s'agit ici de réaliser l'égalité de deux sous-arbres arbitrairement longs, ce que l'on peut obtenir par morphisme non linéaire, c'est à dire générant des égalités. Ceci montre bien comment la classe Rec des forêts reconnaissables n'est pas close en général par morphisme (mais Rec est close par morphismes linéaires).

Voyons maintenant sur un deuxième exemple (inspiré du classique calcul de $n!$)

comment les arbres des forêts RATEG peuvent contenir un nombre quelconque des sous-arbres égaux situés à des profondeurs toutes différentes (contrairement au cas des $\phi(\text{Rec})$).

La résolution par point fixe de l'équation

$$F(x) = x(a(x), F(a(x))) \text{ ou } \bar{a} \text{ donnera}$$

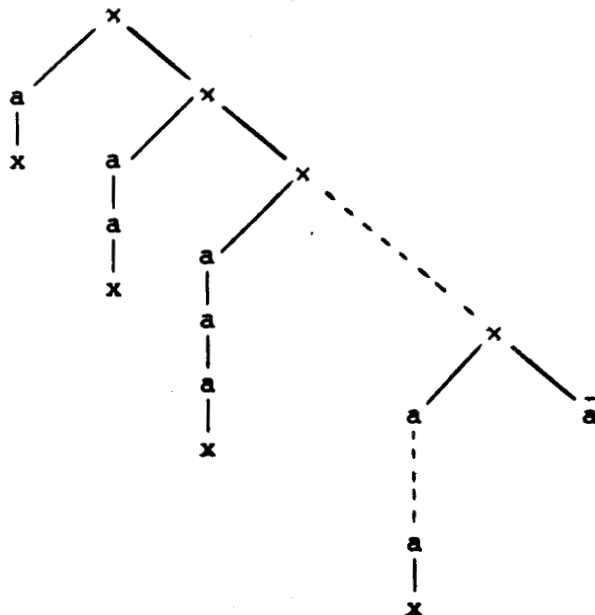
$$F_0(x) = \Omega \text{ (l'indéfini)}$$

$$F_1(x) = \bar{a} \text{ ou } \Omega$$

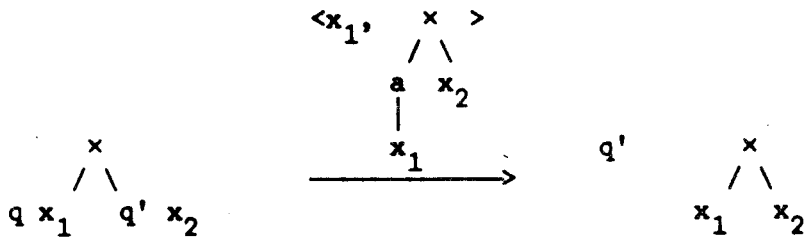
$$F_2(x) = \begin{array}{c} x \\ / \quad \backslash \\ a \quad \bar{a} \\ | \\ x \end{array} \text{ ou } \bar{a} \text{ ou } \Omega$$

$$F_3(x) = \begin{array}{c} x \\ / \quad \backslash \\ a \quad x \\ | \quad / \quad \backslash \\ x \quad a \quad \bar{a} \\ | \\ a \\ | \\ x \end{array} \text{ ou } \begin{array}{c} x \\ / \quad \backslash \\ a \quad \bar{a} \\ | \\ x \end{array} \text{ ou } \bar{a} \text{ ou } \Omega$$

et de $F_{n+1}(x)$ on déduit l'arbre



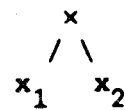
Pour vérifier alors que la sous-branche gauche d'un "x" quelconque contient un "a" de moins que celle du "x" successeur immédiat de droite on appliquera ce qui sera une règle de RATEG de la forme



L'application de cette règle peut être comprise comme suit :

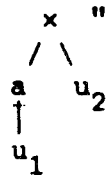
On transite du couple d'états $\langle q, q' \rangle$ à q' en lisant

si et seulement si, dans l'arbre lu,



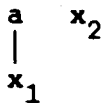
l'arbre t_1 substitué à x_1 et l'arbre t_2 substitué à x_2 vérifient

" $\exists u_1$ et u_2 tels que $t_1 = u_1$ et $t_2 =$



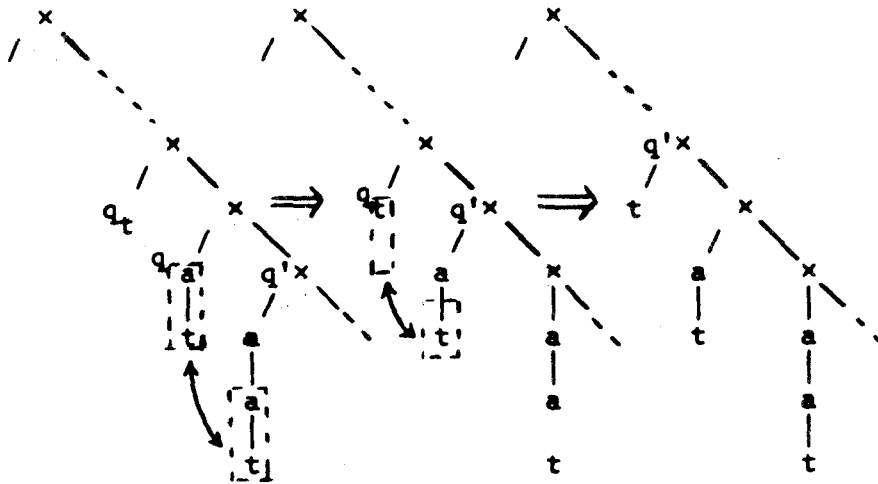
Cette dernière contrainte étant la traduction du membre de contrôle

$\langle x_1, \begin{array}{c} x \\ / \quad \backslash \\ a \quad x_2 \\ | \\ x_1 \end{array} \rangle$ où u_1 est substitué à x_1 et u_2 à x_2 .



Deux applications consécutives d'une telle règle donneront la

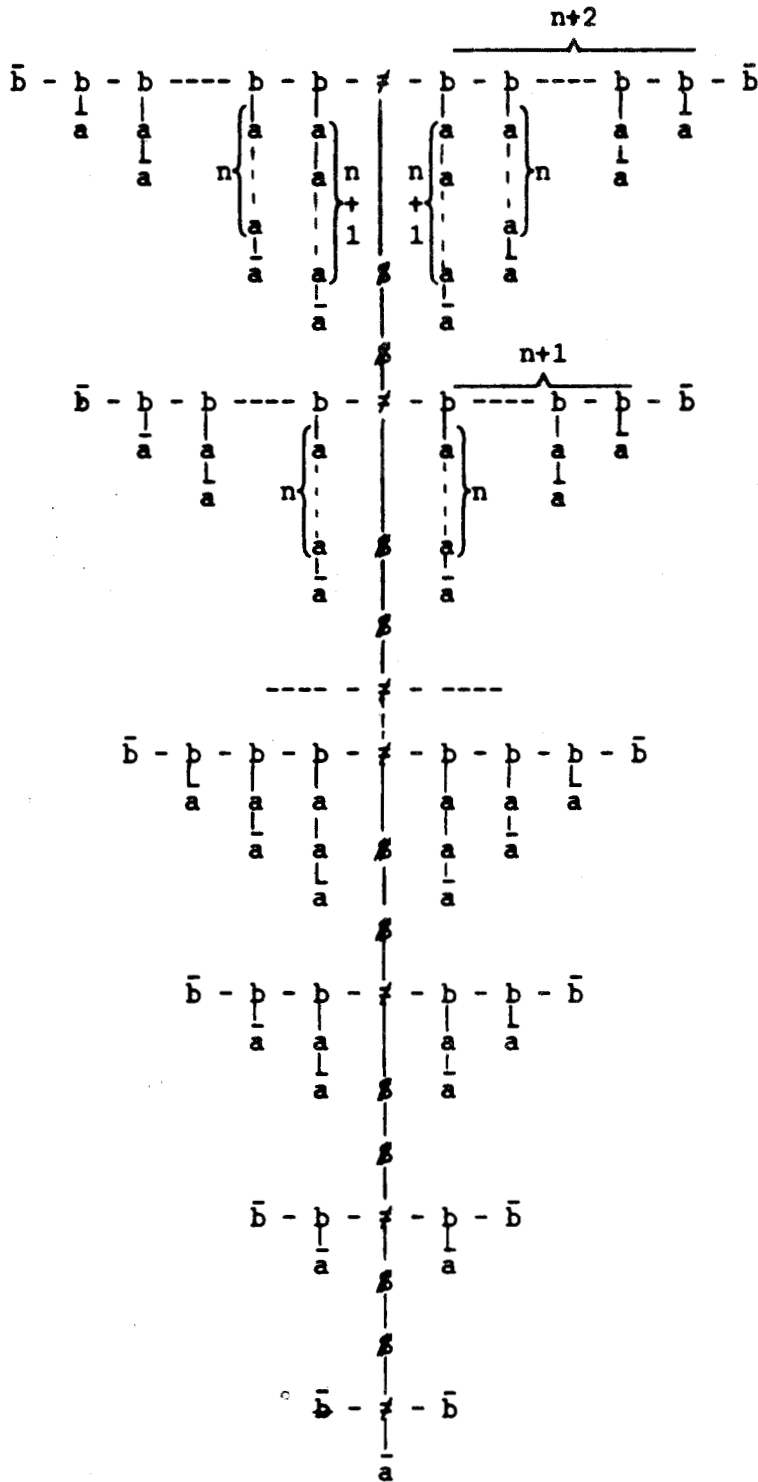
séquence =



et par conséquent on testera la présence d'un même t tout au long de l'arbre. (Le lecteur trouvera au chapitre II § III, un exemple totalement développé dérivé de celui-ci).

On remarquera qu'on en est amené à manipuler des suites finies d'arbres, et c'est ici que le choix de la structure de magmaïde prend son intérêt (voir chapitre 0).

Considérons maintenant le 3ème exemple plus complexe de la "fougère" que l'on représentera par un arbre de la forme suivante :



Il est facile de constater que sur chaque branche horizontale de "b" on passe d'un "b" au "b" consécutif en ajoutant ou en enlevant un "a" à la branche verticale de "b". Quant aux branches de "b" (à distance finie ici de deux "b") on constate qu'elles ne diffèrent que d'un "b" - le fait d'avoir un "a" de plus sur la branche de ce "b" étant lié aux conditions sur les "b" et non sur les "b". On peut alors se donner les règles suivantes

(en considérant des règles symétriques pour les branches droite et gauche des "≠"), règles qui sont de la forme de celles des reconnaisseurs classiques pour les 7 premières et qui imposent pour les trois dernières des structures particulières avec contraintes d'égalité :

$$\bar{a} \rightarrow q_0 \bar{a}$$

$$\bar{b} \rightarrow q_1 \bar{b}$$

$$\begin{array}{c} a \\ | \\ q_0 x_1 \end{array} \rightarrow q_2 \begin{array}{c} a \\ | \\ x_1 \end{array}$$

$$\begin{array}{c} b \\ / \quad \backslash \\ q_0 x_1 \quad q_1 x_2 \end{array} \rightarrow q_3 \begin{array}{c} b \\ / \quad \backslash \\ x_1 \quad x_2 \end{array} \quad \text{et} \quad \begin{array}{c} b \\ / \quad \backslash \\ q_1 x_1 \quad q_0 x_2 \end{array} \rightarrow q'_3 \begin{array}{c} b \\ / \quad \backslash \\ x_1 \quad x_2 \end{array}$$

$$\begin{array}{c} \neq \\ / \quad | \quad \backslash \\ q_1 x_1 \quad q_0 x_2 \quad q_1 x_3 \end{array} \rightarrow q_4 \begin{array}{c} \neq \\ / \quad | \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array}$$

$$\begin{array}{c} \neq \\ | \\ q_4 x_1 \end{array} \rightarrow q_4 \begin{array}{c} \neq \\ | \\ x_1 \end{array}$$

$$\begin{array}{c} b \\ / \quad \backslash \\ q_2 x_1 \quad q_3 x_2 \end{array} \xrightarrow{\begin{array}{c} a, \quad b \\ | \quad / \quad \backslash \\ y_1 \quad y_1 \quad y_2 \end{array}} q_3 \begin{array}{c} b \\ / \quad \backslash \\ x_1 \quad x_2 \end{array}$$

$$\begin{array}{c} b \\ / \quad \backslash \\ q'_3 x_2 \quad q_2 x_1 \end{array} \xrightarrow{\begin{array}{c} b, \quad a \\ / \quad \backslash \quad | \\ y_1 \quad y_2 \quad y_2 \end{array}} q'_3 \begin{array}{c} b \\ / \quad \backslash \\ x_1 \quad x_2 \end{array}$$

$$\begin{array}{c} \neq \\ / \quad | \quad \backslash \\ q'_3 x_1 \quad q_4 x_2 \quad q_3 x_2 \end{array} \xrightarrow{\begin{array}{c} \neq \\ | \\ \neq \\ / \quad \backslash \\ y_1 \quad y_2 \quad \neq \\ / \quad | \quad \backslash \\ y_1 \quad y_3 \quad y_4 \end{array}} q_4 \begin{array}{c} \neq \\ / \quad | \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array}$$

Il est alors aisé de générer de tels arbres puisqu'à partir de

$$\begin{array}{c} \bar{b} - \neq - \bar{b} \\ | \\ \bar{a} \end{array}$$

on déduira

$$\begin{array}{c} \bar{b} - b - \neq - b - \bar{b} \\ | \quad | \quad | \\ \bar{a} \quad \neq \quad \bar{a} \\ | \\ \neq \\ | \\ \bar{b} - \neq - \bar{b} \\ | \\ \bar{a} \end{array}$$

puis de façon générale, étant donné un arbre de la forme

$$\begin{array}{c} u' - b - \neq - b - u \\ | \quad | \quad | \\ v \quad w \quad v \end{array}$$

on générera

$$\begin{array}{c} u' - b - b - \neq - b - b - u \\ | \quad | \quad | \quad | \quad | \\ v \quad a \quad \neq \quad a \quad v \\ | \quad | \quad | \\ v \quad \neq \quad v \\ | \\ u' - b - \neq - b - u \\ | \quad | \quad | \\ v \quad w \quad v \end{array}$$

Il est intéressant de noter que la description donnée ici diffère fondamentalement de l'évolution dynamique décrite par les L-systèmes.

La comparaison des forêts RATEG avec les arbres de dérivation des L-systèmes reste à faire.

Le chapitre 0 des préliminaires est essentiellement consacré à des rappels de la théorie des arbres et des forêts. On y trouvera un rappel de toutes les définitions précisant le cadre formel de notre étude.

Dans le chapitre I nous abordons un premier théorème essentiel qui montre comment, à partir de la classe Rec, engendrer toutes les forêts récursivement énumérables :

Pour toute forêt F récursivement énumérable, il existe deux forêts reconnaissables K_1 et K_2 , deux morphismes ϕ_1 et ϕ_2 complets stricts, un démarquage strict π tels que

$$F = \pi(\phi_1(K_1) \cap \phi_2(K_2))$$

En corollaire, toute forêt récursivement énumérable est transformée d'une forêt reconnaissable par composition de deux bimorphismes.

Nous en déduisons également un résultat important :

Etant donnés ϕ_1 et ϕ_2 complets

K_1 et K_2 reconnaissables

il est indécidable de savoir si

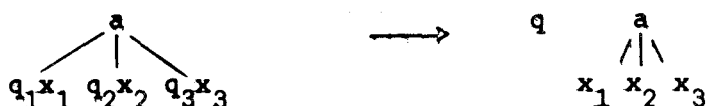
$$\phi_1(K_1) \cap \phi_2(K_2) = \emptyset$$

Le chapitre II se divise en deux parties : dans la première partie nous définissons de façon rigoureuse une relation d'équivalence mettant en évidence que les torsions (n -uples de variables) n'ont d'autre but ici que de préciser des contraintes d'égalité.

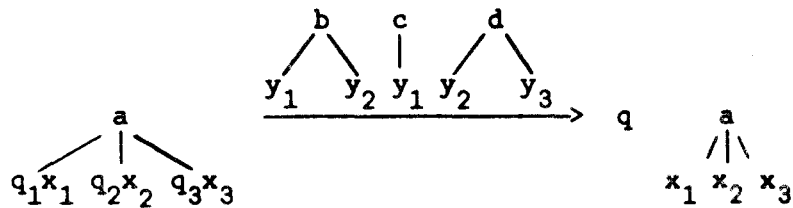
On y redéfinit aussi un algorithme d'unification (Guard [JG], Huet [H], Robinson [R]) qui nous servira à établir la clôture par intersection des forêts RATEG. La 2ème partie consiste en la définition de la nouvelle classe d'automates appelés Reconnaisseurs Avec Tests d'EGalité ou RATEG.

Intuitivement ce sont des reconnaisseurs qui ajoutent à l'automate classique d'états finis d'arbres la possibilité de tester des contraintes d'égalité de sous-arbres.

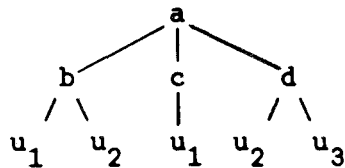
Rappelons en effet qu'une règle de l'automate classique s'écrit par exemple



En imposant aux successeurs de a des contraintes de forme et d'égalité, nous aurons pour les RATEG des règles de la forme



qui s'appliqueront à un arbre de la forme



Nous étudions alors les propriétés de la classe des forêts RATEG (qui sont manifestement récursives). Et entre autres nous démontrons le résultat important suivant :

La classe des forêts RATEG contient la clôture par union intersection et morphisme complet de la classe des forêts reconnaissables.

Le lecteur y verra le rôle essentiel de la "complétude" pour ne pas atteindre la classe des récursivement énumérables (Chapitre I).

Il était alors naturel de chercher à prouver la réciproque. Nous avons distingué le cas des RATEG homogènes : on dit qu'un arbre est h -homogène si chaque branche menant du sommet à une variable est de longueur h et chaque branche menant du sommet à un terminal de longueur $\leq h$. Un RATEG est h -homogène si les arbres figurant dans ses règles sont tous h -homogènes. Une forêt est RATEG h -homogène si elle est reconnue par un RATEG h -homogène.

Dans le chapitre III nous montrons l'inclusion des RATEG homogènes dans la clôture de Rec par union intersection et morphisme complet et donnons même une caractérisation de telles forêts :

Une forêt F est RATEG h -homogène si et seulement si il existe

K et K' forêts reconnaissables

ϕ complet strict

π démarquage propre

tels que

$$F = \pi[\phi(K) \cap \phi(K')]$$

Ce résultat diffère fondamentalement de celui établi au chapitre I concernant les forêts récursivement énumérables par le fait que π est ici un démarquage complet.

Nous conjecturons l'égalité dans le cas général de la classe RATEG et de la clôture de la classe des forêts reconnaissables par homomorphismes complets, union et intersection.

CHAPITRE 0

PRELIMINAIRES

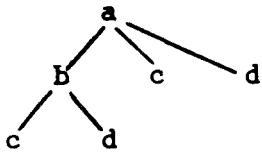
Présentation intuitive et informelle

Nous présentons ici l'idée intuitive des objets de notre étude et introduisons informellement les notions mathématiques de façon que le lecteur puisse appréhender globalement notre travail.

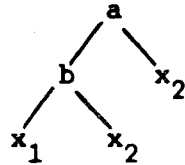
Un arbre peut être représenté par un graphe :

- les noeuds sont labellés par des éléments (lettres) d'un ensemble Σ (alphabet gradué) ;
- le nombre de leurs successeurs (degré) est donné ;
- les branches sont terminées par des lettres (de degré 0) ou des variables indicées.

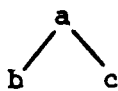
Soit par exemple



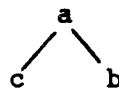
ou



Un arbre est ordonné ; on distinguera en effet



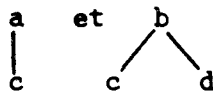
et



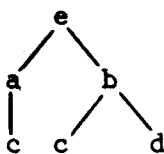
Deux arbres sont dits égaux si et seulement si ils sont superposables.

L'exemple suivant introduit aisément la notion de greffe :

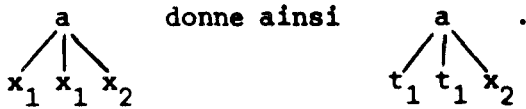
soit une lettre e de degré 2 et 2 arbres a et b ; on peut alors



obtenir



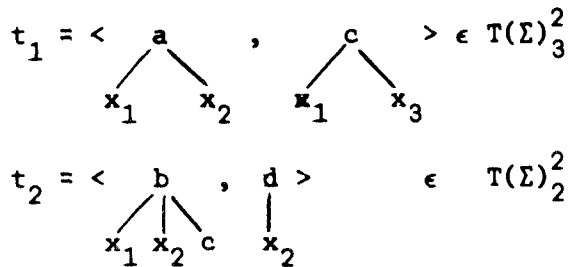
La substitution formalise la greffe : elle remplace chaque occurrence d'une même variable par un même arbre. La substitution de t_1 à x_1 dans



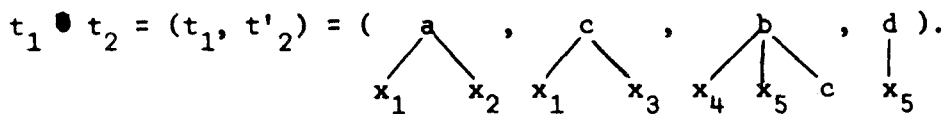
On étend l'appellation d'arbres aux n-uples d'arbres (t_1, t_2, \dots, t_n) . La notation $T(\Sigma)_q^p$ - voir § 0.B.3 - correspond ainsi à l'ensemble des p-uples d'arbres sur Σ indicés par des variables choisies dans $\{x_1, x_2, \dots, x_q\}$.

Le produit tensoriel - noté \otimes - est alors défini sur des arbres : à n arbres t_1, t_2, \dots, t_n il associe le n-uple $\langle t_1, t'_2, \dots, t'_n \rangle$ en décalant les indices de sorte que les t'_i n'aient pas de variables communes.

Soit par exemple

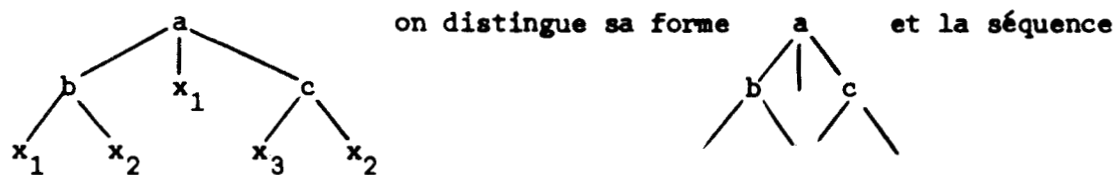


alors $t_1 \otimes t_2$ est le 4-uple défini par



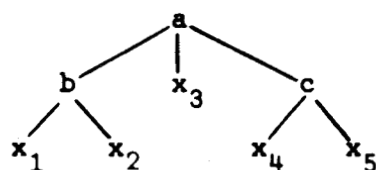
Le produit de composition - noté \cdot - qui formalise la substitution, et le produit tensoriel \otimes sont les deux opérations binaires définies dans le cadre du **magmoïde** $T(\Sigma)$, ensemble des séquences finies d'arbres sur Σ - le magmoïde est la structure algébrique de notre étude et assure une bonne formalisation algébrique des arbres. (Voir § 0.B.3) -

Parmi les éléments de $T(\Sigma)$ les p-uples d'arbres réduits à des variables jouent un rôle particulier : intuitivement si l'on considère un arbre t indicé tel que



$(x_1 x_2 x_1 x_3 x_2)$ des variables qui terminent ses branches.

On associe à la forme précédente l'arbre noté



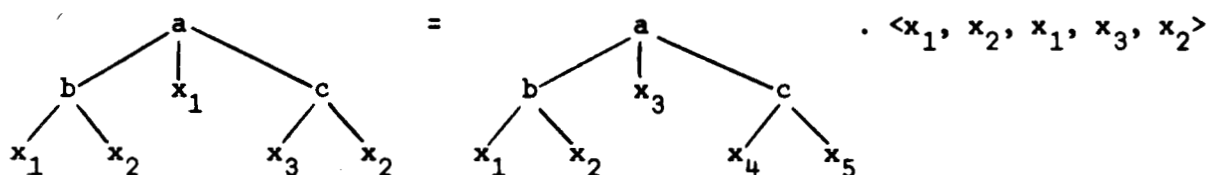
appelé \hat{t} , arbre initial de t , élément du magmaïde $\hat{T}(\Sigma)$ (voir § 0.B.4).

Le 5-uple $\theta = (x_1 x_2 x_1 x_3 x_2)$ définit des contraintes d'égalité : on substituera dans t un même arbre en 1er et 3e rang, et en 2e et 5e rang.

θ est appelé torsion, élément du magmaïde des torsions Θ (voir § 0.B.2).

Le lemme 0.1 montre que tout arbre peut se décomposer canoniquement de façon unique en un arbre initial composé avec sa torsion :

Dans notre exemple, on écrira



Une forêt est un ensemble d'arbres de $T(\Sigma)_0^1$ c'est à dire des 1-uples sans variables dont les branches sont achevées par des lettres de Σ de degré 0, appelées terminaux.

La classe des forêts reconnaissables (Rec) est un exemple de forêts reconnues par des machines appelées les automates d'états finis d'arbres (Voir § 0.C.1).

L'étude d'une structure algébrique est inséparable de celle des morphismes associés : nous introduisons alors intuitivement la notion de morphisme d'arbres, transformation qui à un arbre associe un arbre.

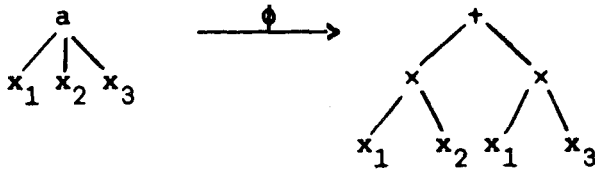
Un morphisme ϕ est une application de $T(\Sigma)$ dans $T(\Delta)$, compatible avec les produits de composition et tensoriel c'est à dire tel que

$$\phi(u.v) = \phi(u).\phi(v)$$

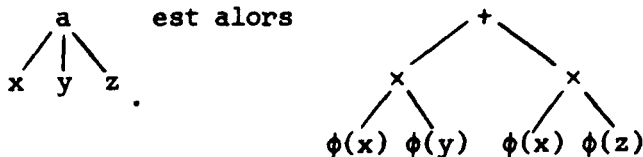
$$\phi(u \otimes v) = \phi(u) \otimes \phi(v)$$

Il est donc entièrement défini par sa donnée sur Σ et s'étend par composition.

Soit par exemple



l'image de



Dans cet exemple x_1 se répète et par duplication on recopie $\phi(x)$.

On dira que ϕ n'est pas linéaire puisqu'il génère des égalités. On voit alors immédiatement que Rec n'est pas close par morphisme c'est à dire que les morphismes ne conservent pas la reconnaissabilité ou encore que les automates d'états finis ne peuvent reconnaître $\phi(\text{Rec})$.

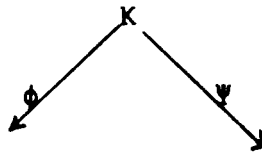
Précisons les différents cas des morphismes :

- ϕ est linéaire si pour tout $a \in \Sigma$ chaque variable de a apparaît au plus une fois dans $\phi(a)$, c'est à dire que ϕ n'effectue pas de copie.
- ϕ est complet si chaque variable de a apparaît au moins une fois dans $\phi(a)$ c'est à dire que, ϕ conservant toutes les variables, il conserve toute l'information contenue dans les arbres.
- ϕ est strict si l'image de toute lettre a de Σ contient au moins une lettre de Δ .
- ϕ est un démarquage s'il est linéaire et tel que $\phi(a)$ contient au plus une lettre de Δ .

- ϕ est un démarquage propre s'il est complet linéaire strict et respecte l'ordre des variables c'est à dire que ϕ ne transforme que les labels.

Ces différents cas sont illustrés par les exemples 0.5, 0.6, 0.7 et 0.8. Le lecteur verra en fait que les morphismes d'arbres que nous utiliserons sont des cas particuliers de morphismes de magmoïdes (voir en § 0.D.1). De même nous nous limiterons à des cas particuliers de bimorphismes de magmoïdes (voir en 0.E) dérivés des morphismes en tant que composés d'un morphisme inverse, d'une intersection avec des reconnaissables et d'un morphisme.

Un bimorphisme est un triplet $B = (\phi, K, \Psi)$ représenté par le schéma



et l'image par B d'une forêt F se définit par

$$B(F) = \Psi[\phi^{-1}(F) \cap K]$$

Considérons les deux exemples suivants :

1) Associativité de + :

soit

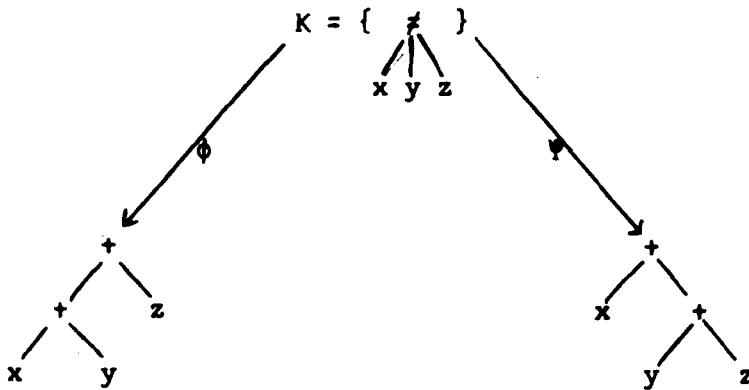
- la forêt K reconnaissable des arbres de la forme



- ϕ défini par $\phi(\begin{array}{c} z \\ / \quad | \quad \backslash \\ x \quad y \quad z \end{array}) = \begin{array}{c} + \\ / \quad \backslash \\ + \quad x_3 \\ / \quad \backslash \\ x_1 \quad x_2 \end{array}$

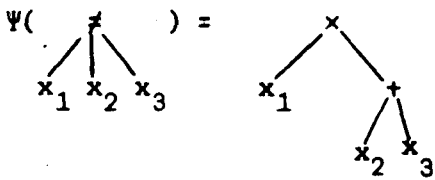
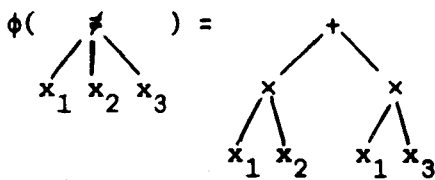
- Ψ défini par $\Psi(\begin{array}{c} z \\ / \quad | \quad \backslash \\ x_1 \quad x_2 \quad x_3 \end{array}) = \begin{array}{c} + \\ / \quad \backslash \\ x_1 \quad + \\ \quad \quad / \quad \backslash \\ \quad \quad x_2 \quad x_3 \end{array}$

Alors on a le schéma suivant d'un bimorphisme linéaire car ϕ et Ψ sont linéaires :

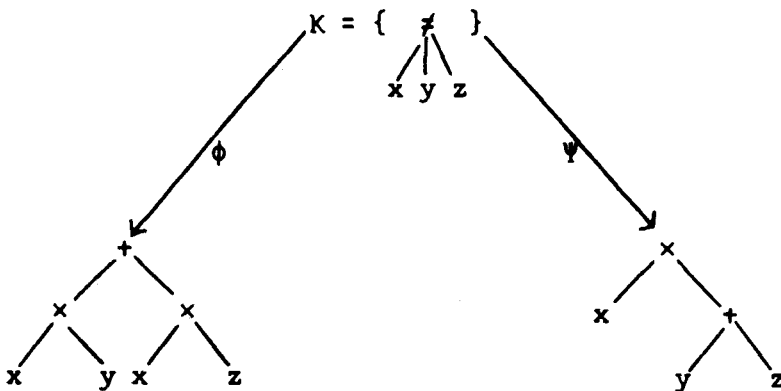


2) distributivité de + et \times :

Avec la même forêt K, on définit



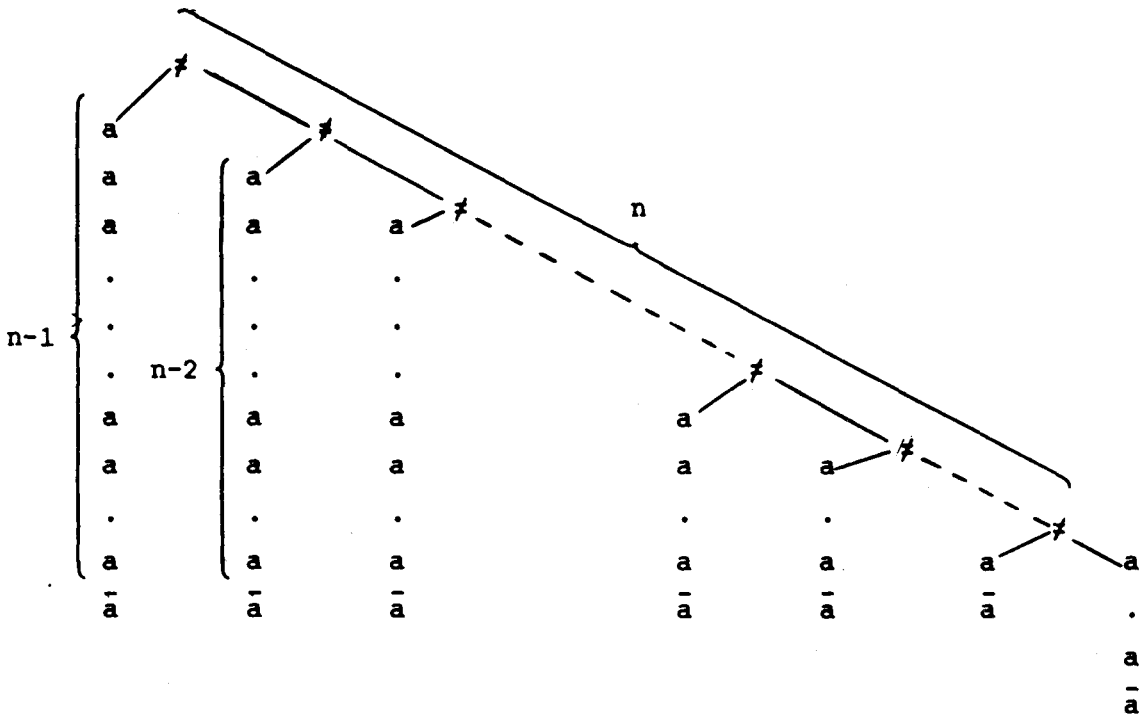
ϕ n'est pas linéaire car il génère des égalités et le bimorphisme associé qu'on représentera par



simule alors une 1ère phase d'analyse en vérifiant l'égalité des 1ère et 3ème branche, une 2e phase de contrôle par intersection avec une forêt

reconnaissable (ici K elle-même) et une 3e phase de synthèse par le morphisme Ψ .

Par génération d'égalités, la non linéarité d'un morphisme ne permet pas la conservation de la reconnaissabilité. Considérons l'exemple suivant d'un arbre où des contraintes d'égalité se propagent sur toute son image sans que leur nombre en soit borné.



L'ensemble de tels arbres n'est manifestement pas l'image par morphisme d'une forêt reconnaissable.

Il nous a cependant paru très intéressant de reconnaître de tels ensembles d'arbres et dans le chapitre II le lecteur trouvera la description formelle des machines qui reconnaissent ces classes d'arbres, machines que l'on a tout naturellement appelées "Reconnaisseurs Avec Test d'Égalité". Les forêts RATEG reconnues par ces automates se révéleront avoir de bonnes propriétés.

A - LES ARBRES

Définitions

0.1 - Un **alphabet gradué** est la donnée d'un couple (Σ, d) où Σ est un ensemble quelconque d'éléments appelés lettres et d une application de Σ dans \mathbb{N} associant à chaque lettre un degré ou arité. Par la suite nous prendrons toujours Σ fini.

Notation :

$\Sigma_i = d^{-1}(i)$ est le sous-ensemble des lettres de Σ de degré i .

0.2 - On appelle **ensemble des variables** (ou index) l'ensemble infini

$$X = \{x_1, x_2, \dots, x_n, \dots\}$$

Notation :

$$X_n = \{x_1, x_2, \dots, x_n\}$$

$$X_0 = \emptyset$$

0.3 - On appelle **ensemble des arbres sur Σ indexés par X_n** l'ensemble $T(\Sigma)_n^1$ défini par

$$- \Sigma_0 \cup X_n \subset T(\Sigma)_n^1$$

$$- \text{si } a \in \Sigma_p \text{ et } t_1, \dots, t_p \in T(\Sigma)_n^1 \text{ alors } a(t_1, \dots, t_p) \in T(\Sigma)_n^1$$

Remarque :

On trouvera dans d'autres ouvrages la notation $T_\Sigma(X_n)$ pour $T(\Sigma)_n^1$.

On verra plus loin comment les notations de magmaïde justifient cette écriture.

0.4 - Un **noeud** d'un arbre t est une occurrence dans t d'une lettre de Σ ou d'un index.

0.5 - Le label d'un noeud étant cette lettre ou cet index.

0.6 - La taille d'un arbre t noté $|t|$ est le nombre de ses noeuds, plus précisément :

- si $t = x_i \in X_n$ $|t| = 0$
- si $t = a \in \Sigma_0$ $|t| = 1$
- si $t = a(t_1 \dots t_n)$ $|t| = 1 + \sum_{i=1}^n |t_i|$

Sur tout arbre t on peut définir un ordre partiel noté $<_t$ entre ses noeuds :

- $a R_t b$ ssi t s'écrit

$$t = t_1(r_1 \dots r_i, a(s_1 \dots s_j, b(u_1 \dots u_k), s_{j+2} \dots s_\ell), r_{i+2} \dots r_n)$$

0.7 - b est dit successeur immédiat de a

et a prédecesseur immédiat de b

- $<_t$ est la clôture transitive de R_t .

Cet ordre introduit de nouvelles définitions :

0.8 - Le sommet de t est l'élément de t minimal pour $<_t$

0.9 - Les feuilles de t sont les noeuds autres que les variables, maximaux pour $<_t$ donc de label appartenant à Σ_0 .

0.10- Un arbre sans variables est un arbre de $T(\Sigma_0^1)$

0.11- Le feuillage d'un arbre sans variable est le mot obtenu par concaténation des feuilles. C'est l'application ϕ définie par $\phi : T(\Sigma_0^1) \rightarrow \Sigma_0^*$

- si $t \in \Sigma_0$ $\phi(t) = t$
- si $t = t_0(t_1, t_2, \dots, t_n)$ $\phi(t) = \phi(t_1) \phi(t_2) \dots \phi(t_n)$

0.12 Une sous-branche de t est une suite $(a) = (a_1 \dots a_p)$ où $\forall i$ a_{i+1} est successeur immédiat de a_i .

0.13 La longueur d'une sous branche (a) notée $\text{long}(a)$ est

$\text{long}(a) = p-1$ si a_p est une variable

$= p$ sinon ($a_p \in \Sigma_0$)

0.14 Une branche de t est une sous-branche où a_1 est minimal (sommet) et a_p maximal (feuille ou variable).

0.15 La profondeur de t est la borne supérieure des longueurs de ses branches.

0.16 Un arbre h -homogène ou de profondeur homogène h est un arbre dont toutes les branches joignant le sommet à une variable sont de longueur h , les autres joignant le sommet à une feuille de Σ_0 étant de longueur $\leq h$.

0.17 La distance entre deux noeuds α, β d'un arbre t est définie par

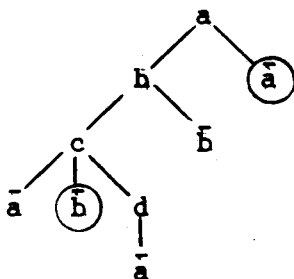
$$d_t(\alpha, \beta) = \sup [\text{long}(\alpha, v), \text{long}(\beta, v)]$$

où v est le premier prédecesseur commun à α et β .

Exemple 0.1 : Un arbre de $T(\Sigma)$:

Soit $\Sigma = \{(c, 3), (a, 2), (b, 2), (d, 1), (\bar{a}, 0), (\bar{b}, 0)\}$

$t = a(b(c(\bar{a}, \bar{b}), d(\bar{a})), \bar{b}), \bar{a}$ se représente



Il est sans variable, de sommet a , de taille 9, de profondeur 5, de feuillage $\bar{a} \bar{b} \bar{a} \bar{b} \bar{a}$; c est un successeur immédiat de b ; $(b \ c \ d)$ est une sous-branche ; $(a \ b \ \bar{b})$ est une branche de longueur 3. La distance de \bar{b} à \bar{a} (noeuds cerclés) est 3.

B - LES MAGMOIDES

Notation :

On notera $[p]$ l'ensemble $\{1, 2, \dots, p\}$ des entiers de 1 à p .

1) Définitions

0.18 Un **magmoïde** est un 5-uple $\langle M, \cdot, \otimes, e, e_0 \rangle$ noté M (en confondant structure algébrique et support) où

- M est un ensemble
- \cdot une opération binaire partielle sur M appelée composition
- \otimes une opération binaire sur M appelée produit tensoriel
- e et e_0 2 éléments distingués de M

et qui vérifie les 5 axiomes suivants :

0.19 (i) - $\forall p, q \in \mathbb{N}$, il existe une partie $M_{q,p}^D$ de M appelée fibre p - q de M . Les fibres sont disjointes deux à deux et leur réunion est l'ensemble M .

Tout élément de $M_{q,p}^D$ est dit de degré inférieur q , de degré supérieur p .

(ii) - $\forall p, p', q, q' \in \mathbb{N}$, $\forall m \in M_{q,p}^D$, $\forall m' \in M_{q',p'}^D$,

$m \cdot m'$ est défini ssi $q = p'$ et $m \cdot m' \in M_{q,p}^D$. Le produit de composition est associatif.

(iii) - $\forall p, p', q, q' \in \mathbb{N}$, $\forall m \in M_{q,p}^D$, $\forall m' \in M_{q',p'}^D$,

$m \otimes m' \in M_{q+q',p+p'}^D$. Le produit tensoriel est associatif.

(iv) - $\forall m_1, m_2, m'_1, m'_2 \in M$

si $(m_1 \cdot m_2) \otimes (m'_1 \cdot m'_2)$ est défini, il est égal à $(m_1 \otimes m'_1) \cdot (m_2 \otimes m'_2)$

(v) - $e \in M_1^1$, $e_0 \in M_0^0$

$\forall p$ soit $e_p = \underbrace{e \otimes e \otimes \dots \otimes e}_p \in M_p^D$

On a : $\forall m \in M_{q,p}^D$ $e_p \cdot m = m$

$\forall m \in M_{q,p}^D$ $m \cdot e_p = m$

$e_0 \otimes m = m \otimes e_0 = m$

e_p et e_o sont les éléments neutres respectifs de la composition et du produit tensoriel. Ils sont uniques pour chaque fibre M_p^D .

Nous rappelons **ensuite** trois exemples classiques de magmoïdes.

2) Le magmoïde des torsions

0.20 Une torsion θ est une application de $[p]$ dans $[q]$ o ù $p, q \in \mathbb{N}$ et on identifiera θ à un p -uplet

$$\langle q ; x_{\theta(1)}, x_{\theta(2)}, \dots, x_{\theta(p)} \rangle$$

Remarque importante :

On notera indifféremment une torsion avec ses variables x_i ou les seuls **indices** i . Par exemple $\langle 3 ; 1, 1, 2 \rangle = \langle 3, x_1, x_1, x_2 \rangle$

On appelle θ_q^D l'ensemble des applications de $[p]$ dans $[q]$. Produits de composition et tensoriel sont définis par

$$- \text{ soit } \theta \in \theta_q^D, \theta' \in \theta_r^q$$

$\theta \cdot \theta'$ est l'application de $[p]$ dans $[r]$ égale à $\theta' \circ \theta$.

$$- \text{ soit } \theta \in \theta_q^D, \theta' \in \theta_{q'}^{D'}$$

$\theta \otimes \theta'$ est l'application de $[p+p']$ dans $[q+q']$ égale à

$$\theta(i) \text{ si } i \leq p$$

$$q + \theta'(i-p) \text{ si } p < i \leq p+p'$$

L'élément neutre e est l'unique application de $[1]$ dans $[1]$ noté Id_1 .

L'élément neutre e_o est l'unique application de θ_o^o noté Id_o .

$e_p \in \theta_p^D$ est l'application identique de $[p]$ dans $[p]$. On la notera Id_p .

N.B. si $p \neq 0$ $\theta_o^D = \emptyset$ $\theta_p^o = \{0\}$

On vérifie alors que $\theta = \cup_{p,q} \theta_q^D$ est un magmoïde.

Les projections

Définition : $\forall n \in \mathbb{N}, \forall i \in [n]$

π_n^i est l'élément de Θ_n^1 qui à 1 associe i .

0.21 π_n^i est appelée projection.

C'est un cas particulier de torsion. On notera .

$$\pi_n^i(t) = \pi_n^i \cdot t = \pi_n^i \langle p ; t_1, \dots, t_n \rangle = t_i$$

Exemple 0.2 : Projection

$$\pi_3^2 \langle 4 ; \begin{array}{c} a \\ / \ \backslash \\ 1 \ 2 \end{array}, \begin{array}{c} b \\ / \ \backslash \\ 3 \ 4 \end{array}, \begin{array}{c} c \\ | \\ 4 \end{array} \rangle = \begin{array}{c} b \\ / \ \backslash \\ 3 \ 4 \end{array}$$

3) Le magma de $T(\Sigma)$

Soit p et $q \in \mathbb{N}$.

On note $T(\Sigma)_q^p$ l'ensemble $\{q\} \times (T(\Sigma)_q^1)^p$ des p -uplets d'éléments de $T(\Sigma)_q^1$

indiqués dans $X_q \langle q ; t_1, t_2, \dots, t_p \rangle$ où $\forall i \in [p] t_i \in T(\Sigma)_q^1$

On note $T(\Sigma) = \bigcup_{p,q} T(\Sigma)_q^p$.

Le produit de composition se définit comme suit.

Soit $t \in T(\Sigma)_q^p$ et $u \in T(\Sigma)_n^q$

$$\begin{aligned} t \cdot u &= \langle q ; t_1, \dots, t_p \rangle \cdot \langle n ; u_1, \dots, u_q \rangle \\ &= \langle n ; v_1, \dots, v_p \rangle \in T(\Sigma)_n^p \end{aligned}$$

où $\forall i \in [p] v_i$ se déduit de t_i en substituant à chaque occurrence x_j de variable de t_i l'arbre u_j de u .

Exemple 0.3 : Produit de composition dans $T(\Sigma)$

$$t = \langle 3 ; \begin{array}{c} a \\ | \\ 1 \end{array}, \begin{array}{c} b \\ / \ \backslash \\ 1 \ 2 \end{array}, \begin{array}{c} c \\ | \\ 3 \end{array} \rangle \in T(\Sigma)_3^3$$

$$u = \langle 4 ; \begin{array}{c} a \\ | \\ 1 \end{array}, \begin{array}{c} b \\ / \quad \backslash \\ 2 \quad 2 \end{array}, \begin{array}{c} d \\ | \\ 4 \end{array} \rangle \in T(\Sigma)_4^3$$

$$t.u = \langle 4 ; \begin{array}{c} a \\ | \\ a \\ | \\ 1 \end{array}, \begin{array}{c} b \\ / \quad \backslash \\ a \quad b \\ | \quad / \quad \backslash \\ 1 \quad 2 \quad 2 \end{array}, \begin{array}{c} c \\ | \\ d \\ | \\ 4 \end{array} \rangle \in T(\Sigma)_4^3$$

L'élément neutre est alors

$$\text{Id}_p = \langle p ; x_1, x_2, \dots, x_p \rangle = \langle p ; 1, 2, \dots, p \rangle \in T(\Sigma)_p^p$$

Le produit tensoriel se définit comme suit .

$$\text{Soit } u = \langle q ; u_1, \dots, u_p \rangle \in T(\Sigma)_q^p$$

$$v = \langle q' ; v_1, \dots, v_{p'} \rangle \in T(\Sigma)_{q'}^{p'}$$

$$u \otimes v = \langle q+q' ; u_1, \dots, u_p, w_1, \dots, w_{p'} \rangle \in T(\Sigma)_{q+q'}^{p+p'}$$

où $\forall i \in [p']$

$w_i = v_i \cdot \langle q+q' ; q+1, q+2, \dots, q+q' \rangle$ c'est à dire que pour tout $i \leq p'$ on substitue x_{q+i} à x_i .

Exemple 0.4 : Produit tensoriel dans $T(\Sigma)$; pour t et u précédemment définis

on aura

$$t \otimes u = \langle 7 ; \begin{array}{c} a \\ | \\ 1 \end{array}, \begin{array}{c} b \\ / \quad \backslash \\ 1 \quad 2 \end{array}, \begin{array}{c} c \\ | \\ 3 \end{array}, \begin{array}{c} a \\ | \\ 4 \end{array}, \begin{array}{c} b \\ / \quad \backslash \\ 5 \quad 5 \end{array}, \begin{array}{c} d \\ | \\ 7 \end{array} \rangle \text{ dans } T(\Sigma)_7^6$$

On vérifie encore que $T(\Sigma)$ est un magmaïde.

Dans [A.D.1 et A.D.2] le lecteur trouvera la notion de magmaïde projetable.

L'idée est que tout magmaïde projetable contient le magmaïde des torsions et que chaque élément est caractérisé par ses "projections". On vérifie alors que l'on a bien formalisé ce qu'on voulait en vérifiant la

Propriété : $T(\Sigma)$ est le magmaïde projetable libre engendré par Σ .

Remarque importante :

Les éléments de $T(\Sigma)$ sont appelés arbres c'est à dire qu'on confondra sous

cette appellation la notion classique d'arbre t de $T(\Sigma)_0^1$, celle de p -uple $\langle q ; t_1, \dots, t_p \rangle$ et même celle de torsion $\langle q ; x_{\theta(1)} \dots x_{\theta(p)} \rangle$ correspondant à des éléments t_i réduits à une variable.

4) Le magmaïde $\hat{T}(\Sigma)$

Soit ϕ l'application qui à tout u de $T(\Sigma)$ associe le mot de X^* obtenu en prenant les variables de u de gauche à droite :

- soit $u = \langle q ; t_1 \dots t_p \rangle \in T(\Sigma)_q^p$

$\phi(u) = \phi(t_1) \times \dots \times \phi(t_p)$ (où \times est la multiplication dans le monoïde X^*)

- soit $u = \langle q \rangle \in T(\Sigma)_q^0$

$\phi(u) = \Lambda$ (élément neutre de X^*)

- soit $t \in T_\Sigma(X_q)$

$t \in \Sigma_0 \quad \phi(t) = \Lambda$

$t = x_i \in X_q \quad \phi(t) = x_i$

$t = a(t_1 \dots t_m) \quad \phi(t) = \phi(t_1) \times \dots \times \phi(t_m)$

On définit $\hat{T}(\Sigma)_q^p = \{u \in T(\Sigma)_q^p \mid \phi(u) = x_1 \dots x_q\}$

et $\hat{T}(\Sigma) = \bigcup_{p, q \geq 0} T(\Sigma)_q^p$.

On vérifie alors que $\hat{T}(\Sigma)$ est le magmaïde libre engendré par Σ et un sous-magmaïde de $T(\Sigma)$. L'élément neutre de $\hat{T}(\Sigma)$ étant $\text{Id}_p = \langle p ; x_1 \dots x_p \rangle \in \hat{T}(\Sigma)_p^p$.

L'ordre suivant noté \leq_I défini sur $\hat{T}(\Sigma)$ introduit la notion d'arbre initial :

$t' \leq_I t$ ssi $\exists u \in \hat{T}(\Sigma) : t = t' \cdot u$

0.22 On dit que t' est un sous-arbre initial de $t \in \hat{T}(\Sigma)$ est par conséquent l'ensemble des arbres initiaux.

Rappelons enfin le lemme suivant permettant d'exprimer de façon unique tout élément de $T(\Sigma)_q^p$ comme composition d'un élément de $\hat{T}(\Sigma)_p^p$ par une torsion de

θ_q^r :

Lemme 0.1 (décomposition canonique).

$\forall p, q \in \mathbb{N}, \forall u \in T(\Sigma)_q^p$

il existe un et un seul $r \in \mathbb{N}$

un et un seul $\tilde{u} \in \tilde{T}(\Sigma)_r^p$

une et une seule torsion $\theta \in \Theta_q^r$

tels que $u = \tilde{u} \cdot \theta$

Exemple 0.4

Nous avons défini $T(\Sigma) = \bigcup_{p,q} T(\Sigma)_q^p$ comme ensemble de p -uplets d'arbres à variables dans X_q . En particulier $T(\Sigma)_q^1$ ensemble des arbres sur Σ à variables dans X_q permet donc de décrire tout arbre avec le formalisme du magmaïde, ce qui se traduit sur l'exemple ($q = 4$) suivant par :

$$t = \begin{array}{c} a \\ / \quad \backslash \\ b \quad 4 \\ / \quad \backslash \\ c \quad 4 \\ / \quad \backslash \\ 1 \quad 2 \quad d \\ \quad \quad | \\ \quad \quad a \end{array} = a \cdot (b \otimes Id_1) \cdot (c \otimes Id_2) \cdot (Id_2 \otimes \bar{a} \otimes Id_2) \cdot \langle 4 ; 1, 2, 4, 4 \rangle$$

Remarque importante :

Dans Θ_q^p du magmaïde des torsions, toute torsion θ s'écrit $\langle q ; x_{\theta(1)} \dots x_{\theta(p)} \rangle$; dans $T(\Sigma)$ tout élément t de $T(\Sigma)_q^p$ s'écrit $\langle q ; t_1 \dots t_p \rangle$; dans $\tilde{T}(\Sigma)$ tout élément \tilde{t} de $\tilde{T}(\Sigma)_q^p$ s'écrit $\langle q, \tilde{t}_1, \dots, \tilde{t}_p \rangle$.

La nécessité de préciser par q l'ensemble X_q des index se justifie par la disjonction des fibres de magmaïde : en effet sinon si $q \leq q'$ on aurait $T(\Sigma)_q^1 \subset T(\Sigma)_{q'}^1$, et $T(\Sigma)_q^p \subset T(\Sigma)_{q'}^p$, ce qui est absurde.

Toutefois lorsque le contexte sous entendra de façon unique q sans confusion possible on mettra q dans l'écriture de θ , t ou \tilde{t} .

C - LES FORETS

1) Définitions

0.23 Une forêt F sur l'alphabet gradué Σ est une partie de $T(\Sigma)_0^1$

0.24 Un automate d'états finis d'arbres M est un quadruplet $M = (\Sigma, Q, F, R)$ où

- Σ est un alphabet gradué fini
- Q un ensemble fini d'états
- $F \subset Q$ un ensemble d'états distingués
- R un ensemble fini de règles

Notons $Q[\Sigma]$ l'alphabet gradué défini par

$$Q[\Sigma] = \{q[a(x_1 \dots x_n)] \mid q \in Q, a(x_1 \dots x_n) \in \Sigma_n\}$$

Il existe deux sortes d'automates, ascendant ou descendant avec des règles définies comme suit :

0.25 règles ascendantes : F est l'ensemble des états finaux,

$$- a(q_{i_1}[x_1] \dots q_{i_n}[x_n]) \rightarrow q[a(x_1 \dots x_n)]$$

si $a \in \Sigma_n$

et $q, q_{i_1} \dots q_{i_n} \in Q$

$$- a \rightarrow q[a] \quad \text{si } a \in \Sigma_0$$

0.26 Un mouvement de l'automate $t \xrightarrow{M} u$ se définit par :

$$t \xrightarrow{M} u \text{ ssi } - t = t_0 \cdot a(q_{i_1}[t_1] \dots q_{i_n}[t_n])$$

$$- u = t_0 \cdot q[a(t_1 \dots t_n)]$$

$$\text{et } - a(q_{i_1}[x_1] \dots q_{i_n}[x_n]) \rightarrow q[a(x_1 \dots x_n)] \in R$$

On note \xrightarrow{M}^* la clôture transitive de \xrightarrow{M} .

Définition :

$t \in T(\Sigma)_0^1$ est reconnu par M ssi

$\exists q \in F : t \stackrel{*}{\underset{M}{\mid}} q[t]$

Notation

$F(M) = \{t \in T(\Sigma)_0^1 \mid \exists q \in F : t \stackrel{*}{\underset{M}{\mid}} q[t]\}$

$F(M)$ est la forêt reconnue par M .

0.27 règles descendantes : F est l'ensemble des états initiaux.

- $q[a(x_1 \dots x_n)] \rightarrow a(q_{i_1}[x_1] \dots q_{i_n}[x_n])$

- $q[a] \rightarrow a$ si $a \in \Sigma_0$

Un mouvement est défini par :

$t \stackrel{*}{\underset{M}{\mid}} u$ ssi - $t = t_0 q[a(t_1 \dots t_n)]$

- $u = t_0 a(q_{i_1}[t_1] \dots q_{i_n}[t_n])$

et - $q[a(x_1 \dots x_n)] \rightarrow a(q_{i_1}[x_1] \dots q_{i_n}[x_n]) \in R$

On note $\stackrel{*}{\underset{M}{\mid}}$ la clôture transitive de $\underset{M}{\mid}$ et on dit que $t \in T(\Sigma)_0^1$ est reconnu par M ssi $\exists q \in F : q[t] \stackrel{*}{\underset{M}{\mid}} t$

Définition :

0.28 Une forêt $F \in T(\Sigma)_0^1$ est reconnaissable ssi il existe un automate d'états finis ascendant M tel que $F = F(M)$.

Notation :

On note Rec la classe des forêts reconnaissables.

Propriétés :

◁ M peut toujours être choisi déterministe dans le cas "ascendant".

◁ M peut être pris "descendant" mais on ne peut plus toujours le choisir déterministe.

2) Grammaires

0.29 Définitions : Une grammaire d'arbres G est un quadruplet $G = (\Sigma, V, R, S)$ où

- Σ est un alphabet gradué fini terminal
- V un alphabet gradué fini de variables (syntaxiques)
- S l'axiome $S \in V$ et de degré 0
- R un ensemble fini de règles de dérivation

0.30 Une règle est la donnée d'un couple $g \rightarrow d$ tel qu'il existe $p \in \mathbb{N}$ appelé

degré : g et $d \in T(\Sigma \cup V)_p^1$.

On écrit $g \rightarrow d$

ou encore $\tilde{g} \rightarrow d$ d'après le lemme 0.1

0.31 Une dérivation notée $t \Rightarrow_G u$ se définit par :

t et $u \in T(\Sigma \cup V)_0^1$ tels que

$$t = v.g.w$$

$$u = v.d.w$$

et $g \rightarrow d \in R$

On note $\stackrel{*}{\Rightarrow}_G$ la clôture transitive de \Rightarrow_G et

0.32 on appelle $F(G)$ la forêt engendrée par G :

$$F(G) = \{t \in T(\Sigma)_0^1 \mid S \stackrel{*}{\Rightarrow}_G t\}$$

Définitions :

0.33 Une règle est régulière ssi elle est de la forme

$$X \rightarrow a(x_1 \dots x_n)$$

où $a \in \Sigma_n$

$$X, x_1 \dots x_n \in V_0$$

- * une grammaire est régulière ssi R est un ensemble de règles régulières
- * une forêt est régulière ssi elle est engendrée par une grammaire régulière.

Propriétés :

- La classe des forêts régulières coïncide avec Rec
- Elle est close par union, intersection, homomorphisme linéaire et homomorphisme inverse.

(Voir les morphismes ci-après § 0.D).

Propriété fondamentale :

Il y a identité entre la classe des langages algébriques sans mot vide et la classe des feuillages des forêts reconnaissables.

Rappelons qu'un langage algébrique est engendré par une grammaire algébrique $G = (\Sigma \cup V, R, S)$ où $\Sigma \cup V$ est un alphabet, S l'axiome et R un ensemble de règles de la forme $\xi \rightarrow u$ où ξ et $u \in (\Sigma \cup V)^*$.

D - LES MORPHISMES

1) Morphisme de magmoïde

0.34 Définition : Une application ϕ du magmoïde M dans le magmoïde N est un k -morphisme de magmoïde ssi

i) ϕ conserve les degrés :

$$\forall m \in M_q^D \quad \phi(m) \in N_q^D$$

ii) ϕ préserve les éléments neutres

iii) ϕ est compatible avec la composition et le produit tensoriel :

$$- \forall m \in M_q^D, \quad \forall m' \in M_r^D$$

$$\phi(m.m') = \phi(m).\phi(m')$$

$$- \forall m \in M_q^D, \quad \forall m' \in M_{q'}^{D'}$$

$$\phi(m \otimes m') = \phi(m) \otimes \phi(m')$$

Un morphisme de magmoïde de $T(\Sigma) \rightarrow M$ est donc totalement défini par sa donnée sur l'alphabet Σ .

Le lecteur se reportera à [A.D.1] pour l'étude formelle de ces morphismes.

2) Morphisme d'arbres

Nous considérerons ici les morphismes d'arbres comme cas particuliers de 1-morphismes de magmoïde

Définitions

0.35 Un **morphisme d'arbre** $\phi : T(\Sigma) \rightarrow T(\Delta)$ est défini par l'application $\Sigma \rightarrow T(\Delta)$ qui respecte les degrés ($\phi(\Sigma_i) \subset T(\Delta)_i^1$) et s'étend par composition.

0.36 Un morphisme est **linéaire** ssi chaque lettre a de Σ apparaît au plus une fois dans $\phi(a)$.

Pour dupliquer ou faire des copies de sous-arbres le morphisme utilisé sera non-linéaire.

Exemples 0.5 :

Soit Σ un alphabet fini et $a \in \Sigma$.

- ϕ_1 défini par - l'identité sur $\Sigma - \{a\}$

$$-\phi_1\left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array}\right) = \begin{array}{c} a \\ / \quad \backslash \\ 1 \quad b \\ \quad \quad \backslash \\ \quad \quad \quad 2 \end{array}$$

ϕ_1 est linéaire.

- ϕ_2 défini par - l'identité sur $\Sigma - \{a\}$

$$-\phi_2\left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array}\right) = \begin{array}{c} a \\ / \quad \backslash \\ 1 \quad b \\ \quad \quad / \quad \backslash \\ \quad \quad \quad 1 \quad 2 \end{array}$$

n'est pas linéaire et duplique la branche gauche de a .

0.37 Un morphisme est **complet** ssi toute lettre a de Σ apparaît au moins une fois dans $\phi(a)$.

Exemple 0.6

ϕ défini par l'identité sur $\Sigma - \{a\}$

$$\text{et } \phi\left(\begin{array}{c} a \\ / \quad | \quad \backslash \\ 1 \quad 2 \quad 3 \end{array}\right) = \begin{array}{c} a \\ / \quad \backslash \\ 1 \quad b \\ \quad \quad / \quad | \quad \backslash \\ \quad \quad \quad 1 \quad 2 \quad 3 \end{array}$$

est complet (mais non linéaire).

Nous verrons que la notion de complétude joue un rôle essentiel dans la conservation de toutes les informations en recopiant au moins une fois chaque branche d'un arbre dans son image.

0.38 Un morphisme est **strict** si pour toute lettre a de Σ , l'image $\phi(a)$ contient au moins une lettre de Δ .

(Ceci est analogue à la notion de morphisme non-effaçant en théorie des langages).

Exemple 0.7

ϕ défini par l'identité sur $\Sigma - \{a\}$

$$\text{et } \phi \left(\begin{array}{c} a \\ / \quad | \quad \backslash \\ 1 \quad 2 \quad 3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ 1 \quad \beta \\ | \\ 2 \end{array}$$

est strict, non complet, linéaire.

0.39 Un **démarquage** est un morphisme linéaire "alphabétique" tel que l'image $\phi(a)$ de toute lettre a de Σ contient au plus une lettre de Δ .

0.40 Un **démarquage propre** est un morphisme linéaire complet strict, qui ne transforme que les labels ($\forall a \in \Sigma_i \quad \phi(a) \in \Delta_i$) et respecte l'ordre des variables.

Exemple 0.8

- ϕ défini par l'identité sur $\Sigma - \{a\}$

$$\text{et } \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) = 1$$

est un démarquage non complet non strict.

- ϕ défini par l'identité sur $\Sigma - \{a\}$

$$\text{et } \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ 1 \quad 2 \end{array}$$

est un démarquage propre.

Remarque

Nous utiliserons encore le fait que toute image $\phi(a)$ pourra se décomposer en un arbre initial $\phi(a) \in \mathcal{T}(\Delta)$ et une torsion θ et qu'un morphisme sera donc défini entre autres par sa torsion.

E - LES BIMORPHISMES

La notion de bimorphismes de magmaïde introduite par M. Dauchet [D] est une généralisation de la notion de bimorphismes de langages.

Nous nous intéressons ici à une classe restreinte de bimorphismes dont on considérera la définition restreinte suivante :

0.41 Définition (restreinte)

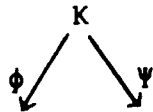
Nous appellerons bimorphisme un triplet $B = (\phi, K, \Psi)$ où

- ϕ et Ψ sont des morphismes (d'arbres)
- K une forêt reconnaissable.

Nous confondrons sous la même appellation la transformation associée.

$$\hat{B} = \{(\phi(t), \Psi(t)) \mid t \in K\}$$

et on utilisera le schéma suivant



L'image par un bimorphisme B d'une forêt F est alors définie par

$$\begin{aligned} B(F) &= \{u \mid \exists t \in K : (t, u) \in \hat{B}\} \\ &= \{\Psi[\phi^{-1}(w) \cap K] \mid w \in F\} \end{aligned}$$

0.42 Nous dirons qu'un bimorphisme est linéaire respectivement complet, strict ssi les 2 morphismes ϕ et Ψ sont linéaires respectivement complets, stricts.

CHAPITRE I

GENERATION DES FORETS RECURSIVEMENT ENUMERABLES

A PARTIR DE FORETS RECONNAISSABLES

Dans ce chapitre, nous montrons que la clôture par union intersection et morphisme des forêts reconnaissables est l'ensemble des forêts récursivement énumérables.

Plus précisément toute forêt récursivement énumérable s'obtient comme simple image par un démarquage strict de l'intersection des images par homomorphismes complets stricts de deux forêts reconnaissables. C'est le théorème I.1.

On peut remarquer que tous les morphismes sont stricts. Si l'on impose comme seule hypothèse supplémentaire que les morphismes soient complets, on obtient alors une inclusion dans la classe des forêts récursives comme nous le verrons au chapitre suivant.

Théorème I.1

Quelle que soit la forêt F récursivement énumérable, il existe

- K_1 et K_2 forêts reconnaissables
- ϕ_1 et ϕ_2 morphismes complets stricts injectifs
- π démarquage strict

tels que $F = \pi[\phi_1(K_1) \cap \phi_2(K_2)]$

Idée de la preuve

Toute forêt récursivement énumérable peut être selon un résultat connu [A.D.3] engendrée par une grammaire d'arbres G "sans torsion" d'axiome S , d'alphabet $\Sigma \cup V$, d'ensemble fini R de règles de la forme $R_i : l_i \rightarrow r_i$ sans torsion où l_i et r_i ont même degré d_i .

Il paraît donc naturel d'essayer de simuler les dérivations dans G de façon analogue à la simulation en langages [A.D.3] et de ne garder que les éléments finalement engendrés par le démarquage π .

Nous allons donc montrer l'équivalence

$$(1) [S \Rightarrow r_{i_1} = u_2 \ell_{i_2} v_2 = u_2 r_{i_2} v_2 \dots \Rightarrow u_n r_{i_n} v_n = T]$$

\Leftrightarrow

$$(2) [\forall T \in F \exists T_1 \in K_1, \exists T_2 \in K_2 : T = \pi(\phi_1(T_1) \cap \phi_2(T_2))]$$

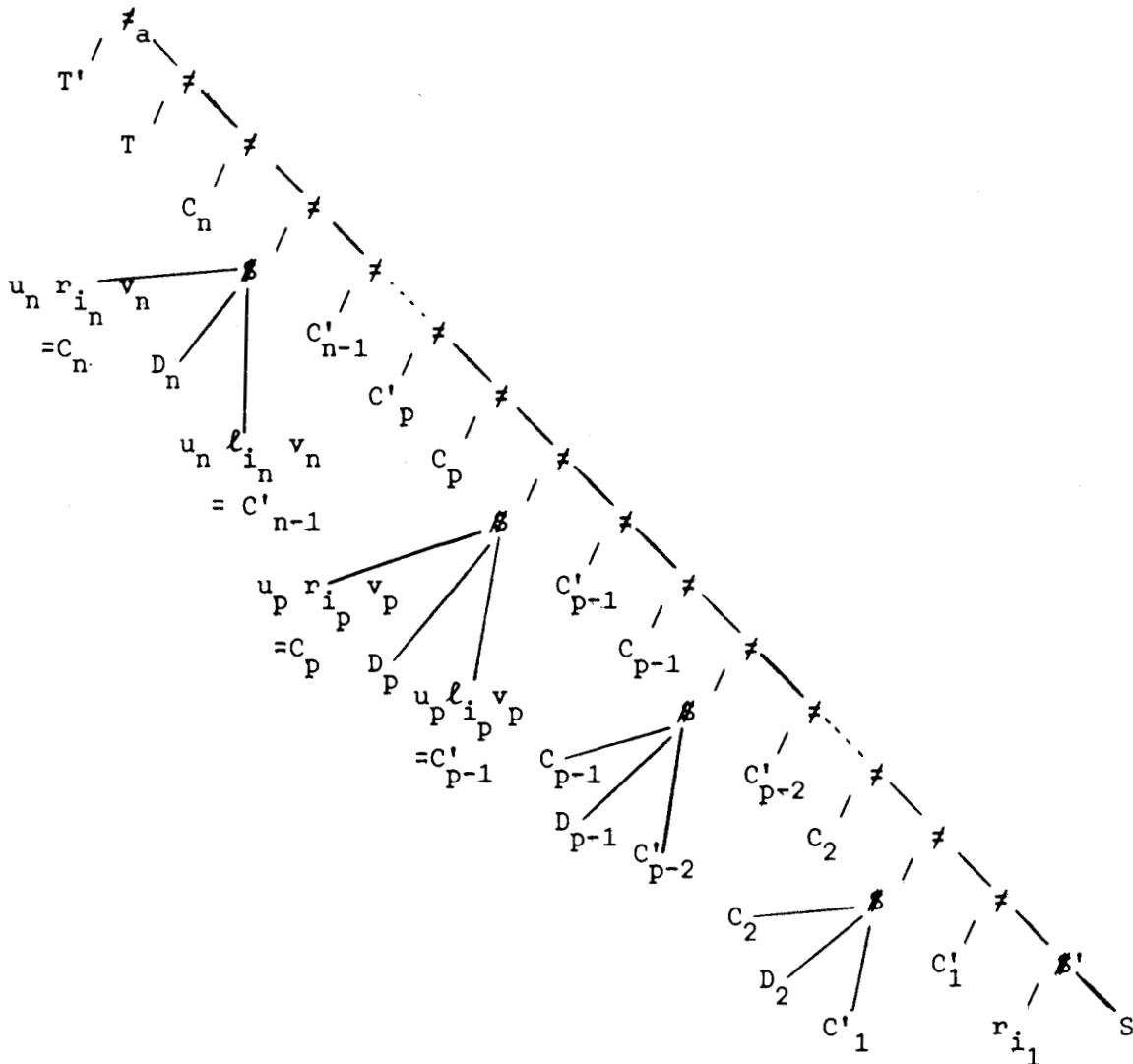
et plus précisément prouver

$$(1) [S \Rightarrow r_{i_1} = u_2 \ell_{i_2} v_2 \Rightarrow u_2 r_{i_2} v_2 \dots$$

$$\dots u_{p-1} r_{i_{p-1}} v_{p-1} = u_p \ell_{i_p} v_p \Rightarrow u_p r_{i_p} v_p \dots \Rightarrow u_n r_{i_n} v_n = T]$$

\Leftrightarrow

(2') il existe dans $\phi_1(K_1) \cap \phi_2(K_2)$ un arbre de la forme suivante :

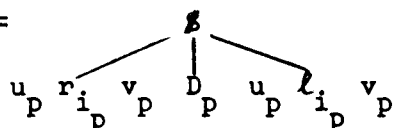


ϕ_2 morphisme non linéaire (donc engendrant des égalités) va entre autres imposer le bon enchaînement des dérivations par l'égalité des contextes

$$C_j = C'_j.$$

Reste alors à simuler un pas de dérivation $u_p \ell_{i_p} v_p \Rightarrow u_p r_{i_p} v_p$ par un

arbre de la forme $t =$



(l'utilité de D_p apparaîtra dans la suite).

Dans le cas des langages ceci se réalise aisément par la génération de

l'ensemble

$$\{(\tilde{u} \tilde{r} \tilde{v} \beta v \ell u \neq)^* \mid \ell \rightarrow r \in R, u \text{ et } v \in \Sigma^*\}$$

par une grammaire algébrique.

Dans le cas des arbres c'est ici que se situe la difficulté : on ne peut en effet générer des arbres de la forme de t par un seul homomorphisme.

En effet u_p est en général non borné et on ne peut générer des arbres

"presque égaux" (en général $\ell_{i_p} \neq r_{i_p}$) que si u_p est borné comme le précise le lemme suivant.

Lemme

Soit u l'image d'un arbre \bar{t} par un homomorphisme ϕ tel que $\alpha \neq \alpha'$ et

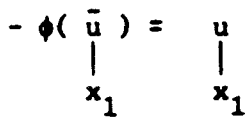
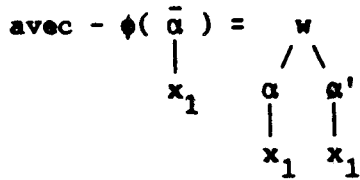
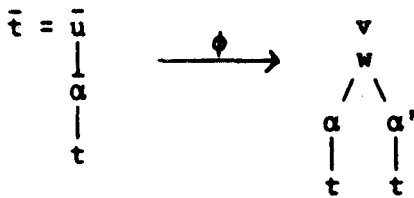
$$\begin{array}{cc}
 \alpha & \alpha' \\
 / \quad \backslash & \\
 | & | \\
 t & t
 \end{array}
 \quad \text{tel que les deux occurrences de } t \text{ soient images d'un}$$

même sous arbre de \bar{t} .

Alors la distance $d(\alpha, \alpha')$ est bornée.

La preuve, formellement laissée au lecteur, repose sur le fait que α et α' occurrent nécessairement dans l'image d'un même noeud $\bar{\alpha}$ de \bar{t} .

On a le schéma suivant :



$$- \phi(t) = t$$

Nous allons montrer comment, par conjonction de contraintes d'égalités générées par ϕ_1 et ϕ_2 "le long de D_p " nous parvenons néanmoins à réaliser cette simulation d'un pas de dérivation.

Nous introduisons les forêts K'_1 , K'_2 et $K' = K'_1 \cap K'_2$ des sous-arbres dont les sommets sont les symboles \neq occurant à une distance égale à 2 modulo 3 du sommet des arbres respectifs de $\phi_1(K_1)$, $\phi_2(K_2)$ et $\phi_1(K_1) \cap \phi_2(K_2)$. Nous montrons alors dans le corollaire I.3 comment tout arbre t de K' simule un pas de dérivation par

$$t \in K' \Leftrightarrow \exists R : \ell \rightarrow r \in G$$

$$\exists u, v \text{ et } D :$$

$$t = \begin{array}{c} \delta \\ / \quad \backslash \\ u \quad u \\ \ell \quad D \quad r \\ v \quad v \end{array}$$

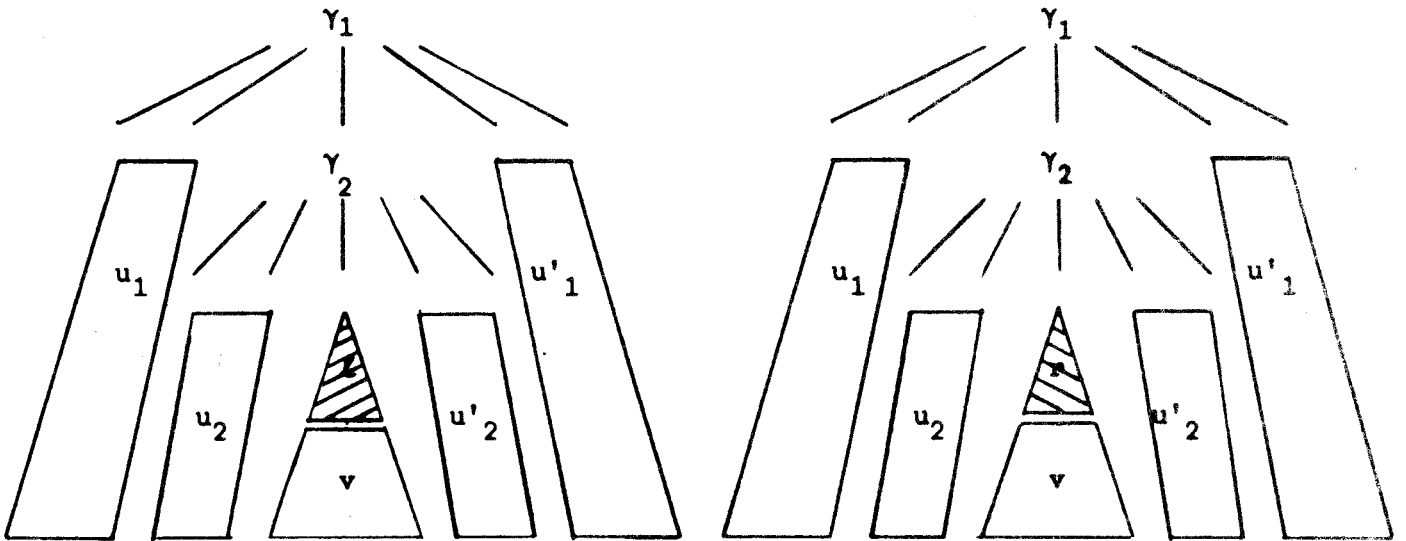
l'idée est de parcourir les sous branches menant du sommet de u jusqu'au sommet de ℓ (ou r) en décomposant $u\ell v$ et $u r v$ le long de l'arbre D et en assurant

- par ϕ_1 les contraintes d'égalités des contextes
- par ϕ_2 le bon enchaînement des dérivations.

Cette idée est illustrée par l'exemple I.1 suivant et précisée par le lemme I.2.

Exemple I.1

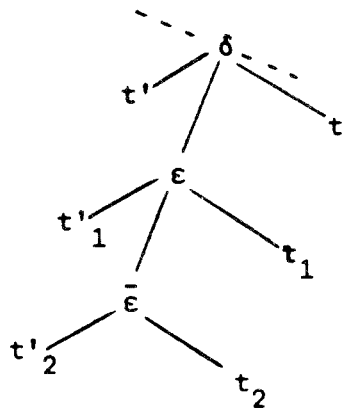
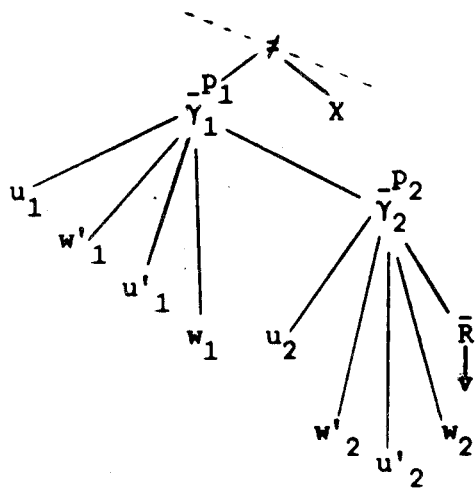
Considérons la dérivation $u\ell v \Rightarrow urv$ décrite par le schéma suivant, utilisant la règle $\bar{R} : \ell \rightarrow v$ de G et où γ_1 et γ_2 sont des lettres de Σ :



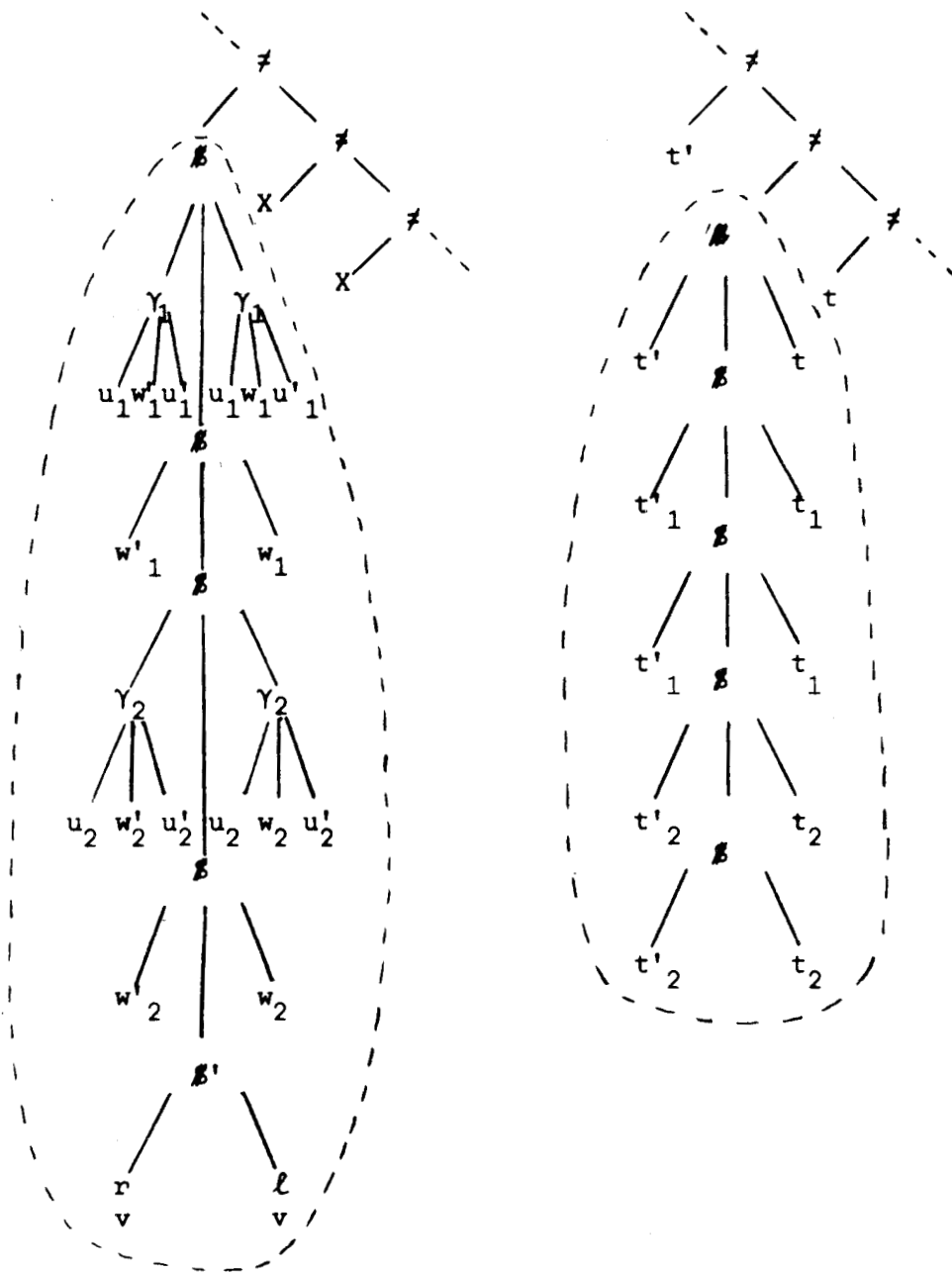
Les branches correspondant à cette dérivation seront respectivement de la forme suivante :

dans K_1 :

, dans K_2 :

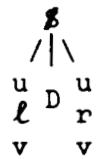


Ce qui donnera par application de ϕ_1 et ϕ_2



(Notons au passage que l'enchaînement des dérivations est assuré par $X = t$).

La branche correspondante de $K' = K'_1 \cap K'_2$ est alors de la forme représentée sur les arbres ci-dessus par le sous-arbre de pointillés.



On y vérifie que ϕ_1 impose l'égalité des contextes u_i et u'_i pour chaque lecture d'un γ_i et ϕ_2 impose le bon enchaînement en vérifiant

$$t'_2 = r = w'_2, t_2 = \ell = w_2$$

$$t'_1 = \begin{array}{c} / \quad \backslash \\ \quad \gamma_2 \\ u_2 \quad r \quad u'_2 \\ \quad \quad \quad \backslash \\ \quad \quad \quad \quad v \end{array} = w'_1, t_1 = \begin{array}{c} / \quad \backslash \\ \quad \gamma_2 \\ u_2 \quad \ell \quad u'_2 \\ \quad \quad \quad \backslash \\ \quad \quad \quad \quad v \end{array} = w_1$$

$$t' = \begin{array}{c} / \quad \backslash \\ \quad \gamma_1 \\ u_1 \quad \gamma_2 \quad u'_1 \\ / \quad \backslash \\ u_2 \quad r \quad u'_2 \\ \quad \quad \quad \backslash \\ \quad \quad \quad \quad v \end{array} = ulv, t = \begin{array}{c} / \quad \backslash \\ \quad \gamma_1 \\ u_1 \quad \gamma_2 \quad u'_1 \\ / \quad \backslash \\ u_2 \quad \ell \quad u'_2 \\ \quad \quad \quad \backslash \\ \quad \quad \quad \quad v \end{array} = urv$$

Preuve du théorème I

Outre la définition de ϕ_1 , ϕ_2 et π , il faut définir les forêts K_1 et K_2 sur de nouveaux alphabets et introduire les nouveaux symboles suivants :

1) Définition de nouvelles lettres

* $\bar{\Sigma}$:

$\forall \alpha \in \Sigma_n$ on lui associe $\{\bar{\alpha}^p \mid p \in [n], \bar{\alpha}^p \in \bar{\Sigma}_{n+2}\}$

$$\bar{\Sigma} = \bigcup_{n, \alpha \in \Sigma_n} \bar{\Sigma}_{n+2}$$

* \bar{R} :

$\forall R_i : \ell_i \rightarrow r_i$ on lui associe \bar{R}_i de même degré d_i que ℓ_i et r_i ;

$$\bar{R} = \bigcup_{i \in |R|} \bar{R}_i$$

* $\bar{\bar{R}}$:

$\forall R_i : S \rightarrow r$ de membre gauche réduit à l'axiome S on lui associe $\bar{\bar{R}}_i$ de degré 0.

$$\bar{\bar{R}} = \bar{\bar{R}}_i$$

* D_Σ :

$\forall a \in \Sigma_n$ on lui associe \bar{z}_a de degré $n+1$.

$$D_\Sigma = \bigcup_{n, a \in \Sigma_n} \{\bar{z}_a\}$$

- * \neq de degré 3
- * δ de degré 4
- * δ' de degré 2
- * ε de degré 3
- * ε' de degré 2

Notation :

Soit T un arbre de sommet $a \in \Sigma_n$, nous noterons T' le n -uplet des sous-arbres successeurs de a , tel que $T = a.T'$.

2) Définition de la forêt reconnaissable K_1 :

K_1 est définie sur l'alphabet

$$\Delta_1 = \Sigma \cup V \cup \bar{\Sigma} \cup \bar{R} \cup \bar{R}_k \cup D_{\Sigma} \cup \{\neq\}$$

et dont les arbres sont de la forme

$$T_1 = \neq_a . (T' \otimes \neq) . (U_n \otimes W_n \otimes \neq) . (U_{n-1} \otimes W_{n-1} \otimes \neq) \\ \dots (U_2 \otimes W_2 \otimes \bar{R}_k)$$

où

$$- \forall i, 2 \leq i \leq n \quad W_i \in T(\Sigma \cup V)_0^1$$

$$- \bar{R}_k \in \bar{R}$$

$$- T' \in T(\Sigma)_0^n \text{ tel que } a.T' = T \in F$$

- $\forall i, 2 \leq i \leq n \quad U_i$ a la forme suivante :

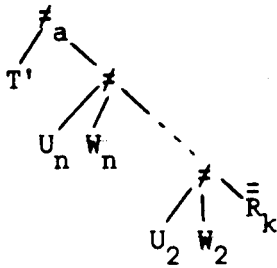
$$U_i = \bar{\alpha}_{j_{1,i}}^{-p_1} (u_{1,i} \otimes w'_{1,i} \otimes u'_{1,i} \otimes w_{1,i} \otimes \bar{\alpha}_{j_{2,i}}^{-p_2}) \\ \cdot (u_{2,i} \otimes w'_{2,i} \otimes u'_{2,i} \otimes w_{2,i} \otimes \bar{\alpha}_{j_{3,i}}^{-p_3}) \\ \dots (u_{m_i-1,i} \otimes w'_{m_i-1,i} \otimes u'_{m_i-1,i} \otimes w_{m_i-1,i} \otimes \bar{\alpha}_{j_{m_i-1,i}}^{-p_{m_i}}) \\ \cdot (u_{m_i,i} \otimes w'_{m_i,i} \otimes u'_{m_i,i} \otimes w_{m_i,i} \otimes \bar{R}_i(v_i))$$

où $\forall k \in [m_i]$

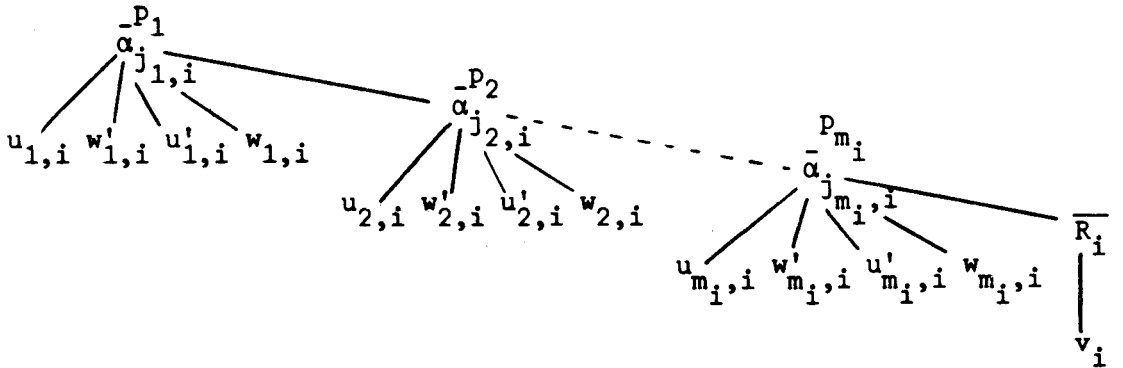
- $u_{k,i} \in T(\Sigma \cup V)_{\circ}^{p_k-1}$
 - $u'_{k,i} \in T(\Sigma \cup V)_{\circ}^{d(\alpha_{j_k})-p_k}$
- } éventuellement vides
- $w_{k,i}$ et $w'_{k,i} \in T(\Sigma \cup V \cup \bar{\Sigma})_{\circ}^1$
 - $\bar{R}_i \in \bar{R}$
 - $v_i \in T(\Sigma \cup V)_{\circ}^1$ éventuellement vide

Soit de façon schématique

$T_1 =$



et $-U_i =$



On montrerait aisément que K_1 est reconnaissable en construisant un automate d'états finis d'arbres dont les états distingueraient la lecture des lettres de $\Sigma \cup V$, \bar{R} et $\bar{\Sigma}$, $\bar{\Sigma}, D_{\Sigma}$ et \neq .

3) Homomorphismes ϕ_1

ϕ_1 est défini sur Δ_1 et à image dans

$$\Delta_2 = \Sigma \cup V \cup \{\mathcal{B}, \mathcal{B}', \neq\}$$

où \mathcal{B} , \mathcal{B}' et \neq sont de nouvelles lettres de degré respectif 3, 2 et 2.

$$* \forall i \leq |R|$$

$$\phi_1 \left(\begin{array}{c} \overline{R}_i \\ / \quad \backslash \\ 1 \quad d_i \end{array} \right) = \mathcal{B}'(r_i \otimes \ell_i) \langle d_i ; \text{Id}_{d_i}, \text{Id}_{d_i} \rangle$$

où Id_{d_i} représente la torsion identité $\langle 1, 2, \dots, d_i \rangle$

$$* \forall k \leq |\overline{R}|$$

$$\phi_1(\overline{R}_k) = \begin{array}{c} \mathcal{B}' \\ / \quad \backslash \\ r_k \quad S \end{array} \text{ avec } R_k : S \rightarrow r_k \in G.$$

$$* \phi_1 \left(\begin{array}{c} \neq \\ / \quad | \quad \backslash \\ 1 \quad 2 \quad 3 \end{array} \right) = \neq(\text{Id}_1 \otimes \neq) \cdot (\text{Id}_2 \otimes \neq) \langle 4 ; 1, 2, 2, 3 \rangle$$

$$* \forall \neq_a \in D_\Sigma, a \in \Sigma_n$$

$$\phi_1 \left(\begin{array}{c} \neq_a \\ / \quad | \quad \backslash \\ 1 \quad n+1 \end{array} \right) = \neq_a(\text{Id}_n \otimes \neq) (\text{Id}_n \otimes a \otimes \text{Id}_1) \langle \text{Id}_n, \text{Id}_{n+1} \rangle$$

d'où

$$\begin{array}{ccc} \begin{array}{c} \neq_a \\ / \quad \backslash \\ T' \quad W \end{array} & \xrightarrow{\phi_1} & \begin{array}{c} \neq_a \\ / \quad \backslash \\ T' \quad \neq \\ \quad \quad T \quad \phi(W) \end{array} \end{array}$$

$$- \forall \overline{\alpha}^p \in \overline{\Sigma}_{n+2}, \forall n \text{ tel que } \alpha \in \Sigma_n$$

$$\phi_1 \left(\begin{array}{c} \overline{\alpha}^p \\ / \quad | \quad \backslash \\ 1 \quad 2 \quad n+2 \end{array} \right) = \mathcal{B}(\alpha \otimes \mathcal{B} \otimes \alpha) \langle n+2 ; \text{Id}_n, p, n+2, n+1, \alpha_p \cdot n+1 \rangle$$

$$\text{où } \alpha_p \cdot n+1 = \alpha(\text{Id}_{p-1} \otimes n+1 \otimes \text{Id}_{n-p})$$

$$- \phi_1 \text{ est l'identité sur } \Sigma \cup V.$$

Par définition ϕ_1 est

- complet

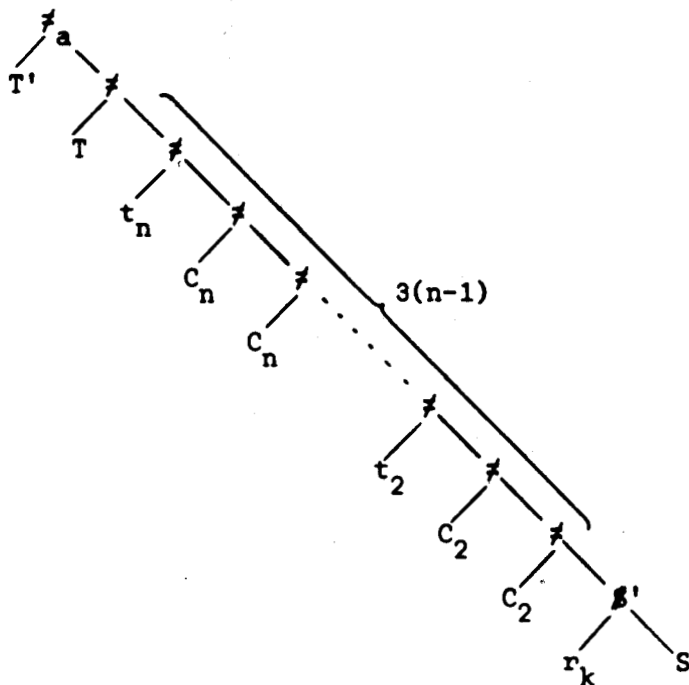
- strict

- non linéaire

- injectif sur K_1

4) Image $\phi_1(K_1)$

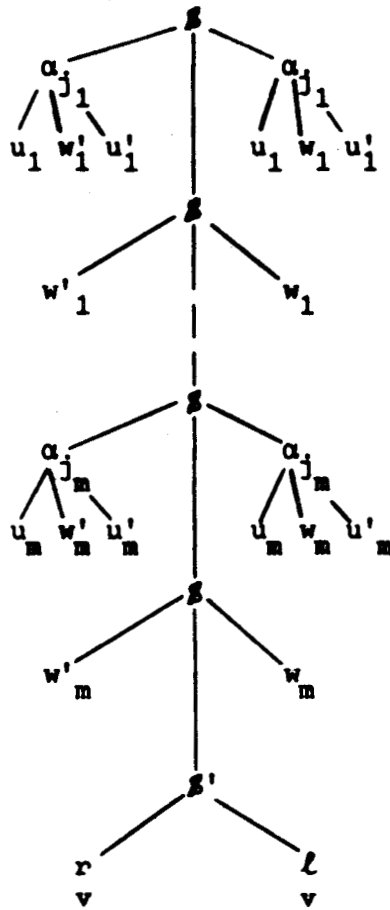
Les arbres $\phi_1(T_1)$ images des T_1 de K_1 ont alors la forme suivante :



où $\forall i, 2 \leq i \leq n, t_i$ a la forme :

$$t = \beta(\alpha_{j_1}(u_1 w'_1 u'_1) \otimes \beta \otimes \alpha_{j_1}(u_1 w_1 u'_1)) \cdot (w'_1 \otimes \beta \otimes w_1) \\ \dots \dots \dots (\alpha_{j_m}(u_m w'_m u'_m) \otimes \beta \otimes \alpha_{j_m}(u_m w_m u'_m)) \\ (w'_m \otimes \beta' \otimes w_m) \cdot (r(v), l(v))$$

soit schématiquement



la branche principale étant composée de $2n$ " g " et un symbole " g'' ".

Définition :

On appellera K'_1 la forêt constituée par les sous-arbres t_i de $\phi_1(K_1)$ dont les sommets \neq occurent à une distance égale à 2 modulo 3 du sommet \neq_a des arbres $\phi_1(T_1)$.

5) Définition de la forêt reconnaissable K_2

K_2 est définie sur l'alphabet

$$\Delta_3 = \Sigma \cup V \cup \{\delta, \delta', \varepsilon, \varepsilon'\} \cup D_\Sigma$$

et ses arbres ont la forme

$$T_2 = \#_a(T' \otimes \delta) (W'_n \otimes U_n \otimes W_n \otimes \delta) \dots (W'_2 \otimes U_2 \otimes W_2 \otimes \delta') (W'_1 \otimes S)$$

où

$$- \forall i, 2 \leq i \leq n$$

$$W_i \text{ et } W'_i \text{ et } W'_1 \in T(\Sigma \cup V)_0^1$$

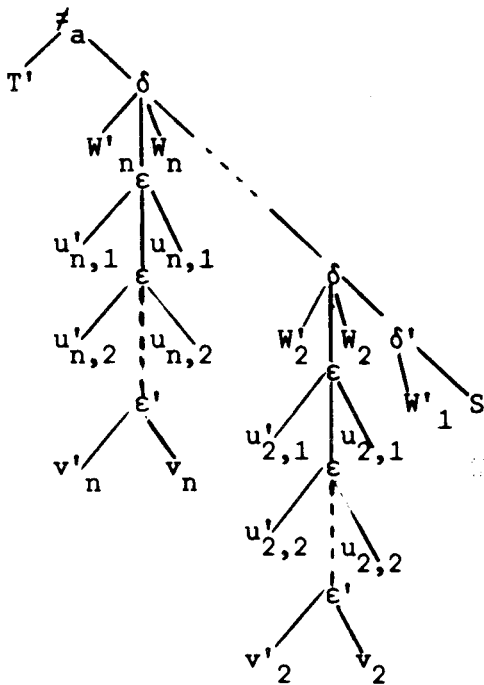
$$- T' \in T(\Sigma)_0^n \text{ tel que } a.T' = T \in F$$

- et U_i de la forme

$$U = \varepsilon(u'_1 \otimes \varepsilon \otimes u_1)(u'_2 \otimes \varepsilon \otimes u_2) \dots (u'_p \otimes \varepsilon' \otimes u_p) (v', v)$$

$$\text{avec } v, v' \in T(\Sigma \cup V)_0^1$$

soit schématiquement



K_2 est manifestement reconnaissable et la construction de l'automate d'arbres qui la reconnaît est laissée au lecteur.

6) Homomorphisme ϕ_2

ϕ_2 est défini sur Δ_3 et à image dans Δ_2 .

$$* \phi_2 \left(\begin{array}{cccc} & \delta & & \\ // & & \backslash & \\ 1 & 2 & 3 & 4 \end{array} \right) = \#(Id_1 \otimes \#)(Id_1 \otimes \# \otimes \#) \langle 6 ; 1, Id_3, 3, 4 \rangle$$

$$* \phi_2 \left(\begin{array}{c} \delta' \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) = \#(\text{Id}_1 \otimes \beta') \langle \text{Id}_1, \text{Id}_2 \rangle$$

$$* \phi_2 \left(\begin{array}{c} \varepsilon \\ / \quad | \quad \backslash \\ 1 \quad 2 \quad 3 \end{array} \right) = \beta(\text{Id}_1 \otimes \beta \otimes \text{Id}_1) \langle 1, \text{Id}_3, 3 \rangle$$

$$* \phi_2 \left(\begin{array}{c} \varepsilon' \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) = \beta(\text{Id}_1 \otimes \beta' \otimes \text{Id}_1) \langle 1, \text{Id}_2, 2 \rangle$$

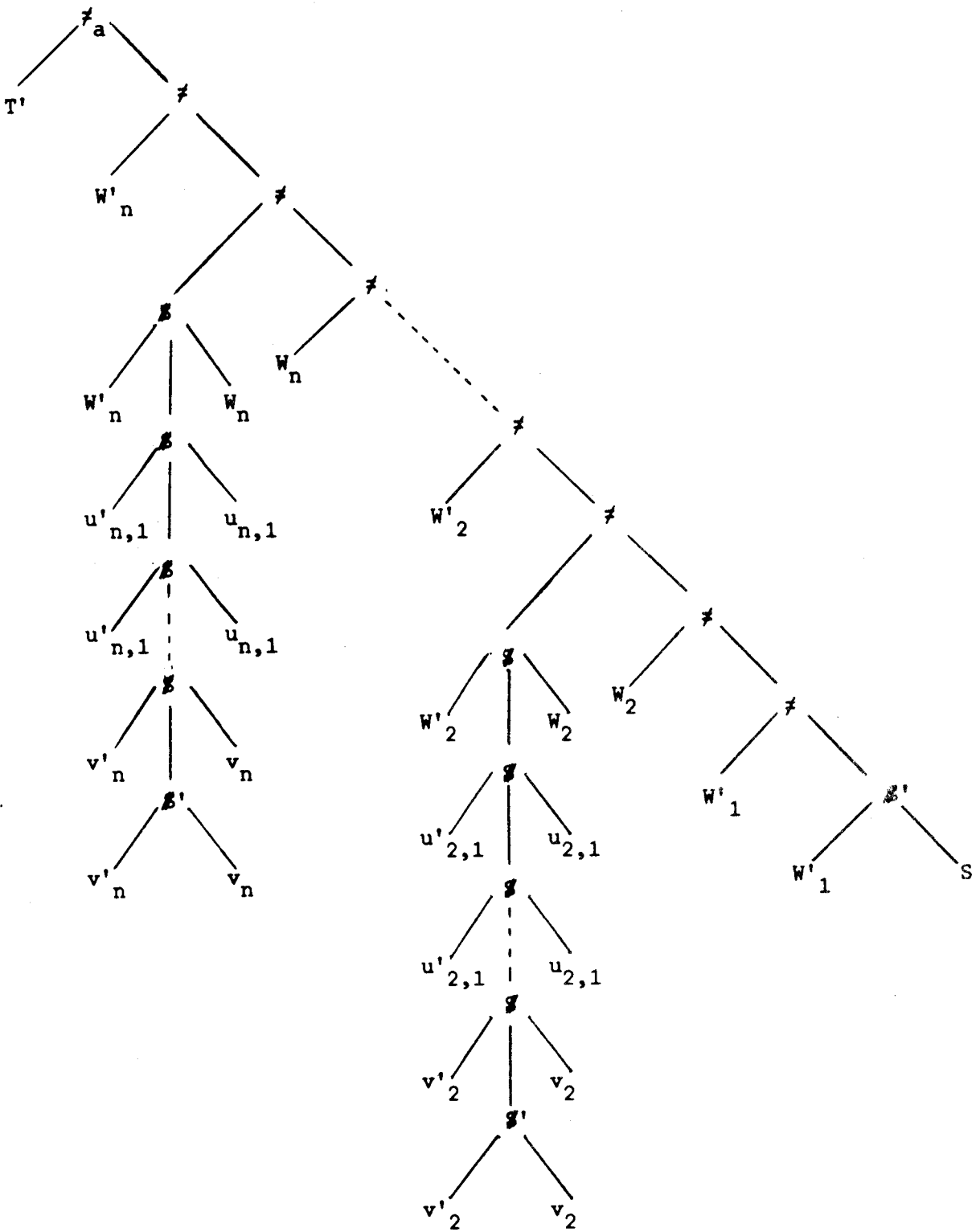
* ϕ_2 est l'identité sur $\Sigma \cup V \cup D_\Sigma$.

ϕ_2 est par définition :

- complet
- strict
- non linéaire
- injectif.

7) Image $\phi_2(K_2)$

On obtient des arbres dont la branche principale comporte comme pour $\phi_1(T_1)$ des symboles " $\#$ " au nombre de $3n-2$, un symbole " $\#_a$ " et un symbole " β " et de la forme :



Définition :

On appellera K'_2 la forêt constituée par les sous-arbres t_i de $\phi_2(K_2)$ dont les sommets \neq occurent à une distance égale à 2 modulo 3 du sommet \neq_a des arbres $\phi_2(T_2)$.

8) Démarquage π :

π est défini sur l'alphabet Δ_2 par

- identité sur $\Sigma \cup V \cup \{\beta, \beta', \neq\}$
- $\forall \neq_a \in D_\Sigma$ (tel que $a \in \Sigma_n$)

$$\pi \left(\begin{array}{c} \neq_a \\ / \quad \backslash \\ 1 \quad n \quad n+1 \end{array} \right) = \begin{array}{c} a \\ / \quad \backslash \\ 1 \quad n \end{array}$$

Par définition π est un démarquage strict qui à toute lettre de Δ_2 associe une lettre de Δ_2/D_Σ .

En particulier on aura

$$\pi[\phi_1(T_1) \cap \phi_2(T_2)] = a.T' = T.$$

Remarque :

La simulation des dérivations dans G par l'intermédiaire de ϕ_1 et ϕ_2 nous oblige à conserver toutes les informations c'est à dire à choisir ϕ_1 et ϕ_2 complets de façon à n'effacer aucune donnée avant l'étape ultime du démarquage qui ne garde que les arbres engendrés T de F. On peut faire ici un parallèle avec la théorie des langages où la nécessité de la complétude a le même rôle dans les simulations de dérivations de mots.

Ces définitions étant posées nous allons prouver le théorème en deux étapes par un lemme et son corollaire d'abord.

Lemme I.2

$t \in K'_1 \cap K'_2 \Leftrightarrow$

t est de la forme

$$\beta(\beta'_1 \otimes \beta \otimes \beta_1) \cdot (\beta'_2 \otimes \beta \otimes \beta_2) (\beta'_2 \otimes \beta \otimes \beta_2) \dots (\beta'_n \otimes \beta \otimes \beta_n) (\beta'_n \otimes \beta \otimes \beta_n) \\ \cdot (\beta'_{n+1} \otimes \beta' \otimes \beta_{n+1}) (\beta'_{n+1} \otimes \beta_{n+1})$$

avec

- $\forall i \in [n]$

$$\beta_i = \alpha_{j_i}(u'_i, \beta_{i+1}, u_i)$$

$$\beta'_i = \alpha_{j_i}(u'_i, \beta'_{i+1}, u_i)$$

- $\beta_{n+1} = \ell v$ avec $\ell \rightarrow r \in R$ et $v \in T(\Sigma \cup V)_0^1$

$$\beta'_{n+1} = rv$$

- $\forall i \in [n+1]$

$$\beta_i \Rightarrow \beta'_i$$

Preuve du lemme I.2 (par induction)

\Rightarrow .

$t \in K'_1 \Leftrightarrow$

t de la forme

$$U = \beta(\alpha_{j_1}(u_1, w'_1, u'_1) \otimes \beta \otimes \alpha_{j_1}(u_1, w_1, u'_1)) \cdot (w_1 \otimes \beta \otimes w'_1) \\ \dots (\alpha_{j_n}(u_n, w'_n, u'_n) \otimes \beta \otimes \alpha_{j_n}(u_n, w_n, u'_n)) \cdot (w'_n \otimes \beta' \otimes w_n)(r(v), \ell(v))$$

$t \in K'_2 \Leftrightarrow$

t de la forme

$$U' = \beta(\beta'_1 \otimes \beta \otimes \beta_1) (\beta'_2 \otimes \beta \otimes \beta_2) (\beta'_2 \otimes \beta \otimes \beta_2) \dots \\ \dots (\beta'_n \otimes \beta \otimes \beta_n) (\beta'_n \otimes \beta \otimes \beta_n) (\beta'_{n+1} \otimes \beta' \otimes \beta_{n+1}) (\beta'_{n+1}, \beta_{n+1})$$

avec U et U' de branche principale longue de $2n$ symboles " β " et un symbole " β' " au rang $2n+1$.

Par identification des branches de β' (en rang impair) on a

$$\beta_{n+1} = \mathcal{L}(v)$$

$$\beta'_{n+1} = r(v) \text{ avec } \mathcal{L} \rightarrow r \in R \text{ et } \beta_{n+1} \stackrel{G}{\Rightarrow} \beta'_{n+1}$$

Au " β " précédent de rang pair on a

$$\beta_{n+1} = w_n = \mathcal{L}(v)$$

$$\beta'_{n+1} = w'_n = r(v)$$

puis au rang impair précédent on a

$$\beta_n = \alpha_{j_n}(u_n, w_n, u'_n)$$

$$\text{or } w_n = \beta_{n+1}$$

$$\text{d'où } \beta_n = \alpha_{j_n}(u_n, \beta_{n+1}, u'_n)$$

et de même

$$\beta'_n = \alpha_{j_n}(u_n, w'_n, u'_n)$$

$$\text{or } w'_n = \beta'_{n+1}$$

$$\text{d'où } \beta'_n = \alpha_{j_n}(u_n, \beta'_{n+1}, u'_n)$$

et par conséquent $\beta_n \Rightarrow \beta'_n$.

Supposons l'hypothèse vérifiée jusqu'à un rang impair k c'est à dire :

$$\beta_k = \alpha_{j_k}(u_k, \beta_{k+1}, u'_k)$$

$$\beta'_k = \alpha_{j_k}(u_k, \beta'_{k+1}, u'_k)$$

et $\beta_k \Rightarrow \beta'_k$.

Par passage au rang pair précédent on a

$$\beta_k = w_{k-1}$$

$$\beta'_k = w'_{k-1}$$

puis au rang impair précédent

$$\beta_{k-1} = \alpha_{j_{k-1}}(u_{k-1}, w_{k-1}, u'_{k-1})$$

$$\beta'_{k-1} = \alpha_{j_{k-1}}(u_{k-1}, w'_{k-1}, u'_{k-1})$$

or $w_{k-1} = \beta_k$, $w'_{k-1} = \beta'_k$ et $\beta_k \Rightarrow \beta'_k$

D'où

$$\beta_{k-1} = \alpha_{j_{k-1}}(u_{k-1}, \beta_k, u'_{k-1})$$

$$\beta'_{k-1} = \alpha_{j_{k-1}}(u_{k-1}, \beta'_k, u'_{k-1})$$

et $\beta_{k-1} \Rightarrow \beta'_{k-1}$

La propriété est donc vraie et en particulier pour le 1er symbole \mathcal{B} de rang impair on a

$$\beta_1 = \alpha_{j_1}(u_1, w_1, u'_1) = \alpha_{j_1}(u_1, \beta_2, u'_1)$$

$$\beta'_1 = \alpha_{j_1}(u_1, w'_1, u'_1) = \alpha_{j_1}(u'_1, \beta'_2, u'_1)$$

avec $\beta_1 \Rightarrow \beta'_1$.

\Leftarrow .

Réciproquement soit t de la forme

$$\mathcal{B}(\beta'_1 \otimes \mathcal{B} \otimes \beta_1)(\beta'_2 \otimes \mathcal{B} \otimes \beta_2)(\beta'_2 \otimes \mathcal{B} \otimes \beta_2) \dots$$

$$\dots (\beta'_n \otimes \mathcal{B} \otimes \beta_n)(\beta'_n \otimes \mathcal{B} \otimes \beta_n)(\beta'_{n+1} \otimes \mathcal{B}' \otimes \beta_{n+1})(\beta'_{n+1}, \beta_{n+1})$$

et vérifiant les conditions du lemme.

On a immédiatement $t \in K'_2$

D'autre part

$$\begin{cases} \beta_{n+1} = \ell(v) \\ \beta'_{n+1} = r(v) \end{cases} \quad \text{avec } \ell + r \in R$$

$$\begin{cases} \beta_i = \alpha_{j_i}(u_i, \beta_{i+1}, u'_i) \\ \beta'_i = \alpha_{j_i}(u_i, \beta'_{i+1}, u'_i) \end{cases}$$

permettent de réécrire t sous la forme

$$\mathcal{B}(\alpha_{j_1}(u_1, \beta'_2, u'_1) \otimes \mathcal{B} \otimes \alpha_{j_1}(u_1, \beta_2, u'_1))(\beta'_2 \otimes \mathcal{B} \otimes \beta_2) \dots$$

$$\dots (\alpha_{j_n}(u_n, \beta'_{n+1}, u'_n) \otimes \mathcal{B} \otimes \alpha_{j_n}(u_n, \beta_{n+1}, u'_n))(\beta'_{n+1} \otimes \beta_{n+1})(r(v), \ell(v))$$

soit $t \in K'_1$ et par conséquent $t \in K'_1 \cap K'_2$.

Corollaire I.3

$\beta \xRightarrow{G} \gamma$ (où β non réduit à S)

\Leftrightarrow

$\exists t \in K'_1 \cap K'_2$ tel que $t = \mathcal{B}(\gamma, \mathcal{B}(v), \beta)$

Preuve du corollaire I.3

$\beta \xRightarrow{G} \gamma \Leftrightarrow \exists R : \ell \rightarrow r : \beta = u\ell v \Rightarrow \gamma = urv.$

Soit $\alpha_1 \alpha_2 \dots \alpha_n$ la sous branche menant du sommet de u jusqu'à ℓ dans β (ou r dans γ).

On écrit :

$$u = \alpha_1(w_1 \otimes \alpha_2 \otimes w'_1)(w_2 \otimes \alpha_3 \otimes w'_2) \dots (w_n \otimes 1 \otimes w'_n) \langle 1 \rangle$$

Alors il faut et il suffit de construire $t \in K'_1 \cap K'_2$ tel que :

$$t = \mathcal{B}(\beta'_1 \otimes \mathcal{B} \otimes \beta_1)(\beta'_2 \otimes \mathcal{B} \otimes \beta_2)(\beta'_2 \otimes \mathcal{B} \otimes \beta_2) \dots \\ \dots (\beta'_n \otimes \mathcal{B} \otimes \beta_n)(\beta'_{n+1} \otimes \mathcal{B}' \otimes \beta_{n+1})(\beta'_{n+1}, \beta_{n+1})$$

avec $\beta_{n+1} = \ell(v)$

$$\beta'_{n+1} = r(v)$$

et $\forall i \in [n]$

$$\beta_i = \alpha_i(w_i, \beta_{i+1}, w'_i)$$

$$\beta'_i = \alpha_i(w_i, \beta'_{i+1}, w'_i)$$

$$\beta_i \Rightarrow \beta'_i$$

soit donc $\beta_1 = \beta$ et $\beta'_1 = \gamma$.

Le lemme précédent entraîne immédiatement l'équivalence annoncée.

Preuve du théorème I.1

\Rightarrow .

La preuve se fait en deux étapes :

1) D'après le corollaire précédent on sait associer à chaque dérivation $u_j \ell_{i_j} v_j \Rightarrow u_j r_{i_j} v_j$ un arbre $t_j \in K'_1 \cap K'_2$. Il nous reste alors à montrer qu'il existe deux arbres t_j^1 et t_j^2 tels que

$$t_j = \phi_1(t_j^1) = (u_j r_{i_j} v_j, \phi_2(t_j^2), u_j \ell_{i_j} v_j)$$

2) En construisant T_1 et T_2 à partir de t_j^1 et t_j^2 associés à chaque dérivation de rang j , il faut encore vérifier que

$$\phi_1(T_1) \cap \phi_2(T_2) \neq \emptyset$$

$$\text{et } \pi[\phi_1(T_1) \cap \phi_2(T_2)] = T \in F.$$

lère étape

Considérons la séquence

$$S \xrightarrow{R_{i_1}} r_{i_1} = u_2 \ell_{i_2} v_2 \xrightarrow{R_{i_2}} u_2 r_{i_2} v_2 \dots \Rightarrow u_{n-1} r_{i_{n-1}} v_{n-1} = u_n \ell_{i_n} v_n \xrightarrow{R_{i_n}} u_n r_{i_n} v_n = T$$

utilisant dans l'ordre les règles $R_{i_1} R_{i_2} \dots R_{i_n}$.

Le corollaire précédent entraîne :

$$\forall j \in [n], j \neq 1$$

$$(u_j \ell_{i_j} v_j \Rightarrow u_j r_{i_j} v_j) \Leftrightarrow$$

$$\exists t_j \in K'_1 \cap K'_2 \text{ tel que}$$

$$\text{si } u_j = \alpha_{1,j}(w_{1,j} \otimes \alpha_{2,j} \otimes w'_{1,j}) \dots (w_{n_j,j} \otimes 1 \otimes w'_{n_j,j}) \langle 1 \rangle$$

$$\text{alors } t_j = \beta(\beta'_{1,j} \otimes \beta \otimes \beta_{1,j})(\beta'_{2,j} \otimes \beta \otimes \beta_{2,j})(\beta'_{2,j} \otimes \beta \otimes \beta_{2,j}) \dots$$

$$\dots (\beta'_{n_j,j} \otimes \beta \otimes \beta_{n_j,j})(\beta'_{n_j+1,j} \otimes \beta \otimes \beta_{n_j+1,j})(\beta'_{n_j+1,j}, \beta_{n_j+1,j})$$

où

$$- \beta_{n_j+1,j} = \ell_{i_j}(v_j)$$

$$\beta'_{n_j+1,j} = r_{i_j}(v_j)$$

$$- \forall k \in [n_j]$$

$$\beta_{k,j} = \alpha_{k,j}(w_{k,j}, \beta_{k+1,j}, w'_{k,j})$$

$$\beta'_{k,j} = \alpha_{k,j}(w_{k,j}, \beta'_{k+1,j}, w'_{k,j})$$

$$- \forall k \in [n_j+1]$$

$$\beta_{k,j} \Rightarrow \beta'_{k,j}$$

Considérons alors pour chaque $\beta_{k,j} = \alpha_{k,j}(w_{k,j}, \beta_{k+1,j}, w'_{k,j})$ le rang p_k de $\alpha_{k,j}$ où apparaît $\beta_{k+1,j}$ (soit encore la co-arité $p_k - 1$ de $w_{k,j}$). On peut alors associer à $\alpha_{k,j}$ un symbole $\bar{\alpha}_{k,j}^{p_k}$ de façon unique.

De même si $\beta_{n+1,j} = \ell_{i_j}(v_j) \Rightarrow \beta'_{n+1} = r_{i_j}(v_j)$ avec la règle $R_{i_j} : \ell_{i_j} + r_{i_j}$ on sait lui associer la lettre \bar{R}_{i_j} de \bar{R} correspondante.

Soit alors t_j^1 l'arbre défini par :

$$t_j^1 = \bar{\alpha}_{1,j}^{-p_1}(w_{1,j} \otimes \beta'_{2,j} \otimes w'_{1,j} \otimes \beta_{2,j} \otimes \bar{\alpha}_{2,j}^{-p_2} \\ \cdot (w_{2,j} \otimes \beta'_{3,j} \otimes w'_{2,j} \otimes \beta_{3,j} \otimes \bar{\alpha}_{3,j}^{-p_3}) \dots \\ \dots (w_{n_j-1} \otimes \beta'_{n_j,j} \otimes w'_{n_j-1,j} \otimes \beta_{n_j,j} \otimes \bar{\alpha}_{n_j,j}^{-p_{n_j}}) \\ \cdot (w_{n_j,j} \otimes \beta_{n_j+1,j} \otimes w'_{n_j,j} \otimes \beta_{n_j+1,j} \otimes \bar{R}_{i_j}(v_j))$$

On a immédiatement $\phi_1(t_j^1) = t_j$.

De même considérons t_j^2 défini par :

$$t_j^2 = \varepsilon(\beta'_{2,j} \otimes \varepsilon \otimes \beta_{2,j})(\beta'_{3,j} \otimes \varepsilon \otimes \beta_{3,j}) \dots \\ \dots (\beta'_{n_j,j} \otimes \varepsilon \otimes \beta_{n_j,j})(r_{i_j}(v_j), \ell_{i_j}(v_j))$$

où

$$- \forall k \in [n_j]$$

$$\beta_{k,j} = \alpha_{k,j}(w_{k,j}, \beta_{k+1,j}, w'_{k,j})$$

$$\beta'_{k,j} = \alpha_{k,j}(w_{k,j}, \beta'_{k+1,j}, w'_{k,j})$$

Alors on a $\phi_2(t_j^2)$ tel que

$$t_j = (u_j r_{i_j} v_j, \phi_2(t_j^2), u_j \ell_{i_j} v_j) = \phi_1(t_j^1).$$

Remarquons que la définition de ϕ_1 et ϕ_2 entraînent l'unicité de t_j^1 et t_j^2 .

2ème étape :

Nous construisons maintenant T_1 et T_2 de K_1 et K_2 à partir des règles R_{i_k} utilisées dans la séquence de dérivation :

$$* T_1 = \neq_a (T' \otimes \neq) (t_n^1 \otimes s_n \otimes \neq) \dots (t_2^1 \otimes s_2 \otimes \bar{R}_{i_1})$$

où

$$\begin{aligned} & - \bar{R}_{i_1} \text{ est associé à } R_{i_1} : S \rightarrow r_{i_1} \\ & \text{tel que } \phi_1(\bar{R}_{i_1}) = \neq'(r_{i_1}, S) \end{aligned}$$

$$- \forall k \in [n], k \neq 1$$

t_k^1 utilise la règle R_{i_k} et vérifie selon le procédé de la 1ère étape

$$\phi_1(t_k^1) = t_k$$

$$- \forall k \in [n], k \neq 1$$

$s_k = u_k \ell_{i_k} v_k$ utilisant le membre gauche de la règle R_{i_k} tel que

$$\phi(s_k) = s_k.$$

$$- T' \text{ vérifie } T = a.T' \in F.$$

Soit $s_1 = S = u_1 \ell_{i_1} v_1$ avec u_1 et v_1 vides et $t_1 = \phi_1(\bar{R}_{i_1})$.

$$* T_2 = \neq_a (T' \otimes \delta) (W'_n \otimes t_n^2 \otimes W_n \otimes \delta) \dots (W'_2 \otimes t_2^2 \otimes W_2 \otimes \delta) (W'_1, S)$$

où

$$- T' \text{ vérifie } T = a.T' \in F$$

$$- \forall k \in [n], k \neq 1$$

t_k^2 est défini par le procédé de la 1ère étape à partir de la règle R_{i_k}

et vérifie

$$t_j = (u_j r_{i_j} v_j, \phi_2(t_j^2), u_j \ell_{i_j} v_j)$$

$$- \forall k \in [n], k \neq 1$$

$$W_k = u_k \ell_{i_k} v_k$$

$$W'_k = u_k r_{i_k} v_k$$

Soit $W'_1 = u_1 r_{i_1} v_1$ avec u_1 et v_1 vides.

$$\text{et } W_1 = u_1 S v_1.$$

Alors en appliquant ϕ_1 à T_1 et ϕ_2 à T_2 on obtient

$$\phi_1(T_1) = \neq_a(T' \otimes \neq)(T \otimes \neq)(t_n \otimes \neq)(s_n \otimes \neq)(s_n \otimes \neq) \\ \dots (t_2 \otimes \neq)(s_2 \otimes \neq)(s_2 \otimes \neq')(s_1, S)$$

$$\phi_2(T_2) = \neq_a(T' \otimes \neq)(W'_n \otimes \neq)(\beta(W'_n, \phi_2(t_n^2), W_n) \otimes \neq)(W_n \otimes \neq) \dots \\ \dots (W'_2 \otimes \neq)(\beta(W'_2, \phi_2(t_2^2), W_2) \otimes \neq)(W_2 \otimes \neq)(W'_1 \otimes \neq'(W'_1, W_1))$$

Or

$$- \forall k \in [n]$$

$$s_k = u_k \ell_{i_k} v_k = W_k$$

$$- \forall k \in [n], k \neq 1$$

$$t_k = \phi_1(t_k^1) = \beta(W'_k, \phi_2(t_k^2), W_k) \\ = \beta(u_k r_{i_k} v_k, \phi_2(t_k^2), u_k \ell_{i_k} v_k)$$

$$- t_1 = \neq'(r_{i_1}, S) = \neq'(W'_1, W_1)$$

soit d'après l'enchaînement des dérivations

$$- \forall k \in [n-1]$$

$$s_{k+1} = u_{k+1} \ell_{i_{k+1}} v_{k+1} = W_{k+1} = W'_k = u_k r_{i_k} v_k$$

et

$$- W'_n = T = u_n r_{i_n} v_n$$

Soit par conséquent $\phi_1(T_1) = \phi_2(T_2)$

En appliquant π qui ne conserve que $a.T'$ on obtient alors

$$T = \pi[\phi_1(T_1) \cap \phi_2(T_2)]$$

\Leftarrow .

Réciproquement soit T_1 et T_2 tels que

$$T = \pi[\phi_1(T_1) \cap \phi_2(T_2)].$$

Nécessairement $\phi_1(T_1) \cap \phi_2(T_2)$ est de la forme

$$\neq_a(T' \otimes \neq)(T \otimes \neq)(\beta(\gamma_n, D_n, \beta_n) \otimes \neq)(\beta_n \otimes \neq) \\ \dots (\gamma_2 \otimes \neq)(\beta(\gamma_2, D_2, \beta_2) \otimes \neq)(\beta_2 \otimes \neq')(\gamma_1, \beta_1)$$

avec

$$- \forall k \in [n-1]$$

$\gamma_k = \beta_{k+1}$ par application de ϕ_1

- $\forall k \in [n]$

$\beta_k \Rightarrow \gamma_k$ par le corollaire précédent

- $\beta_1 = S \Rightarrow \gamma_1 = r_{i_1}$

- $\gamma_n = T$ par application de ϕ_2

d'où la dérivation

$$S \Rightarrow \gamma_1 = r_{i_1} = \beta_2 = u_2 \ell_{i_2} v_2 \Rightarrow \gamma_2 = u_2 r_{i_2} v_2 \dots \Rightarrow \gamma_n = T$$

C.Q.F.D.

Corollaire I.4

Pour toute forêt F récursivement énumérable,

il existe une forêt reconnaissable K_2

un morphisme ϕ_2 complet strict

et un bimorphisme (ϕ_1, K_1, Ψ) où ϕ_1 est complet strict et Ψ strict

tels que

$$F = [\phi_2 \circ (\phi_1, K_1, \Psi)] (K_2)$$

Preuve du corollaire I.4

Le théorème précédent nous donne $\exists \phi_1, \phi_2, K_1, K_2$ et π tels que

$$F = \pi[\phi_1(K_1) \cap \phi_2(K_2)]$$

Utilisons la propriété ensembliste

$$\phi_1(K_1) \cap \phi_2(K_2) = \phi_1[K_1 \cap \phi_1^{-1}(\phi_2(K_2))]$$

On obtient

$$F = \pi \cdot \phi_1[K_1 \cap \phi_1^{-1}(\phi_2(K_2))]$$

ou
$$F = \Psi[K_1 \cap \phi_1^{-1}(\phi_2(K_2))]$$

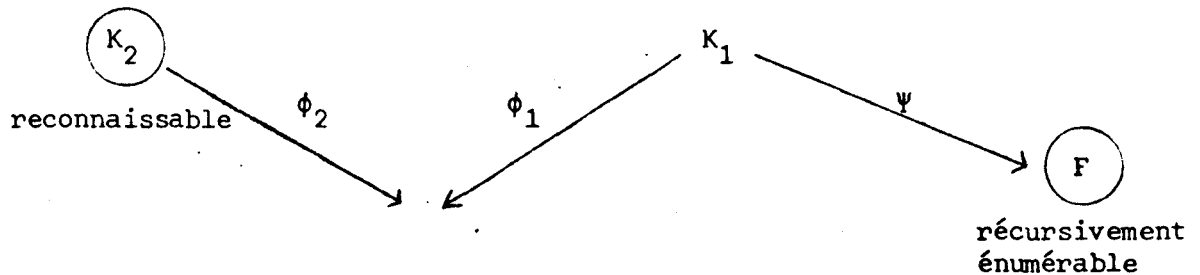
ou
$$F = (\phi_2 \circ B) (K_2)$$

en posant $\Psi = \phi_1 \circ \pi$

et $B = (\phi_1, K_1, \Psi)$

Nous avons ϕ_1 et ϕ_2 complets stricts d'après le théorème précédent et π strict entraîne alors $\Psi = \phi_1 \circ \pi$ est strict.

Nous avons le schéma suivant :



Corollaire I.5

Il est indécidable de savoir si pour ϕ_1 et ϕ_2 complets et K_1 et K_2 reconnaissables, $\phi_1(K_1) \cap \phi_2(K_2) = \emptyset$.

Preuve du corollaire I.5

Soit F récursivement énumérable et le problème d'appartenance indécidable suivant : étant donné t , t appartient-il à F ?

Le théorème nous donne

$$F = \pi[\phi_1(K_1) \cap \phi_2(K_2)]$$

On peut formuler le problème d'appartenance ainsi :

$$\pi^{-1}(t) \cap (\phi_1(K_1) \cap \phi_2(K_2)) \stackrel{?}{=} \emptyset$$

$$\text{ou } \phi_1[\phi_1^{-1}(\pi^{-1}(t)) \cap K_1] \cap \phi_2(K_2) \stackrel{?}{=} \emptyset$$

$K_1 \cap \phi_1^{-1}(\pi^{-1}(t))$ est reconnaissable puisque la reconnaissabilité est conservée par morphisme inverse et intersection.

$$\text{Soit } K'_1 = K_1 \cap \phi_1^{-1}(\pi^{-1}(t)) \in \text{Rec}$$

Si on pouvait alors décider si

$$\phi_1(K'_1) \cap \phi_2(K_2) = \emptyset$$

on saurait décider de l'appartenance de t à F ce qui est absurde.

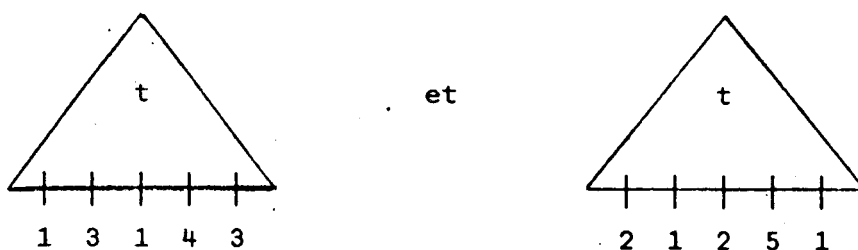
CHAPITRE II

LES FORETS RATEG

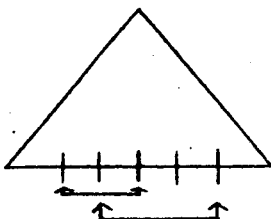
A - PRELIMINAIRES ALGEBRIQUES

Dans l'étude des forêts RATEG nous verrons que le seul rôle des torsions est d'imposer des égalités de sous-arbres et ce, indépendamment de la numérotation choisie pour les variables.

Ainsi les deux écritures



seront équivalentes, une classe d'équivalence pouvant être schématisée par



et les objets que nous manipulerons seront en fait de telles classes.

Nous formalisons mathématiquement cette idée en définissant une relation d'équivalence - notée \equiv - sur les arbres de $T(\Sigma)$.

Cette relation \equiv est l'équivalence associée au préordre sur $T(\Sigma)$ "est un sous arbre au sommet de" et noté $<$.

On dira que t est un sous arbre au sommet de t' ou $t < t'$ si il existe u tel que $t.u = t'$. La notion de sous-arbre joue un rôle essentiel dans notre étude mais $<$ n'est qu'un préordre sur $T(\Sigma)$ car si $t < t'$ et $t' < t$ c'est que justement t et t' ne diffèrent que par les numérotations de leurs torsions.

Nous pouvons alors définir les bornes supérieure (notée \vee) et inférieure (notée \wedge) sur l'ensemble quotient $T(\Sigma)_{/\equiv}$ qui est nous le verrons un inf-demi-treillis.

La notion d'unifié - au sens classique du terme - se confond alors ici avec celle de borne supérieure et l'unification est à la base de la construction de l'intersection de deux RATEG et jouera également un rôle important dans la recherche de l'image par morphisme de forêt RATEG.

Remarque :

Il importe de noter que l'invariance par \equiv est due justement au fait que les RATEG testent des égalités mais ne modifient pas l'ordre des variables. Notons alors qu'il n'y a pas compatibilité des morphismes avec \equiv comme le montre l'exemple suivant :

$$\text{soit } \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) \rightarrow \begin{array}{c} b \\ / \quad \backslash \\ 1 \quad 2 \end{array}$$

$$\text{et } \psi \left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) \rightarrow \begin{array}{c} b \\ / \quad \backslash \\ 2 \quad 1 \end{array}$$

Ces deux morphismes sont strictement différents bien que l'on ait

$$\begin{array}{c} b \\ / \quad \backslash \\ 1 \quad 2 \end{array} \equiv \begin{array}{c} b \\ / \quad \backslash \\ 2 \quad 1 \end{array}$$

I - DÉFINITION DE L'ENSEMBLE ORDONNÉ $T(\Sigma)$ / III

1) Le préordre "est un sous-arbre au sommet de"

Définition

Soit $t, t' \in T(\Sigma)$. La relation t "est un sous-arbre au sommet de" t' notée $t < t'$ est définie par :

$$t < t' \text{ ssi } \exists u \in T(\Sigma) : t.u = t'$$

Exemple II.1 : sous-arbre au sommet

$$t' = \begin{array}{c} a \\ / \quad \backslash \\ b \quad c \\ / \quad \backslash \quad | \\ 1 \quad 3 \quad 1 \end{array} \in T(\Sigma)_3^1 \quad t = \begin{array}{c} a \\ / \quad \backslash \\ 2 \quad c \\ \quad \quad | \\ \quad \quad 4 \end{array} \in T(\Sigma)_4^1$$

On a $t < t'$ avec $u = (1, \begin{array}{c} b \\ / \quad \backslash \\ 1 \quad 3 \end{array}, 3, 1) \in T(\Sigma)_3^4$

Propriété II.1

$<$ est un préordre sur $T(\Sigma)$

Preuve :

Nous vérifions en effet les propriétés de

* réflexivité :

soit $t \in T(\Sigma)_q^p$ et θ l'identité sur $[q]$

alors $t.\theta = t$.

* transitivité

soit $t < t' \Leftrightarrow \exists u : tu = t'$

et $t' < t'' \Leftrightarrow \exists v : t'v = t''$

$\Rightarrow t.(uv) = (t.u.)v = t'.v = t'' \Leftrightarrow t < t''$.

2) L'équivalence \equiv associée au préordre $<$

Nous allons voir qu'intuitivement $t \equiv t'$ ssi ils ne diffèrent que par la numérotation des variables, propriété formalisée par (P).

Définition

Soit $t = \tilde{t}.\theta \in T(\Sigma)_q^p$ et $t' = \tilde{t}'.\theta' \in T(\Sigma)_{q'}^{p'}$

alors

$$(P) \quad t \equiv t' \quad \text{ssi} \quad \left\{ \begin{array}{l} \tilde{t} = \tilde{t}' \\ \text{et il existe une bijection} \\ b : \text{Im}(\theta) \rightarrow \text{Im}(\theta') \\ \text{telle que } b\theta = \theta'.b = \theta' \end{array} \right.$$

Notation :

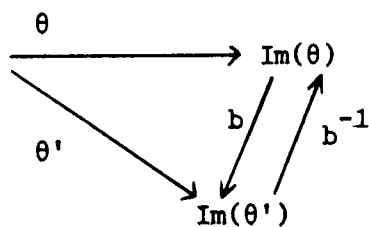
Nous adopterons la convention suivante pour noter la composition des torsions :

$$\theta.\theta' = \theta' \circ \theta$$

où \circ est la composition classique des applications.

Remarque :

b est une bijection mais non une torsion.



On remarquera que cette caractérisation de \equiv équivaut à la définition classique d'une équivalence \equiv associée à un préordre $<$:

$$t \equiv t' \quad \Leftrightarrow \quad \left\{ \begin{array}{l} t < t' \\ \text{et } t' < t \end{array} \right.$$

ce qui se traduit ici par

$$(Q) \quad t \equiv t' \quad \Leftrightarrow \quad \left\{ \begin{array}{l} \exists u : t.u = t' \\ \text{et } \exists v : t'.v = t \end{array} \right.$$

Preuve

- (P) \Rightarrow (Q)

$$t \equiv t' \Leftrightarrow \tilde{t} = \tilde{t}'$$

(P) $\exists b : \text{Im}(\theta) \rightarrow \text{Im}(\theta') : \theta.b = \theta'$

On peut étendre b en une torsion $\beta : [q] \rightarrow [q']$

telle que $\theta.\beta = \theta'$

On a alors $t = \tilde{t}.\theta$

$$t' = \tilde{t}'.\theta' = \tilde{t}.\theta.\beta = t.\beta$$

$$\Leftrightarrow t < t'$$

De même on aurait $t' < t$.

- (Q) \Rightarrow (P)

$$t = t'.v = (t.u).v = t.(uv)$$

$\Rightarrow uv \in \theta_q^q$, et la restriction de u, v à $\text{Im}(\theta)$ est l'identité.

De même la restriction de v, u à $\text{Im}(\theta')$ est l'identité.

Soit β la restriction de u à $\text{Im}(\theta)$ et

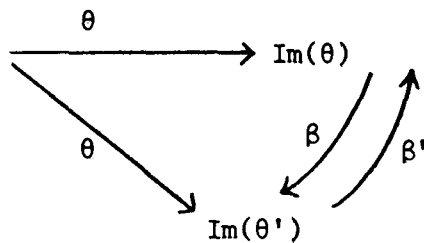
β' la restriction de v à $\text{Im}(\theta')$.

Nécessairement β et β' sont injectives.

Alors $tu = \tilde{t}.\theta' \Rightarrow \text{Im}(u) = \text{Im}(\theta')$

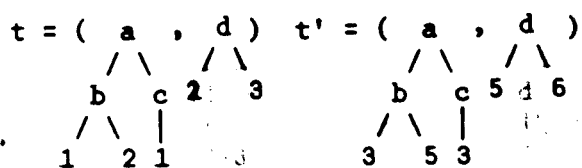
De même $\text{Im}(v) = \text{Im}(\theta)$.

β est donc une bijection et β' la bijection inverse

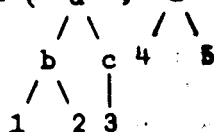


Exemple II.2 : arbres équivalents et non comparables.

a) $t \equiv t'$.



$$t = \tilde{t} \cdot \theta = (a, d) \cdot \langle 5; 1, 2, 1, 2, 3 \rangle$$



$$t' = \tilde{t}' \cdot \theta' = \tilde{t}' \cdot \langle 5; 3, 5, 3, 5, 6 \rangle$$

On a

$$t \cdot u = t' \text{ avec } u = (3, 5, 6, 4, 5) \text{ soit } t < t'$$

$$t' \cdot v = t \text{ avec } v = (1, 2, 1, 2, 3) \text{ soit } t' < t$$

La bijection b est alors définie par :

$$b : \text{Im}(\theta) = \{1, 2, 3\} \rightarrow \text{Im}(\theta') = \{3, 5, 6\}$$

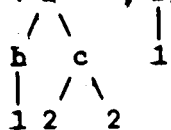
$$\text{vérifie - } b(1) = 3$$

$$\text{- } b(2) = 5$$

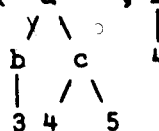
$$\text{- } b(3) = 6$$

b) $t \not\equiv t'$

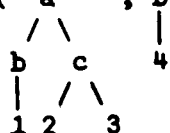
$$t = (a, b) \in T(\mathbb{E})_4^2$$



$$t' = (a, b) \in T(\mathbb{E})_5^2$$



$$\text{vérifient } t = \tilde{t} \cdot \theta = (a, b) \cdot \langle 1, 2, 2, 1 \rangle$$



$$t' = \tilde{t}' \cdot \theta' = \tilde{t}' \cdot \langle 3, 4, 5, 4 \rangle$$

mais on a $\theta(2) = \theta(3)$ et $\theta'(2) \neq \theta'(3)$ il n'y a donc pas bijection entre θ et θ' c'est à dire t et t' ne sont pas comparables.

Définition :

Nous appellerons $T(\Sigma)_{/\equiv}$ l'espace quotient des classes d'équivalence \bar{t} des arbres t définies sur $T(\Sigma)$ par la relation \equiv et nous noterons \leq l'ordre quotient canoniquement induit par $<$ sur $T(\Sigma)_{/\equiv}$:

$$t \in T(\Sigma)_q^p \quad t' \in T(\Sigma)_q^p,$$

$$\bar{t} \leq \bar{t}' \iff \exists u \in T(\Sigma)_q^p : tu \equiv t'.$$

Notation

La classe \bar{t} d'un arbre t sera généralement représentée par $\tilde{t}.\bar{\theta}$ où $\bar{\theta}$ est la torsion classique en ordre croissant

$$\bar{\theta} : [n] \rightarrow [|\text{Im } \bar{\theta}([n])|]$$

où $[n] = \{1, 2 \dots n\}$

et $[|\text{Im } \bar{\theta}([n])|] = \{1, 2 \dots |\text{Im } \bar{\theta}([n])|\}$

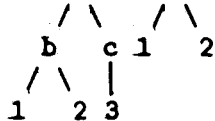
avec $|\text{Im } \bar{\theta}([n])|$ représentant le nombre des éléments images de $\{1, 2 \dots n\}$ par $\bar{\theta}$. (N.B. $|\text{Im } \bar{\theta}([n])| \leq n$).

On retrouvera cette notation dans les exemples suivants.

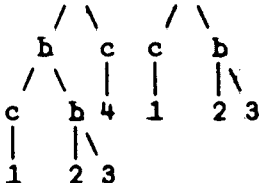
Exemple II.3 : Classes ordonnées et non comparables de $T(\Sigma)_{/\equiv}$

a) $\bar{t} \leq \bar{t}'$

$$\bar{t} = (a , d) = \tilde{t} \cdot \langle 5 ; 1, 2, 3, 1, 2 \rangle$$



$$\bar{t}' = (a , d) = \tilde{t}' \cdot \langle 7 ; 1, 2, 3, 4, 1, 2, 3 \rangle$$



En prenant $t = \bar{t}$ et $t' = \bar{t}'$ on construit facilement

$$u = (5 ; c, b, 4, 4, 5)$$

$$\begin{array}{c} | \quad / \quad \backslash \\ 1 \quad 2 \quad 3 \end{array}$$

tel que $tu = t'$ soit $tu \equiv t'$ et $\bar{t} \leq \bar{t}'$.

b) $\bar{t} \not\leq \bar{t}'$.

$$\bar{t} = (a, d) = \tilde{t}. \langle 5 ; 1, 2, 3, 1, 2 \rangle$$

$$\begin{array}{c} / \quad \backslash \quad / \quad \backslash \\ b \quad c \quad 1 \quad 2 \\ / \quad \backslash \quad | \\ 1 \quad 2 \quad 3 \end{array}$$

$$\bar{t}' = (a, d) = \tilde{t}'. \langle 6 ; 1, 2, 3, 4, 1, 2 \rangle$$

$$\begin{array}{c} / \quad \backslash \quad / \quad \backslash \\ b \quad c \quad c \quad c \\ / \quad \backslash \quad | \quad | \\ c \quad b \quad 4 \quad 1 \quad 2 \\ | \quad / \quad \backslash \\ 1 \quad 2 \quad 3 \end{array}$$

\bar{t} et \bar{t}' sont 2 classes incomparables. En effet $\forall t \in \bar{t}, \forall t' \in \bar{t}', \forall u$
 $tu \not\equiv t'$

car $t.u$ s'écrira (a, d) et ne peut vérifier $u_2 = \begin{array}{c} b \\ / \quad \backslash \\ 2 \quad 3 \end{array} = c = \begin{array}{c} c \\ | \\ 2 \end{array}$.

$$\begin{array}{c} / \quad \backslash \quad / \quad \backslash \\ b \quad c \quad u_1 \quad u_2 \\ / \quad \backslash \quad | \\ u_1 \quad u_2 \quad u_3 \end{array}$$

3) Propriétés de \equiv et \leq .

P II.2 \equiv est compatible avec le degré supérieur.

Soit $t \in T(\Sigma)_q^p, t' \in T(\Sigma)_{q'}^{p'}$

$t \equiv t' \Rightarrow p = p'$

(Ceci de façon évidente puisque $\tilde{t} = \tilde{t}' \in T(\Sigma)_r^p$)

On peut alors restreindre pour tout p l'équivalence sur $T(\Sigma)^p$ et considérer

l'espace quotient $T(\Sigma)^p / \equiv$. Par contre \equiv n'est pas compatible avec le degré inférieure : $t \equiv t' \not\Rightarrow q = q'$

(trivial car on choisit arbitrairement les variables).

P II.3 \equiv est compatible avec le produit tensoriel.

Soit $u \equiv u'$ avec $u \in T(\Sigma)_q^p$ et $u' \in T(\Sigma)_{q'}^p$,

c'est à dire $u = \tilde{u} \cdot \theta$ et $\theta \cdot b = \theta'$
 $u' = \tilde{u}' \cdot \theta'$

soit $v = \tilde{v} \cdot \mu \in T(\Sigma)_s^r$

On a $u \otimes v = (\tilde{u} \otimes \tilde{v}) \langle \theta \otimes \mu \rangle \in T(\Sigma)_{q+s}^{p+r}$
 $u' \otimes v = (\tilde{u}' \otimes \tilde{v}) \langle \theta' \otimes \mu \rangle \in T(\Sigma)_{q'+s}^{p+r}$

b est une bijection de $\text{Im}(\theta) \rightarrow \text{Im}(\theta')$.

Considérons les applications constantes q et q' . ($q < q'$) on peut alors étendre b en une bijection de $\text{Im}(\theta) \cup \text{Im}(\mu+q) \rightarrow \text{Im}(\theta) \cup \text{Im}(\mu+q')$

par : $\forall i \in \text{Im}(\mu+q)$

$$b(i) = i + (q' - q)$$

On aura donc

$$u \otimes v = \widetilde{u \otimes v} \cdot (\theta \otimes \mu)$$

$$u' \otimes v = \widetilde{u' \otimes v} \cdot (\theta' \otimes \mu)$$

et $\exists b : \text{Im}(\theta \otimes \mu) \rightarrow \text{Im}(\theta' \otimes \mu)$

tel que $(\theta \otimes \mu) \cdot b = (\theta' \otimes \mu)$

soit $u \otimes v \equiv u' \otimes v$.

Le lecteur vérifierait de même $v \otimes u \equiv v \otimes u'$.

P II.4 \equiv n'est pas compatible avec la composition à gauche.

$$\text{Soit } u = \begin{pmatrix} a & b \\ / \quad \backslash \\ 1 & 2 & 1 \end{pmatrix} \in T(\Sigma)_3^2$$

$$u' = \begin{pmatrix} a & b \\ / \quad \backslash \\ 2 & 3 & 2 \end{pmatrix} \in T(\Sigma)_3^2$$

$$\text{et } v = (v_1, v_2, v_3) \in T(\Sigma)_p^3$$

On a bien $u \equiv u'$

$$\text{mais } u.v = \left(\begin{array}{cc} a & b \\ / \ \backslash & | \\ v_1 v_2 & v_1 \end{array} \right)$$

$$u'.v = \left(\begin{array}{cc} a & b \\ / \ \backslash & | \\ v_2 v_3 & v_2 \end{array} \right)$$

et donc $u.v \not\equiv u'.v$

On ne peut donc pas munir $T(\Sigma)_{/\equiv}$ d'une structure de magmaïde.

On a cependant un lemme de compatibilité à droite avec la composition :

Lemme II.5

$$\forall u, v, v'$$

$$v \equiv v' \Rightarrow u.v \equiv u.v'$$

Preuve :

Soit $u \in T(\Sigma)_q^p$, $v \in T(\Sigma)_r^q$ et $v' \in T(\Sigma)_{r'}^q$.

$$v \equiv v' \Rightarrow \left\{ \begin{array}{l} v = \tilde{v}.\theta \\ v' = \tilde{v}.\theta' \end{array} \right. \quad \text{où } \left\{ \begin{array}{l} - \tilde{v} \in T(\Sigma)_s^q, \\ - \theta \in \Theta_r^s \text{ et } \theta' \in \Theta_{r'}^s, \\ - \text{et } \theta.b = \theta' \end{array} \right.$$

On a alors :

$$t = u.v = u.(\tilde{v}.\theta) = (u.\tilde{v}).\theta = \tilde{t}.\mu.\theta$$

$$t' = u.v' = u.(\tilde{v}.\theta') = (u.\tilde{v}).\theta' = \tilde{t}.\mu.\theta'$$

$$\text{et } \theta.b = \theta' \Rightarrow \mu.(\theta.b) = (\mu.\theta).b = (\mu.\theta')$$

on a donc par définition $t \equiv t'$

$$\text{soit } u.v \equiv u.v'.$$

Remarque

Ainsi la notation $t.\bar{u}$ a un sens et on a immédiatement $\overline{t.u} = t.\bar{u}$.

Remarque

Il importe de faire la différence entre la relation d'ordre partiel sur $\hat{T}(\Sigma)$ notée \leq "est un sous-arbre initial de" et le préordre $<$ défini sur $T(\Sigma)$.

P II.6 \equiv n'est pas compatible avec \leq .

Il suffit de noter que la notion d'arbre initial donc d'être un élément de $\hat{T}(\Sigma)$ n'est pas compatible avec \equiv par conséquent l'ordre \leq défini sur $\hat{T}(\Sigma)$ ne peut être compatible avec \equiv .

Notons toutefois que

$$t \equiv t' \text{ sur } T(\Sigma)$$

$$\Rightarrow \tilde{t} \leq \tilde{t}' \text{ et } \tilde{t}' \leq \tilde{t} \text{ car } \tilde{t} = \tilde{t}'.$$

4) Bornes inférieure et supérieure dans $T(\Sigma)_{/\equiv}$

Nous étudions maintenant les propriétés de bornes supérieure et inférieure pour l'ordre \leq induit sur $T(\Sigma)_{/\equiv}$.

Notations

- Soit E une partie de $T(\Sigma)_{/\equiv}$.

Nous noterons $\wedge(E)$ la borne inférieure (si elle existe) de E et $\vee(E)$ sa borne supérieure (si elle existe).

- Si E se réduit à 2 éléments $\{t, t'\}$ - on notera encore

$$\wedge(\{t, t'\}) = t \wedge t'$$

$$\vee(\{t, t'\}) = t \vee t'$$

Définition :

On dira que t et t' sont compatibles ssi l'ensemble $\{t, t'\}$ admet un majorant.

Remarque :

Il n'est pas nécessaire que t et t' soient compatibles pour que leur borne inférieure existe.

Exemple :

$$t = \begin{array}{c} a \\ | \\ 1 \end{array} \quad t' = \begin{array}{c} b \\ | \\ 1 \end{array}$$

on a $t \wedge t' = 1$.

Propriétés de \wedge

P II.7 Si $\wedge E$ existe, alors il existe p tel que $E \subset T(\Sigma)_{\equiv}^p$

P II.8 Pour tout $p \in \mathbb{N}$ $T(\Sigma)_{\equiv}^p$ est un inf-demi-treillis.

P II.9 Dans le cas d'ensembles E finis, \wedge est récursive

i.e. il existe un algorithme construisant $\wedge E$ pour tout E fini.

Preuve

II.7 \rightarrow Si t et t' sont de degré supérieur différent alors $t \wedge t' = \emptyset$.

II.8 et II.9

Il faut prouver que pour tout p , toute partie non vide de $T(\Sigma)_{\equiv}^p$ a une borne inférieure et que la construction est effective dans le cas E fini.

Remarquons d'abord que

- il est décidable de savoir si 2 arbres sont égaux
- l'ordre $<$ est décidable, ie il est décidable de savoir si $t < t'$.

Soit alors $E \neq \emptyset$ une partie de $T(\Sigma)_{\equiv}^D$ et fixons un représentant pour chaque classe de E .

Nous appellerons $E' \subset T(\Sigma)^D$ l'ensemble ainsi défini. On va construire un minorant $m = \tilde{m}.\theta$ de E' et montrer que

$$m' \text{ minore } E' \Rightarrow m' < m$$

ce qui prouvera ainsi que $\bar{m} = \wedge E$.

- Nous définissons \tilde{m} comme l'élément de $T(\Sigma)$ dont "les noeuds labellés dans Σ sont tous les noeuds labellés dans Σ communs à tous les arbres de E' . (Deux arbres équivalents par \equiv ayant les mêmes noeuds, on peut raisonner indifféremment sur les classes de \equiv ou sur les représentants).

Si E' est fini la construction de \tilde{m} est alors effective.

- Définition de θ

Pour tout t de E' on a $t = \tilde{m}.t'$.

On choisit alors θ telle que

$$\theta(i) = \theta(j) \Leftrightarrow \forall t \in E' \quad \pi_i(t') = \pi_j(t')$$

Il est évident que le choix de θ est effectif si E' est fini.

Nous avons donc construit (effectivement si E est fini) un élément m .

Vérifions que m est la borne inférieure cherchée.

- Pour tout t de E' on a d'après le choix de θ : $t = \tilde{m}.\theta.t'' = m.t''$ et donc immédiatement m minore E' .

- Soit un minorant m' quelconque de E' . On a $m' = \tilde{m}'.\theta'$

et pour tout t de E' $t = \tilde{m}'.t'''$

Le choix de \tilde{m} impose que \tilde{m}' est sous-arbre initial de \tilde{m} soit

$$m = \tilde{m}'.m''$$

Montrons alors $\theta' < m''$ c'est à dire

$$\theta'(i) = \theta'(j) \Rightarrow \pi_i(m'') = \pi_j(m'')$$

On a

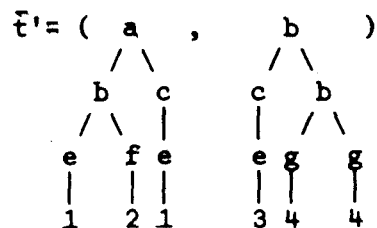
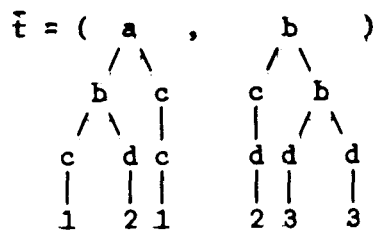
$$\theta'(i) = \theta'(j) \Rightarrow \forall t \in E' \quad \pi_i(t''') = \pi_j(t''')$$

Le fait que $\widetilde{\pi}_i(m'') = \widetilde{\pi}_j(m'')$ est alors évident et le choix de θ impose $\pi_i(m'') = \pi_j(m'')$.

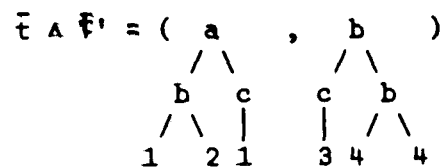
On a donc montré $m' \leq m$.

Exemple II.4 : Bornes inférieures

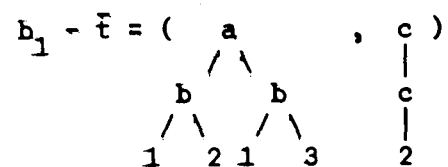
a) $\bar{t} \wedge \bar{t}'$ dans le cas où \bar{t} et \bar{t}' ne sont pas compatibles :



On vérifie facilement que



b) $\bar{t} \wedge \bar{t}'$ dans le cas où \bar{t} et \bar{t}' sont compatibles.



$$\bar{t}' = \left(\begin{array}{cc} & a \\ & / \quad \backslash \\ b & & b \\ & / \quad \backslash \quad / \quad \backslash \\ c & d & c & e \\ & | \quad | \quad | \quad | \\ & 1 & 2 & 1 & 2 \end{array} , \begin{array}{c} c \\ | \\ 3 \end{array} \right)$$

On a

$$\bar{t} \wedge \bar{t}' = \left(\begin{array}{cc} & a \\ & / \quad \backslash \\ b & & b \\ & / \quad \backslash \quad / \quad \backslash \\ 1 & 2 & 1 & 3 \end{array} , \begin{array}{c} c \\ | \\ 4 \end{array} \right)$$

$$b_2 - \bar{t} = \left(\begin{array}{cc} & a \\ & / \quad \backslash \\ & & b \\ & / \quad \backslash \\ 1 & 1 & 2 \end{array} \right)$$

$$\bar{t} = \left(\begin{array}{cc} & a \\ & / \quad \backslash \\ & & b \\ & / \quad \backslash \\ 1 & 2 & 1 \end{array} \right)$$

alors

$$\bar{t} \wedge \bar{t}' = \left(\begin{array}{cc} & a \\ & / \quad \backslash \\ & & b \\ & / \quad \backslash \\ 1 & 2 & 3 \end{array} \right)$$

Toute partie majorée de $T(\Sigma)$ est pour un certain p incluse dans $T(\Sigma)^P$.

Comme on a $T(\Sigma)_{\cong}^P$ inf-demi-treillis on a :

Proposition II.10

Toute partie majorée E de $T(\Sigma)_{\cong}$ admet une borne supérieure $\vee E$.

Nous allons maintenant considérer dans le cas E fini la construction effective de la borne supérieure $\vee E$.

Puisque la borne supérieure est associative on est ramené à un algorithme de construction de la borne supérieure d'un couple (\bar{t}, \bar{t}') de deux éléments compatibles de $T(\Sigma)_{\cong}$. C'est le problème d'unification (du 1er ordre).

II - ALGORITHME D'UNIFICATION

Le corollaire précédent montre que pour la borne supérieure $\bar{t} \vee \bar{t}'$ existe il faut et il suffit que \bar{t} et \bar{t}' soient compatibles. Nous allons donner un algorithme qui décide de l'existence de $\bar{t} \vee \bar{t}'$ et la construit quand elle existe. Bien que la notion d'unification soit définie sur $T(\Sigma)_{/\equiv}$, comme les classes sont de manipulation peu aisée l'algorithme suivant manipulera des représentants de classes et définira $U(t, t')$ comme l'unifié des arbres t et t' représentants des classes \bar{t} et \bar{t}' .

1) Forme générale de l'algorithme

$U(t, t')$ est défini récursivement par

(S) $U(t, t') =$ si degré sup(t) \neq degré sup (t') alors Ω
sinon si $t \equiv t'$ alors t
sinon si $b(t, t')$ alors
 $f_u(t, t').U \gamma (f_r(t, t'))$
sinon Ω

Nous définirons par la suite les opérateurs b , f et γ dont les domaines de définition et de valeurs sont :

$$b : (T(\Sigma))^2 \rightarrow \{\text{vrai}, \text{faux}\}$$

$$\gamma : (T(\Sigma))^2 \rightarrow (T(\Sigma))^2$$

$$f : (T(\Sigma))^2 \rightarrow (T(\Sigma))^3$$

et $f_u : \pi_1.f$

$$f_r = \langle x_2, x_3 \rangle . f$$

Nous imposerons à ces opérateurs une certaine compatibilité avec \equiv :

(P1)

i) $\bar{t} \vee \bar{t}'$ existe $\Rightarrow b(t, t') = \text{vrai}$

$$\text{ii) } f_u(t, t') \cdot \overline{\gamma(f_r(t, t'))} = \bar{t} \vee \bar{t}'$$

$$\text{ou } \begin{cases} t \equiv u \\ \text{et } t' \equiv u' \end{cases} \Rightarrow f_u(t, t') \cdot \overline{\gamma(f_r(t, t'))} = f_u(u, u') \cdot \overline{\gamma(f_r(u, u'))}$$

Remarque :

La cohérence de la dernière égalité se justifie par le lemme de compatibilité à droite avec la composition (II.5).

Comme le degré supérieur est aussi compatible avec \equiv on déduira du schéma (S) par (P1) un schéma (\bar{S}) manipulant les classes de $T(\Sigma)_{/\equiv}$:

$$(\bar{S}) \quad \bar{U}(\bar{t}, \bar{t}') = \begin{array}{l} \underline{\text{si degré sup}(\bar{t}) \neq \text{degré sup}(\bar{t}') \text{ alors } \Omega} \\ \underline{\text{sinon si } \bar{t} = \bar{t}' \text{ alors } \bar{t}} \\ \underline{\underline{\text{sinon si } b(\bar{t}, \bar{t}') \text{ alors} \\ f_u(t, t') \cdot \bar{U} \gamma(f_r(t, t'))}} \\ \underline{\text{sinon } \Omega} \end{array}$$

De plus on s'impose (P2) :

(P2)

Il existe une application Λ de $T(\Sigma)^2$ dans \mathbb{N} telle que

$$\Lambda(\gamma(f_r(t, t'))) < \Lambda(t, t')$$

Le lecteur vérifiera facilement que (P1) implique la correction partielle du programme (S) à savoir que quand (S) s'arrête on a

$$\overline{U(t, t')} = \bar{t} \vee \bar{t}' \quad \text{si } \bar{t} \vee \bar{t}' \text{ existe}$$

$$= \Omega \text{ sinon}$$

Clairement (P2) assure l'arrêt du programme.

Remarquons par conséquent que b est une condition suffisante d'existence de l'unifié. D'autre part \equiv est décidable, il suffit alors de se donner b , f et γ récursives vérifiant (P1) et (P2).

2) Définition de b f et γ

a) - $f : (T(\Sigma))^2 \rightarrow (T(\Sigma))^3$ est une application qui à (t, t') associe (v, u, u') tels que

- torsion de v complète

- $\bar{v} = \bar{t} \wedge \bar{t}'$

- $t = v.u$ et $t' = v.u'$

Comme Λ est récursif (propriété (iii) de Λ) la construction de v est effective donc celle de u et u' et f est récursive.

On pose

$$f_u(t, t') = v$$

$$f_r(t, t') = (u, u')$$

Remarque : t et t' étant des représentants de \bar{t} et \bar{t}' on les choisira vérifiant la condition initiale $\text{Im}(\text{torsion de } t) \cap \text{Im}(\text{torsion de } t') = \emptyset$.

b) Définition :

* un couple (u, u') de $(T(\Sigma))^2$ est une frange si $\text{degré sup}(u) = \text{degré sup}(u')$.

$f_r(t, t')$ est une frange.

* une frange est sans recouvrement ssi $\forall i \pi_i(u) \in \theta$ ou $\pi_i(u') \in \theta$

Exemple II.5 : Franges avec et sans recouvrement

$$t = (1, c, d) \\ \begin{array}{ccc} & | & / \backslash \\ & 2 & 1 \quad 2 \end{array}$$

$$t' = (a, c, d) \\ \begin{array}{ccc} & | & | & / \backslash \\ & 3 & 4 & a \quad 6 \\ & & & | \\ & & & 5 \end{array}$$

$$f_u(t, t') = (1, c, d)$$

$$\begin{array}{c} | \quad / \quad \backslash \\ 2 \quad 3 \quad 4 \end{array}$$

$$f_r(t, t') = ((1, 2, 1, 2), (a, 4, a, 6))$$

$$\begin{array}{c} | \quad | \\ 3 \quad 5 \end{array}$$

est sans recouvrement.

- la frange $((1, 2, d), (1, c, e))$

$$\begin{array}{c} / \quad \backslash \quad | \quad / \quad \backslash \\ 1 \quad 2 \quad 2 \quad 4 \quad 5 \end{array}$$

a 2 arbres distincts au 3e rang de ses 2 composantes et n'est donc pas sans recouvrement.

* Une frange sans recouvrement est sans cycle ssi

$$\left\{ \begin{array}{l} - \pi_i(u) = x_j \\ \text{et } \pi_i(u') \neq x_j \end{array} \right\} \Rightarrow \text{Im}(\text{torsion } \pi_i(u')) \neq j$$

$$\left\{ \begin{array}{l} - \pi_i(u') = x_j \\ \text{et } \pi_i(u) \neq x_j \end{array} \right\} \Rightarrow \text{Im}(\text{torsion } \pi_i(u)) \neq j$$

Exemple II.6 : Franges avec ou sans cycle.

Le 1er exemple précédent de frange sans recouvrement est sans cycle.

Par contre la frange $(u, u') = ((a, b), (1, b))$ comporte un cycle

$$\begin{array}{c} | \quad / \quad \backslash \quad | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \quad 3 \quad 4 \end{array}$$

puisque l'on retrouve la variable x_1 dans $\pi_1(u) = a$ et $\pi_1(u') = 1$

$$\begin{array}{c} | \\ 1 \end{array}$$

Définition de b :

Nous poserons

$b(t, t') = \text{vrai}$ ssi $f_r(t, t')$ - construit dans l'algorithme - est sans recouvrement et sans cycle. (Rappelons que b est une condition suffisante d'existence de l'unifié de t, t').

c) Opérateur γ

γ ne s'applique dans (S) qu'à des franges sans recouvrement et sans cycle.

D'autre part

$$- f_r(t, t') = (u, u) \Rightarrow t = t'$$

$$- t \neq t' \Rightarrow t \neq t'$$

donc γ ne s'applique qu'à des franges (u, u') sans recouvrement sans cycle et où $u \neq u'$. Ceci nous permet de restreindre comme suit la définition de γ :

Soit i tel que $\pi_i(u) \neq \pi_i(u')$.

D'après les restrictions précédentes un tel i existe et de plus il existe j

tel que $\pi_i(u) = x_j$ ou $\pi_i(u') = x_j$

Supposons $\pi_i(u) = x_j$ et soit $\pi_i(u') = v$

Soit v (respectivement v') l'arbre obtenu en substituant dans u (respectivement u') l'arbre V à x_j . On posera

$$\gamma(u, u') = (v, v').$$

Nous allons maintenant prouver (P1) et (P2).

3) Propriétés de h , f , et γ

P II.11 $f_u(t, t'). \vee \overline{f_r(t, t')} = \tilde{t} \vee \tilde{t}'$

Preuve :

Elle repose sur le lemme suivant.

Lemme II.12

$$v.(\tilde{w} \vee \tilde{w}') = \overline{v.w} \vee \overline{v.w'}$$

Preuve du lemme

- Soit $t > w$ et $t > w'$

Alors $v.t > v.w$ et $v.t > v.w'$

$$\Rightarrow \overline{v.t} = v.\bar{t} \geq \overline{v.w}$$

$$\text{et } v.\bar{t} \geq \overline{v.w'}$$

$$\Rightarrow v.\bar{t} \geq \overline{v.w} \vee \overline{v.w'}$$

En particulier pour t élément de la classe $(\bar{w} \vee \bar{w}')$ on a

$$v.(\bar{w} \vee \bar{w}') \geq \overline{v.w} \vee \overline{v.w'}$$

- Réciproquement soit u élément de la classe $\overline{v.w} \vee \overline{v.w'}$.

$$\text{On a donc } \bar{u} \geq \overline{v.w} \quad u > v.w$$

$$\text{et } \bar{u} \geq \overline{v.w'} \quad \text{donc } u > v.w'$$

u peut s'écrire $u = v.u'$ et on a

$$v.u' > v.w$$

$$v.u' > v.w'$$

La torsion de v étant complète (i.e. surjective), on a alors

$$u' > w$$

$$\text{et } u' > w'$$

soit donc $\bar{u}' \geq \bar{w} \wedge \bar{w}'$

Par conséquent

$$v.\bar{u}' = \overline{v.u'} = \bar{u} = \overline{v.w} \vee \overline{v.w'} \geq v.(\bar{w} \wedge \bar{w}')$$

Appliquons alors le lemme précédent avec $f_p(t, t') = (u, u')$.

On aura

$$f_u(t, t').(\bar{u} \vee \bar{u}') = \overline{f_u(t, t').u} \vee \overline{f_u(t, t').u'}$$

$$\text{or } t = f_u(t, t').u$$

$$t' = f_u(t, t').u'$$

$$\text{d'où } f_u(t, t').(\bar{u} \vee \bar{u}') = \bar{t} \vee \bar{t}'.$$

Remarque

Il est important que la torsion de v soit surjective sinon les deux membres du lemme II.12 peuvent ne pas exister simultanément ainsi que le montre l'exemple suivant :

$$v = \begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array}, \bar{w} = (b, \begin{array}{c} c \\ / \quad \backslash \\ 1 \quad 2 \end{array}, d), \bar{w}' = (b, \begin{array}{c} c \\ / \quad \backslash \\ 1 \quad 2 \end{array}, \begin{array}{c} e \\ | \\ 3 \end{array})$$

Manifestement $\bar{w} \vee \bar{w}'$ n'existe pas et pourtant $\overline{v.w} \vee \overline{vw}'$ existe et se définit par

$$\begin{array}{c} (a) \\ / \quad \backslash \\ b \quad c \\ | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array} \vee \begin{array}{c} (a) \\ / \quad \backslash \\ b \quad c \\ | \quad / \quad \backslash \\ 1 \quad 2 \quad 3 \end{array} = \begin{array}{c} (a) \\ / \quad \backslash \\ b \quad c \\ | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array}$$

P II.13 $\vee \gamma(\overline{f_r(t, t')}) = \overline{\vee (f_r(t, t'))}$

Preuve :

Nous allons montrer simultanément une propriété intermédiaire des égalités dans les torsions :

Propriété II.14 :

Si dans $f_r(t, t')$ $x_i = x_j$, tout majorant commun de t et t' est nécessairement obtenu en substituant un même arbre à x_i et x_j .

La preuve de cette propriété se fait par récurrence dans (S) :

- la condition initiale

$$\text{Im}(\text{torsion de } t) \cap \text{Im}(\text{torsion de } t') = \emptyset$$

entraîne que la propriété est initialement vraie.

- D'autre part f_r ne modifie pas la numérotation des variables et conserve donc cette propriété.

- Montrons que γ conserve encore cette propriété :

soit avec les notations définies en II.2.C :

$$\gamma(u, u') = (v, v') \text{ avec } \pi_i(u) = x_j \text{ et } \pi_i(u') = v$$

soit $w_1 > u$ et $w_1 > u'$

donc $w_1 = u.w$ et $w_1 = u'.w'$

$$\pi_i(w_1) = \pi_i(u.w) = x_j.w = \pi_j(w) = \pi_i(u'.w') = v.w'$$

On construit alors \hat{w} à partir de w tel que :

- pour tout x_ℓ figurant dans v mais pas dans u on égale la $\ell^{\text{ème}}$ composante de w à celle de w' .

- Pour tout x_ℓ figurant dans v et dans u on a par hypothèse $\pi_\ell(w) = \pi_\ell(w')$.

$$\text{Alors } v.\hat{w} = u.w = w_1$$

Comme tout x_ℓ de v figure dans u' on a

$$v'.w' = u'.w' = w_1$$

Donc w_1 majore aussi v et v' .

D'autre part tout majorant de v et v' majore évidemment u et u' d'où l'égalité annoncée en b.

Enfin à toute occurrence de x_ℓ dans v ou v' doit être nécessairement substitué un même arbre d'où la propriété des égalités dans les torsions.

Il est important de noter que l'arbre substitué peut éventuellement se réduire à une torsion c'est à dire à une variable comme on le verra dans l'exemple II.7.

De P II.11 et P II.13 on déduit immédiatement la propriété P1 ii) :

$$f_u(t, t') \vee \overline{\gamma(f_r(t, t'))} = \bar{t} \vee \bar{t}'.$$

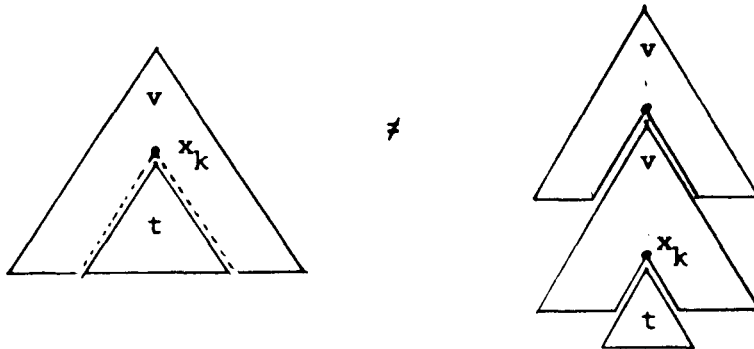
Prouvons alors P1 i) :

Si $\bar{t} \vee \bar{t}'$ existe $(u, u') = f_r(t, t')$ est sans recouvrement. En effet supposons le contraire ; soit i tel que $\pi_i(u) \neq \emptyset$ et $\pi_i(u') \neq \emptyset$.

Nécessairement les sommets de $\pi_i(u)$ et $\pi_i(u')$ sont des lettres différentes de Σ sinon on aurait $\pi_i(u) = \pi_i(u')$ contenu dans $\bar{t} \wedge \bar{t}'$. Puisqu'alors u et u' se composent au même rang i de deux arbres distincts on ne peut construire leur unifié et t et t' étant alors incomparables $\bar{t} \vee \bar{t}'$ n'existe pas.

D'après la propriété des égalités dans les torsions si $f_r(t, t')$ a un cycle (soit par exemple $(x_k, v.x_k)$ ce cycle) alors tout majorant commun à t et t' s'obtient nécessairement en substituant un même arbre à chaque occurrence de x_k ce qui est impossible et t et t' sont incompatibles.

On aurait en effet



Si $\bar{t} \vee \bar{t}'$ existe on a donc $f_r(t, t')$ sans recouvrement ni cycle et $b(t, t') = \text{vrai}$.

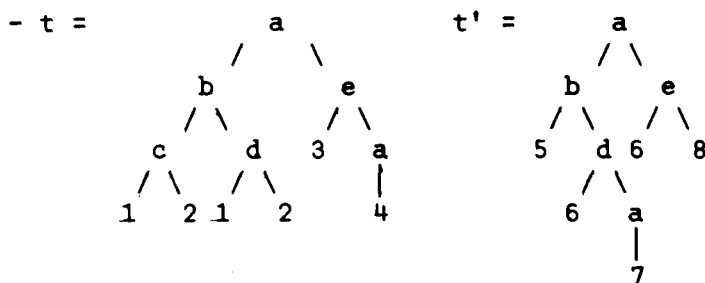
Montrons enfin la propriété P2 :

Soit A la fonction qui à (t, t') associe le nombre de variables différentes apparaissant effectivement dans (t, t') .

f_r ne modifie pas ce nombre mais γ le diminue effectivement de 1 puisque à chaque substitution d'un V à x_j ce x_j disparaît.

4) Exemple II.7 : recherche d'unifiés de 2 arbres.

a)



$$- t \wedge t' = \begin{array}{c} a \\ / \quad \backslash \\ b \quad e \\ / \quad \backslash \quad / \quad \backslash \\ 1 \quad d \quad 4 \quad 5 \\ / \quad \backslash \\ 2 \quad 3 \end{array} = f_u(t, t')$$

$$- f_r(t, t') = (u_1, u'_1) = ((c, 1, 2, 3, a), (5, 6, a, 6, 8))$$

$$\begin{array}{c} \begin{array}{c} c \\ / \quad \backslash \\ 1 \quad 2 \end{array} \quad \begin{array}{c} a \\ | \\ 4 \end{array} \quad \begin{array}{c} a \\ | \\ 7 \end{array} \end{array}$$

est sans cycle et sans recouvrement donc $b(t, t') = \text{vrai}$, et $u_1 \neq u'_1$

donc γ s'applique une lère fois :

Substituons a à 2 . On obtient

$$\begin{array}{c} | \\ 7 \end{array}$$

$$v_1 = (c, 1, a, 3, a)$$

$$\begin{array}{c} \begin{array}{c} c \\ / \quad \backslash \\ 1 \quad a \end{array} \quad \begin{array}{c} a \\ | \\ 7 \end{array} \quad \begin{array}{c} a \\ | \\ 4 \end{array} \end{array}$$

$$v'_1 = (5, 6, a, 6, 8) = v_1$$

$$\begin{array}{c} a \\ | \\ 7 \end{array}$$

$$\text{et } u_1 \vee u'_1 = v_1 \vee v'_1$$

Cherchons alors $v_1 \vee v'_1$.

$$- v_1 \Delta v'_1 = (1, 2, a, 4, 5) = f_u(v_1, v'_1)$$

$$\begin{array}{c} a \\ | \\ 3 \end{array}$$

$$- f_r(v_1, v'_1) = (u_2, u'_2) = ((c, 1, 7, 3, a), (5, 6, 7, 6, 8))$$

$$\begin{array}{c} \begin{array}{c} c \\ / \quad \backslash \\ 1 \quad a \end{array} \quad \begin{array}{c} a \\ | \\ 7 \end{array} \quad \begin{array}{c} a \\ | \\ 4 \end{array} \end{array}$$

est sans cycle et sans recouvrement donc $b(v_1, v'_1) = \text{vrai}$, et $u_2 \neq u'_2$

donc γ s'applique une 2e fois :

substituons c à 5 . On obtient $v_2 = (c, 1, 7, 3, a) = v_1$

$$\begin{array}{c} \begin{array}{c} c \\ / \quad \backslash \\ 1 \quad a \end{array} \quad \begin{array}{c} a \\ | \\ 7 \end{array} \quad \begin{array}{c} a \\ | \\ 4 \end{array} \end{array}$$

$$v'_2 = \left(\begin{array}{c} c \\ / \quad \backslash \\ 1 \quad a \\ | \\ 7 \end{array} \right), 6, 7, 6, 8) \text{ et } v_1 \vee v'_1 = v_2 \vee v'_2.$$

Cherchons alors $v_2 \vee v'_2$.

$$- v_2 \wedge v'_2 = \left(\begin{array}{c} c \\ / \quad \backslash \\ 1 \quad a \\ | \\ 2 \end{array} \right), 3, 2, 5, 6) = f_u(v_2, v'_2)$$

$$f_r(v_2, v'_2) = (u_3, u'_3) = ((1, 7, 1, 7, 3, \begin{array}{c} a \\ | \\ 4 \end{array}), (1, 7, 6, 7, 6, 8))$$

est sans recouvrement ni cycle, donc $b(v_2, v'_2) = \text{vrai}$ et $u_3 \neq u'_3 \neq u_3$
donc γ s'applique une 3ème fois.

Substituons a à 8 . On obtient

$$v_3 = (1, 7, 1, 7, 3, \begin{array}{c} a \\ | \\ 4 \end{array}) = u_3$$

$$v'_3 = (1, 7, 6, 7, 6, \begin{array}{c} a \\ | \\ 4 \end{array})$$

$$\text{et } v_2 \vee v'_2 = v_3 \vee v'_3$$

Cherchons $v_3 \vee v'_3$:

$$v_3 \wedge v'_3 = (1, 2, 3, 2, 4, \begin{array}{c} a \\ | \\ 5 \end{array}) = f_u(v_3, v'_3)$$

$$f_r(v_3, v'_3) = (u_4, u'_4) = ((1, 7, 1, 7, 3, 4), (1, 7, 6, 7, 6, 4))$$

est sans recouvrement ni cycle donc $b(v_3, v'_3) = \text{vrai}$ et $u_4 \neq u'_4$.

On applique γ et cette fois la substitution porte sur une même variable
selon la propriété des égalités dans les torsions :

On substitue 1 à 6 :

$$v_4 = (1, 7, 1, 7, 3, 4) = u_4$$

$$v'_4 = (1, 7, 1, 7, 1, 4)$$

$$\text{et } v_3 \vee v'_3 = v_4 \vee v'_4.$$

On aura

$$v_4 \wedge v'_4 = (1, 2, 1, 2, 3, 4)$$

$$f_r(v_4, v'_4) = (u_5, u'_5) = ((1, 7, 1, 7, 3, 4), (1, 7, 1, 7, 1, 4))$$

(u_5, u'_5) est encore sans cycle ni recouvrement, $b(v_4, v'_4) = \text{vrai}$ et

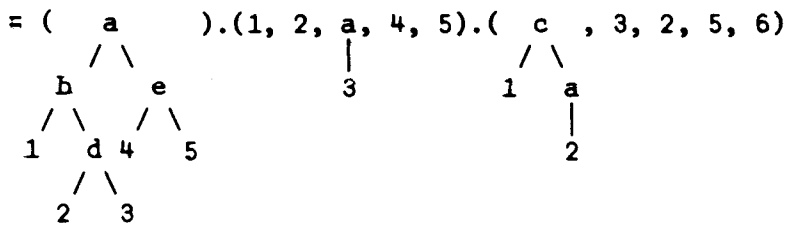
γ s'applique une dernière fois en substituant à 1 et 3 une variable commune soit finalement

$$v_5 = v'_5 = (1, 7, 1, 7, 1, 4)$$

On a alors $v_5 \vee v'_5 = v_5$ et l'algorithme s'arrête.

D'où en récapitulant et en prenant l'image de la torsion de v_5 dans $\{1, 2, 3\}$:

$$t \vee t' = (t \wedge t') \cdot (v_1 \wedge v'_1) \cdot (v_2 \wedge v'_2) \cdot (v_3 \wedge v'_3) \cdot (v_4 \wedge v'_4) \cdot v_5$$

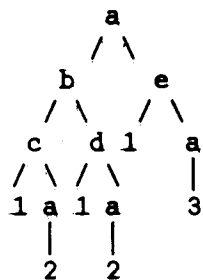


$$\cdot (1, 2, 3, 2, 4, \underset{\substack{| \\ 5}}{a}) \cdot (1, 2, 1, 2, 3, 4) \cdot (1, 2, 1, 2, 1, 3)$$



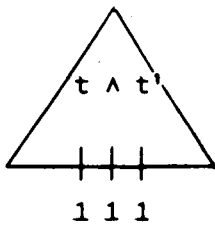
et en recomposant

$$t \vee t' =$$

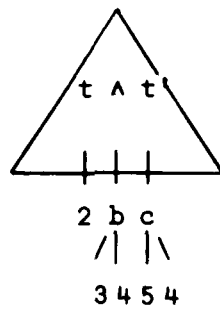


b)

t =



t' =



$$f_r(t, t') = (u, u') = ((1, 1, 1), (2, \begin{matrix} b & c \\ / \quad \backslash \\ 3 & 4 & 5 & 4 \end{matrix}))$$

est sans recouvrement ni cycle donc $b(t, t') = \text{vrai}$, et $u \neq u'$ donc γ s'applique.

On substitue b à 1 d'où

$$v = (\begin{matrix} b \\ / \quad \backslash \\ 3 & 4 \end{matrix} , \begin{matrix} b \\ / \quad \backslash \\ 3 & 4 \end{matrix} , \begin{matrix} b \\ / \quad \backslash \\ 3 & 4 \end{matrix})$$

$$v' = (2, \begin{matrix} b & c \\ / \quad \backslash & / \quad \backslash \\ 3 & 4 & 5 & 4 \end{matrix})$$

et $u \vee u' = v \vee v'$,

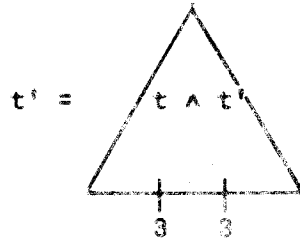
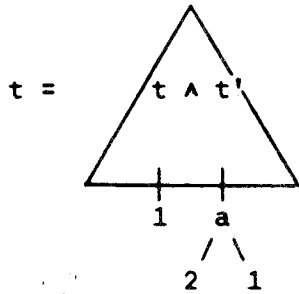
Cherchons $v \wedge v'$:

$$v \wedge v' = (1, \begin{matrix} b \\ / \quad \backslash \\ 2 & 3 \end{matrix} , 4)$$

$$f_r(v, v') = ((\begin{matrix} b \\ / \quad \backslash \\ 3 & 4 \end{matrix} , 3, 4, \begin{matrix} b \\ / \quad \backslash \\ 3 & 4 \end{matrix}), (2, 3, 4, \begin{matrix} c \\ / \quad \backslash \\ 5 & 4 \end{matrix}))$$

elle n'est pas sans recouvrement donc $b(v, v')$ est faux et $t \vee t'$ n'existe pas.

c)



$$f_r(t, t') = ((1, \begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array}), (3, 3)) = (u, u')$$

est sans recouvrement et sans cycle donc $b(t, t')$ est vrai et $u \neq u'$

donc γ s'applique. On substitue $\begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array}$ à 3 d'où

$$v = (1, \begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array})$$

$$v' = (\begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array}, \begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array})$$

$$\text{et } u \vee u' = v \vee v'$$

Cherchons

$$v \wedge v' = (1, \begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array})$$

$$f_r(v, v') = ((1, 2, 1), (\begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array}, 2, 1)) = (w, w')$$

Cette frange a un cycle en rang 1 puisque

$$\pi_1(w) = 1 \quad \text{et} \quad \pi_1(w') = \begin{array}{c} a \\ \swarrow \searrow \\ 2 \quad 1 \end{array}$$

donc $b(v, v)$ est faux et l'algorithme s'arrête ; $t \vee t'$ n'existe pas.

B - LES FORETS RATEG

L'étude des propriétés de clôture des forêts reconnaissables par union intersection et morphisme nous a amenés à introduire la notion de forêts RATEG.

I - INTRODUCTION AUX RATEG

Les forêts RATEG sont des forêts reconnues par des automates particuliers : ce sont des automates ascendants d'états finis d'arbres classiques qui permettent en plus de tester des égalités de sous arbres - d'où leur nom de Reconnaisseurs Avec Tests d'Egalité -

De plus ces tests peuvent se faire sur des sous-arbres situés à distance bornée des noeuds en cours de reconnaissance.

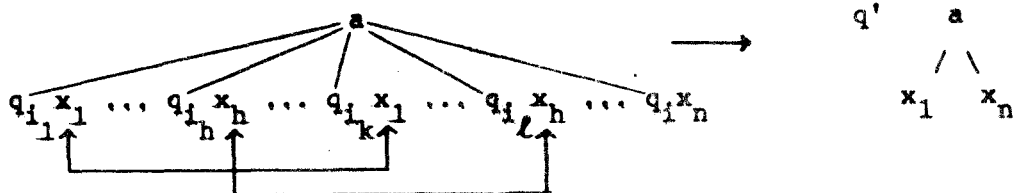
Plus précisément :

* un automate classique a des règles de la forme



qui ne font qu'assurer des contraintes de reconnaissabilité entre les sommets.

* si on ajoute par une torsion la possibilité de tester des égalités uniquement sur les successeurs immédiats des noeuds en cours de reconnaissance, on obtient des règles de la forme



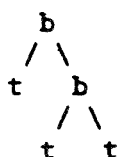
permettant de reconnaître des arbres de la forme



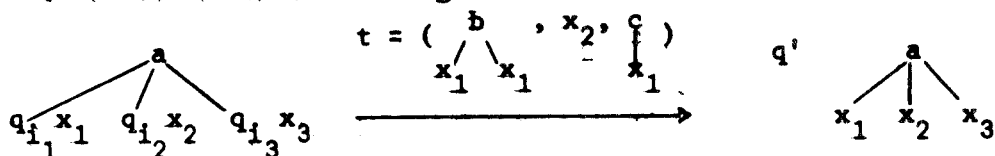
si et seulement si $t_1 = t_k$ et $t_h = t_l$.

On obtient ainsi une classe de forêts strictement incluse dans l'ensemble des images homomorphes de reconnaissables et sans beaucoup d'intérêt car trop restreinte.

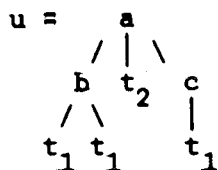
Il est visible qu'on ne pourra par ces automates reconnaître les images homomorphes de reconnaissables de la forme



* On introduit alors des règles de la forme

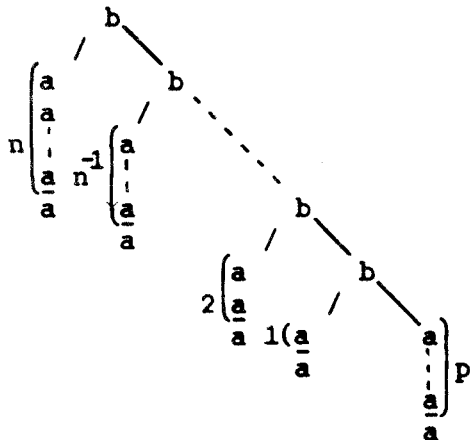


qui ne s'appliqueront à un arbre u que si u vérifie

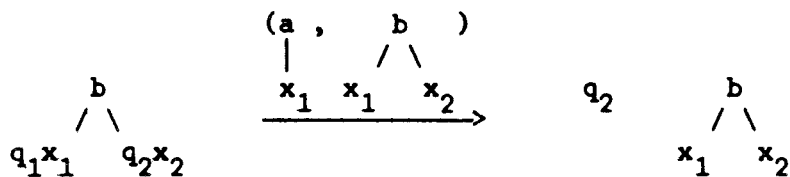


Ainsi l'adjonction à la règle d'un arbre t permet de contrôler des égalités de sous-arbres situés à distance bornée du noeud concerné.

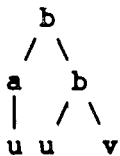
Considérons par exemple la forêt des arbres de la forme



où les branches gauches de deux "b" consécutifs diffèrent d'un "a".
 Le lecteur constatera sans difficulté que la vérification de cette
 contrainte se fera par une règle



reconnaissant des arbres de la forme



Nous donnerons en Exemples II.10 et II.11 deux RATEG et l'ensemble de toutes
 leurs règles.

* Pour des raisons techniques (établissement de la proposition II.30 en
 particulier) nous introduirons également la notion de RATEG généralisé,
 se différenciant du précédent par le fait que le membre gauche (et droit)
 des règles généralisées sera un arbre de $T(\Sigma)_n^1$ et non plus une lettre de
 Σ_n . Et l'on vérifiera (proposition II.20) que la classe des forêts RATEG
 coïncide avec celles reconnues par les RATEG généralisés.

Nature des règles des RATEG

On remarquera, comme annoncé précédemment, que les torsions n'interviennent dans les règles que pour imposer des tests d'égalité.

De ce fait la vraie nature des RATEG est de manipuler des éléments de $T(\Sigma)_{\equiv}$.

Dans la présentation formelle (II.B.II.1) nous définirons d'abord les $\bar{\text{RATEG}}$ dont les règles font apparaître des classes de $T(\Sigma)_{\equiv}$. Nous verrons néanmoins (II.B.II.2) que dans la pratique on se ramènera à l'étude des RATEG ce qui revient à travailler sur des représentants de classes c'est à dire "aux numérotations de torsions près".

II - DÉFINITIONS FORMELLES

Notations

- $x = (x_1, x_2, \dots, x_n)$ est un n-uple de variables x_i que l'on écrira encore $(1, 2, \dots, n)$.

- $u = (u_1, u_2, \dots, u_n)$ est un n-uple d'arbres u_i de $T(\Sigma)_{p_i}^1$ qui s'écrira encore $u = \tilde{u}.\theta$ où \tilde{u} est l'arbre sans torsion correspondant de $T(\Sigma)$ et θ la torsion.

- $q = (q_1, q_2, \dots, q_n)$ est un n-uple d'états q_i et pour tout q et tout x de même ordre n , nous noterons

$$q[x] = (q_1, q_2, \dots, q_n) [(x_1, x_2, \dots, x_n)] = (q_1[x_1], q_2[x_2], \dots, q_n[x_n])$$

le n-uple des variables x_i affectées des états q_i de même rang i .

De même $q[u] = (q_1[u_1], \dots, q_n[u_n])$ désignera le n-uple d'arbres u_i dont le sommet est affecté de l'état q_i de même rang i .

1) Les RATEG

Définition

Un RATEG est un quadruplet $M = (\Sigma, Q, F, \bar{R})$ où

- Σ est un alphabet gradué fini
- Q un ensemble fini d'états
- F un ensemble d'états finaux pris dans Q
- \bar{R} un ensemble fini de règles définies comme suit :

Définition

Toute règle d'un RATEG a l'une des 2 formes suivantes :

- si $a \in \Sigma_0$

$$a \xrightarrow{\bar{\theta}} q[a] \text{ avec } \bar{\theta} \text{ classe de la torsion vide .}$$

N.B. Rappelons que la torsion vide est la seule application de \emptyset dans \emptyset .

- Si $a \in \Sigma_n$

$$a(q[x]) \xrightarrow{\bar{t}} q'[a(x)]$$

où \bar{t} désigne une classe de $T(\Sigma)_{/\equiv}$.

Définition

Un mouvement élémentaire de RATEG M est défini par

- $u \in T(\Sigma)_p^n$

- $v.a.q[u] \stackrel{r}{\sim} v.q'[a(u)]$

ssi - il existe dans M une règle r :

$$a(q[x]) \xrightarrow{\bar{t}} q'[a(x)]$$

- et il existe $t \in \bar{t}$, $t \in T(\Sigma)_q^n$, tel que

$$\exists w \in T(\Sigma)_p^q : u = t.w$$

Remarque : Cette définition est indépendante du représentant choisi de \bar{t} et le lecteur prouvera facilement le lemme suivant.

Lemme II.15

Soit $t \equiv t'$

alors $(\exists w : u = t.w) \Leftrightarrow (\exists w' : u = t'.w')$

L'idée de la preuve est que les torsions de deux représentants d'une même classe testent les mêmes égalités.

Nous introduisons maintenant les RATEG dont les définitions diffèrent des précédentes en remplaçant les classes de $T(\Sigma)_{/\equiv}$ par un représentant, c'est à dire cela revient intuitivement à choisir un système de représentants de règles. Par la suite nous ne manipulerons plus que des RATEG.

2) Les RATEGDéfinition

Un RATEG est un quadruplet $M = (\Sigma, Q, F, R)$ où

- Σ est un alphabet gradué fini
- Q un ensemble fini d'états
- F un ensemble d'états finaux pris dans Q
- R un ensemble fini de règles définies comme suit :

Définition

Une règle de RATEG est de la forme

- si $a \in \Sigma_0$

$$a \xrightarrow{\theta} q[a] \text{ avec } \theta \text{ torsion vide}$$

- si $a \in \Sigma_n$

$$a(q[x]) \longrightarrow q'[a(x)]$$

- où - $t \in T(\Sigma)_p^n$
 - $q \in Q^n$
 - $x = (x_1, \dots, x_n)$
 - $q' \in Q$

Nous définissons immédiatement des règles équivalentes.

Définition

Soit $r : a(q[x]) \xrightarrow{t} q'[a(x)]$

$r' : a(q[x]) \xrightarrow{t'} q'[a(x)]$

On dira $r \equiv r'$ ssi $t \equiv t'$.

Définition

Un mouvement élémentaire d'un RATEG est défini par :

$v.a.q[u] \xrightarrow{r} v.q'[a(u)]$

\Leftrightarrow

$\left(\begin{array}{l}
 - \text{il existe } r : \\
 a(q[x]) \xrightarrow{t} q'[a(x)] \\
 - \text{il existe } w : \\
 u = t.w
 \end{array} \right.$

Propriété II.16

$r \equiv r' \Rightarrow$

$\left(\begin{array}{l}
 v.a.q[u] \xrightarrow{\frac{r}{M}} v.q'[a(u)] \\
 \Leftrightarrow v.a.q[u] \xrightarrow{\frac{r'}{M}} v.q'[a(u)] \\
 \Leftrightarrow v.a.q[u] \xrightarrow{\frac{r'}{M}} v.q'[a(u)]
 \end{array} \right.$

Remarque

Les définitions de règle et de mouvement d'un RATEG sont des généralisations de celles d'un automate d'états finis d'arbres classique.

Rappelons en effet qu'une règle d'un tel automate s'écrit :

$$\begin{aligned} & - \text{ si } a \in \Sigma_0 \\ & \quad a \rightarrow q[a] \end{aligned}$$

$$- \text{ si } a \in \Sigma_n$$



Il est alors aisé de les réécrire sous la forme

$$- a \xrightarrow{\theta} q[a] \text{ avec } \theta \text{ torsion vide}$$

$$- a(q[x]) \xrightarrow{t} q'[a(x)]$$

$$\text{avec } q = (q_1 \dots q_n)$$

$$x = (x_1 \dots x_n)$$

$$t = (x_1 \theta x_1 \dots \theta x_1) = \text{Id}_n$$

et on écrit alors

$$v, a, q[u] \vdash v q'[a(u)]$$

ssi $a q[x] \rightarrow q'[a(x)]$ dans R.

Par conséquent tout automate d'états finis $M = (\Sigma, Q, F, R)$ est un RATEG

$M' = (Q, \Sigma, F, R')$ où R' se déduit de R en adjoignant à chaque règle la

torsion vide ou la torsion identité.

Lemme (des torsions "surjectives") II.17 :

Dans la décomposition $t = \tilde{t} \cdot \theta$ on peut toujours considérer θ surjective.

Preuve :

Considérons $t = \tilde{t} \cdot \theta$ où $\tilde{t} \in T(\Sigma)_p^n$

et $\theta \in \Theta_r^p$ soit $\theta : [p] \rightarrow [r]$

Soit $q = \text{Sup Im}(\theta)$

Alors $\forall p' \geq q$, soit $\theta_{p'} : [p] \rightarrow [p']$ associée à θ par :

$$\forall i \in [p] \quad \theta_{p'}(i) = \theta(i)$$

Il est évident que $\forall p' \geq q$, $\forall w \in T(\Sigma)_o^q$ et $\forall w' \in T(\Sigma)_o^{p'-q}$

$$\theta_q \cdot w = \theta_{p'} \langle w, \theta w' \rangle.$$

Dans la définition d'un mouvement d'un RATEG, le couple (θ, w) n'intervenant que par $\theta \cdot w$, on peut par conséquent supposer $\theta = \theta_q$, c'est à dire prendre θ surjective sans restreindre la définition.

Dans l'exemple II.8 cela revient à considérer w réduit à ses 3 premières composantes. Par la suite on supposera toujours θ surjective.

3) Les RATEG généralisés

Définition

Un RATEG généralisé est un quadruplet $M = (\Sigma, Q, F, R)$ où Σ, Q, F sont définis comme pour un RATEG et R l'ensemble fini de règles de la forme suivante :

- si $a \in \Sigma_o$

$$a \xrightarrow{\theta} q[a] \text{ avec } \theta \text{ torsion vide}$$

- si $a \in T(\Sigma)_n^1$ (membre gauche non réduit à une lettre)

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$

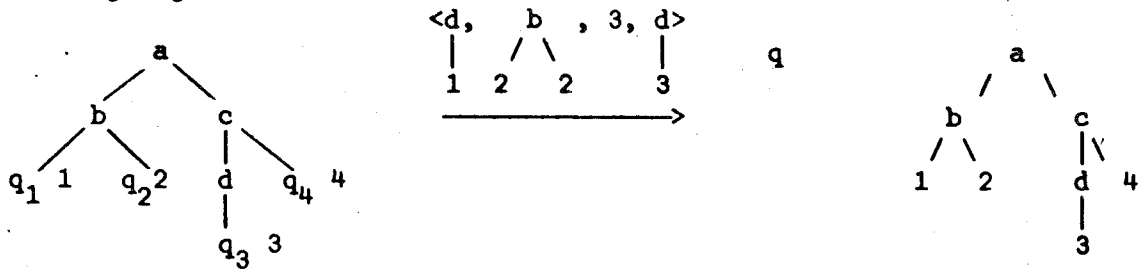
où $q \in Q^n$

$q' \in Q$

$t \in T(\Sigma)_p^n$

Exemple II.9

Une règle généralisée



que l'on écrira encore

$$a(b \otimes c)(Id_2 \otimes d \otimes Id_1)(q[x]) \xrightarrow{t} q[a(b \otimes c)(Id_2 \otimes d \otimes Id_1)(x)]$$

III - DEUX EXEMPLES DE RATEG

Exemple II.10

Soit $M = (\Sigma, Q, F, R)$ avec

$$\Sigma = \{\bar{a}, a(1), b(1, 2)\}$$

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

$$F = \{q_1\}$$

et R ensemble des règles suivantes :

$$R_0 \bar{a} \rightarrow q_0[\bar{a}]$$

$$R_1 a(q_0[1]) \rightarrow q_0[a(1)]$$

$$R_2 b((q_0, q_0)[1, 2]) \xrightarrow{\langle 1, 2 \rangle} q_1[b(1, 2)]$$

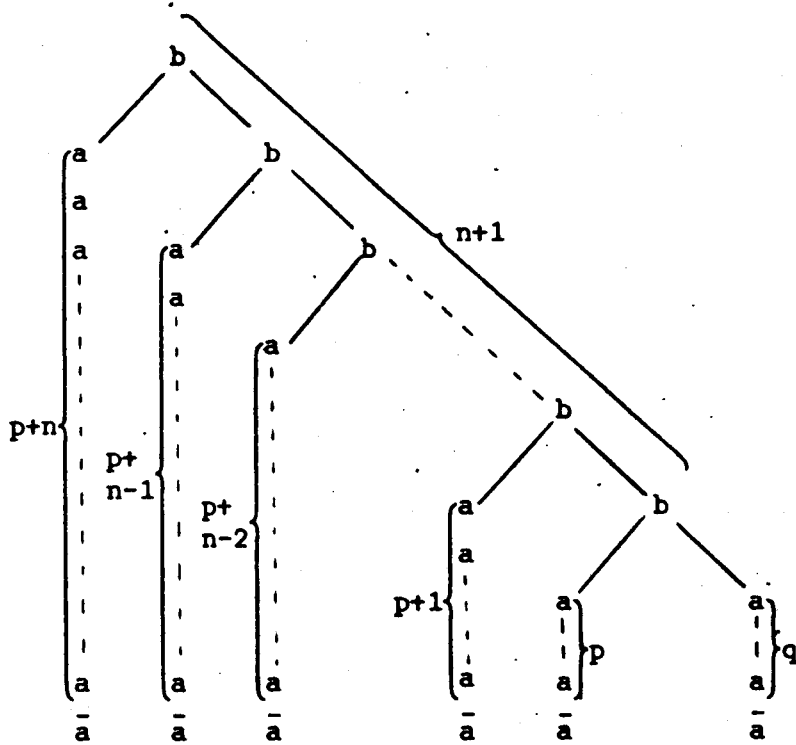
$$R_3 b((q_0, q_1)[1, 2]) \xrightarrow{\langle a, b \rangle} q_1[b(1, 2)]$$

La règle R1 permet de reconnaître une branche de "a" de longueur quelconque.

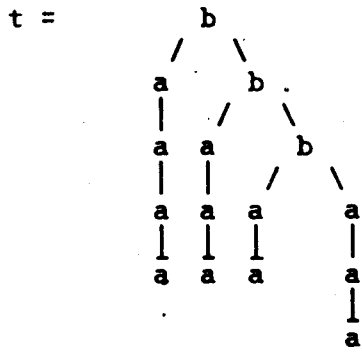
La règle R2 assure comme R0 le départ de tout mouvement de $t \in F(M)$. Enfin

R3 permet le passage d'un "b" à son prédécesseur immédiat "b" en vérifiant l'adjonction d'un "a" à la branche gauche.

On vérifie en effet que la forêt RATEG reconnue par M est constituée des arbres de la forme suivante (où p, q et n \in)



Ecrivons par exemple le mouvement reconnaissant



- R_0^* $t \mid b(a(aa q_0[\bar{a}]) \otimes b(a(aq_0[\bar{a}]) \otimes b(a(q_0[\bar{a}] \otimes aaq_0[\bar{a}])))$
- R_1^* $t \mid b(q_0[aaaa\bar{a}] \otimes b(q_0[aaa\bar{a}] \otimes b(q_0[aa\bar{a}] \otimes q_0[aa\bar{a}]))$
- R_2 $t \mid b(q_0[aaaa\bar{a}] \otimes b(q_0[aaa\bar{a}] \otimes q_1[b(aa\bar{a}, aa\bar{a}])))$
- R_2 $t \mid b(q_0[aaaa\bar{a}] \otimes q_1[b(aa\bar{a} \otimes b(aa\bar{a}, aa\bar{a}]))$
- R_3 $t \mid q_1[t].$



Exemple II.11

Considérons $M = (\Sigma, Q, F, R)$ avec

- l'alphabet $\Sigma = \{\neq(1, 2, 3), \mathcal{E}(1, 2), a(1), \bar{a}\}$
- l'ensemble d'états $Q = \{q_0, q_1, q_2, q_3, q_f\}$
- l'ensemble d'états finaux $F = \{q_f\}$
- et l'ensemble fini des règles R suivantes :

$$R_0. \bar{a} \xrightarrow{\theta} q_0[\bar{a}] \text{ (avec } \theta \text{ torsion vide)}$$

$$R_1. a(q_0[1]) \xrightarrow{1} q_1[a(1)]$$

$$R_2. \mathcal{E}(q_0[1], q_0[2]) \xrightarrow{\langle 1, 1 \rangle} q_2[\mathcal{E}(1, 2)]$$

$$R_3. \neq(q_1[1], q_2[2], q_1[3]) \xrightarrow{\begin{array}{c} \langle \bar{a}, \mathcal{E}, a \rangle \\ | \quad / \quad \backslash \quad | \\ 1 \quad 1 \quad 1 \quad 1 \end{array}} q_2[\neq(1, 2, 3)]$$

$$R_4. \neq(q_1[1], q_2[2], q_1[3]) \xrightarrow{\begin{array}{c} \langle a, \neq, a \rangle \\ | \quad / \quad \backslash \quad | \\ 1 \quad 1 \quad 2 \quad 1 \quad 1 \\ \langle 1, \neq, 1 \rangle \end{array}} q_2[\neq(1, 2, 3)]$$

$$R_5. \neq(q_1[1], q_2[2], q_1[3]) \xrightarrow{\begin{array}{c} a \quad 2 \quad a \\ | \quad / \quad \backslash \quad | \\ \langle 1, \neq, 1 \rangle \end{array}} q_3[\neq(1, 2, 3)]$$

$$R_6. \neq(q_1[1], q_3[2], q_1[3]) \xrightarrow{\begin{array}{c} a \quad 2 \quad a \\ | \quad / \quad \backslash \quad | \\ \langle 1, \neq, 1 \rangle \end{array}} q_3[\neq(1, 2, 3)]$$

$$R_7. \neq(q_0[1], q_3[2], q_0[3]) \xrightarrow{\begin{array}{c} a \quad 2 \quad a \\ | \quad / \quad \backslash \quad | \\ 1 \quad 1 \end{array}} q_f[\neq(1, 2, 3)]$$

On peut remarquer que la torsion $\langle 1, 1 \rangle$ attachée à la règle R_2 est superflue car q_0 n'est atteint qu'en un sommet \bar{a} et on pourrait la réécrire

$$R'_2 : \mathcal{E}(q_0[1], q_0[2]) \xrightarrow{\langle 1, 2 \rangle} q_2[\mathcal{E}(1, 2)]$$

ou même

$$R''_2 : \mathcal{E}(q_0[1], q_0[2]) \xrightarrow{\langle \bar{a}, \bar{a} \rangle} q_2[\mathcal{E}(1, 2)]$$

La règle R_3 (qui assure elle aussi le départ de tout mouvement de M) pourrait aussi s'écrire

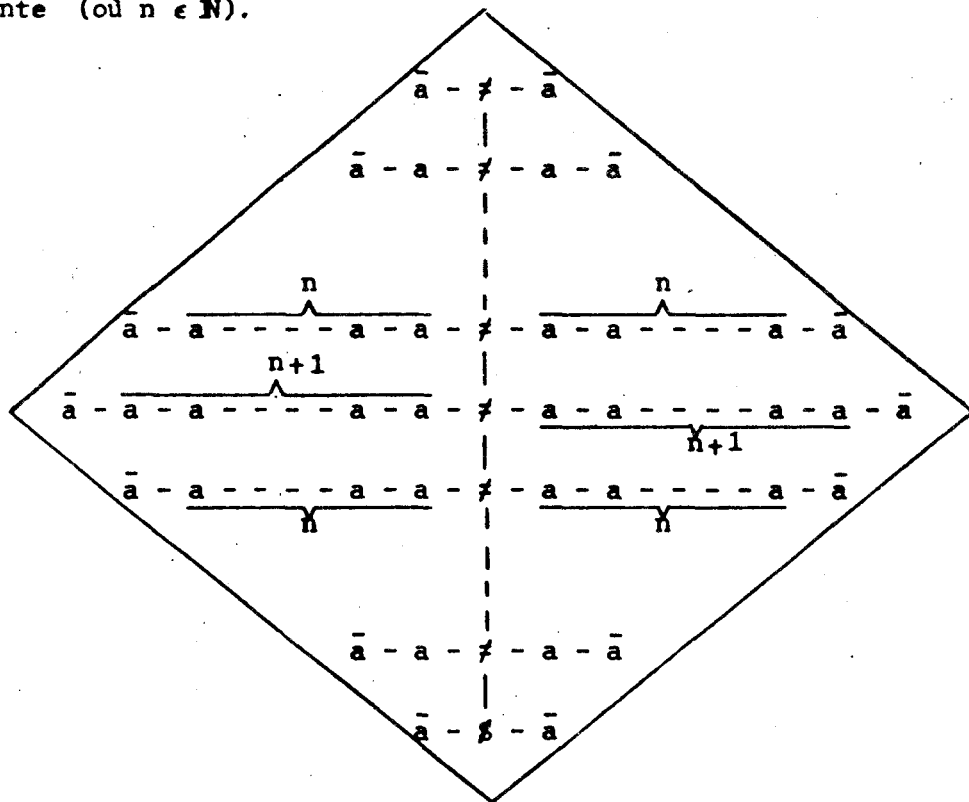
$$R'_3 : \#(q_1[1], q_2[2], q_1[3]) \xrightarrow{\begin{array}{c} \langle a, \# , a \rangle \\ | \quad / \quad \backslash \quad | \\ a \quad \bar{a} \quad \bar{a} \quad a \end{array}} q_2[\#(1, 2, 3)]$$

Enfin R_7 (qui assure le passage à l'état final q_f) pourrait s'écrire encore

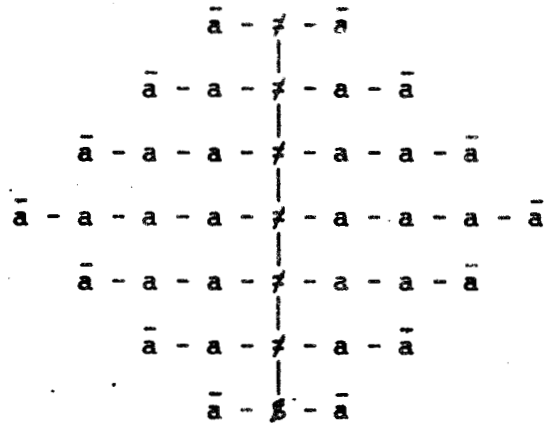
$$R'_7 : \#(q_0[1], q_3[2], q_0[3]) \xrightarrow{\begin{array}{c} \langle \bar{a}, \# , \bar{a} \rangle \\ / \quad \backslash \\ a \quad 2 \quad a \\ | \quad | \\ a \quad a \end{array}} q_f[\#(1, 2, 3)]$$

On constate aisément que la règle R_1 permet de remonter les branches $a \dots a \bar{a}$ de longueur quelconque dans le même état q_1 . Enfin R_3 assure le passage d'un " $\#$ " au " $\bar{\#}$ " précédent en ajoutant un "a" de part et d'autre tandis que R_5 et R_6 assure ce même passage en enlevant un "a" de part et d'autre de $\bar{\#}$.

Manifestement ce RATEG M reconnaît la forêt RATEG des arbres de la forme suivante (où $n \in \mathbb{N}$).



Vérifions le pour l'arbre suivant :



Dans M le mouvement reconnaissant t est le suivant :

$$\begin{array}{l}
 t \mid_{R_0^*} \neq (q_0[\bar{a}] \otimes \neq \otimes q_0[\bar{a}]).(a.q_0[\bar{a}] \otimes \neq \otimes a.q_0[\bar{a}]) \\
 .(aaq_0[\bar{a}] \otimes \neq \otimes aaq_0[\bar{a}]).(aaaq_0[\bar{a}] \otimes \neq \otimes aaaq_0[\bar{a}]) \\
 .(aaq_0[\bar{a}] \otimes \neq \otimes aaq_0[\bar{a}]).(aq_0[\bar{a}] \otimes \beta \otimes aq_0[\bar{a}]) \\
 .(q_0[\bar{a}] \otimes q_0[\bar{a}])
 \end{array}$$

$$\begin{array}{l}
 \mid_{R_1^*} \neq (q_0[\bar{a}] \otimes \neq \otimes q_0[\bar{a}]).(q_1[aa\bar{a}] \otimes \neq \otimes q_1[aa\bar{a}]) \\
 .(q_1[aaa\bar{a}] \otimes \neq \otimes q_1[aaa\bar{a}]).(q_1[aaaa\bar{a}] \otimes \neq \otimes q_1[aaaa\bar{a}]) \\
 .(q_1[aaa\bar{a}] \otimes \neq \otimes q_1[aaa\bar{a}]).(q_1[aa\bar{a}] \otimes \beta \otimes q_1[aa\bar{a}]) \\
 .(q_0[\bar{a}] \otimes q_0[\bar{a}]) \\
 = \neq.w_1.w_2.w_3.w_4.w_5.w_6.w_7
 \end{array}$$

$$\mid_{R_2} \neq.w_1.w_2.w_3.w_4.w_5.(q_1[aa\bar{a}] \otimes q_2[\beta(\bar{a}, \bar{a})] \otimes q_1[a \bar{a}])$$

$$\mid_{R_3} \neq.w_1.w_2.w_3.w_4.(q_1[aaa\bar{a}] \otimes q_2[\neq(aa\bar{a} \otimes \beta(\bar{a}, \bar{a}) \otimes aa\bar{a})] \otimes q_1[aaa\bar{a}])$$

$$\mid_{R_4} \neq.w_1.w_2.w_3.(q_1[aaaa\bar{a}] \otimes q_2[\neq(aa\bar{a} \otimes \neq(aa\bar{a} \otimes \beta(\bar{a}, \bar{a}) \otimes aa\bar{a}) \otimes aa\bar{a})] \otimes q_1[aaaa\bar{a}])$$

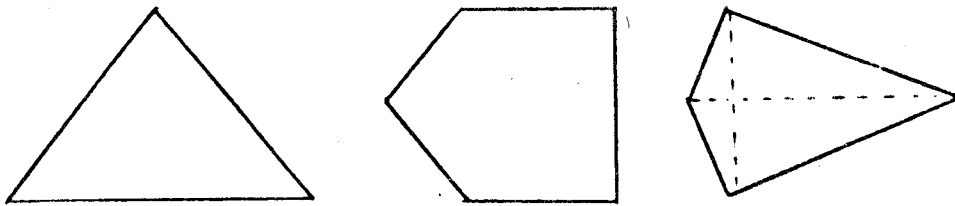
$$\begin{array}{l}
 \mid_{R_4} \neq.w_1.w_2.(q_1[aaa\bar{a}] \otimes q_2[\neq(aaaa\bar{a} \otimes \neq(aa\bar{a} \otimes \neq(aa\bar{a} \otimes \beta(\bar{a}, \bar{a}) \otimes aa\bar{a}) \otimes aa\bar{a}) \\
 \otimes aaa\bar{a})] \otimes q_1[aaa\bar{a}])
 \end{array}$$

$$\begin{array}{l}
 \mid_{R_5} \neq.w_1.(q_1[aa\bar{a}] \otimes q_3[\neq(aa\bar{a} \otimes \neq(aaaa\bar{a} \otimes \neq(aa\bar{a} \otimes \neq(aa\bar{a} \otimes \beta(\bar{a}, \bar{a}) \otimes aa\bar{a}) \\
 \otimes aa\bar{a}) \otimes aaa\bar{a}) \otimes aa\bar{a}]) \otimes q_1[aa\bar{a}])
 \end{array}$$

$$\frac{}{R_6} \neq (q_0[\bar{a}] \otimes q_3[\neq(aa \otimes \neq(aa\bar{a} \otimes \neq(aaaa \otimes \neq(aa\bar{a} \otimes \neq(aa \otimes \\ \neq(\bar{a},\bar{a}) \otimes a\bar{a}) \otimes aa\bar{a}) \otimes aaaa) \otimes aa\bar{a}) \otimes a\bar{a})]) \otimes q_0[\bar{a}])$$

$$\frac{}{R_7} q_f[t].$$

Le lecteur pourra sans aucune difficulté s'inspirer des règles précédentes pour construire les RATEG reconnaissant les forêts RATEG dont les arbres ont les formes ainsi schématisées :



et toute forme déduite des précédentes par composition et itération (opération "étoile").

IV - PROPRIÉTÉS DES RATEG

L'essentiel des propriétés de clôture se résume en le théorème suivant :

Théorème II.18

La classe des RATEG est close par union, intersection et morphisme complet et contient la classe des reconnaissables.

Ce théorème est une conséquence immédiate des propositions II.22, II.23, II.24, II.27 énoncées et démontrées ci-après.

On déduit le corollaire suivant qui concrétise le but poursuivi dans cette étude :

Corollaire II.19

La classe des RATEG contient la clôture des reconnaissables par union intersection et morphisme complet.

On conjecture la réciproque - établie au dernier chapitre dans le cas particulier des RATEG homogènes.

Soulignons une fois encore l'importance de prendre des morphismes complets ce qui assure par clôture des RATEG d'obtenir ainsi une classe de forêts incluses dans la classe des forêts récursives, puisque sans cette restriction on obtient toutes les récursivement énumérables comme on l'a vu au chapitre I.

Proposition II.20

La classe des forêts reconnues par les RATEG généralisés coïncide avec la classe des forêts RATEG.

La preuve consiste à associer à toute règle d'un RATEG généralisé M_g un nombre fini de règles non généralisées et à considérer alors le RATEG M obtenu de sorte que tout mouvement dans M_g soit associé à un mouvement composé dans M . Nous prouvons d'abord le lemme suivant.

Lemme II.21

Pour tout RATEG généralisé M_g il existe un RATEG M tel que $\forall r$ règle de M_g ,
 \exists un ensemble d'états Q_r de M vérifiant

$$i - r \neq r' \Rightarrow Q_r \cap Q_{r'} = \emptyset$$

$$ii - \alpha = v.a(q[u]) \mid \frac{r}{M_g} v.q'[a(u)] = \beta$$

$$\Leftrightarrow \alpha \mid_M^* \beta \text{ avec des états extrêmes dans } Q \text{ de } M_g$$

et des états intermédiaires dans Q_r de M .

Preuve du lemme :

Soit $M_g = (\Sigma, Q, F, R)$ et une règle r de R :

$$a(q[x]) \xrightarrow{\bar{t}} q'[a(x)]$$

avec $T = |a| =$ taille de l'arbre a .

- Si $T = 1$ (ii) est immédiat.

Supposons alors (ii) vrai pour toute règle généralisée telle que la taille de l'arbre a soit $\leq T - 1$.

- Et soit a de taille T :

$$a(x_1, x_2, \dots, x_n) = a_1(x_1, x_2, \dots, x_i, a_2(x_{i+1}, \dots, x_k), x_{k+1}, \dots, x_n)$$

avec - $|a_1| = T - 1$

$$- a_2 \in \Sigma_{k-i}$$

On construit alors les 2 règles \bar{r}_j et r_j :

- \bar{r}_j non généralisée

$$a_2(\bar{q}[\bar{x}]) \xrightarrow{\theta} \bar{q}_j[a_2(\bar{x})]$$

où $\bar{x} = (x_1, x_2, \dots, x_{k-i})$ de $(k-i)$ variables

$\bar{q} = (q_{i+1}, q_{i+2}, \dots, q_k)$ de $(k-i)$ états de Q ,

composantes de rang $i+1, i+2, \dots, k$ de q

\bar{q}_j nouvel état de Q_r vérifiant (i)

θ torsion identité

- et r_j déduite de r avec a_1 de taille $T-1$:

$$a_1((q_1, \dots, q_i, \bar{q}_j, q_{k+1}, \dots, q_n) [x_1, \dots, x_{n-k+i+1}]) \xrightarrow{\bar{t}} q'[a(x_1, x_2, \dots, x_{n-k+i+1})]$$

avec $\bar{t} = (t_1, \dots, t_i, a_2(t_{i+1}, \dots, t_k), t_{k+1}, \dots, t_n)$.

Considérons le mouvement suivant de M_g utilisant la règle r :

$$\alpha = v.a(q[u]) \Big| \frac{r}{M_g} v.q'[a(u)] = \beta$$

Avec les règles \bar{r}_j et r_j on a

$$\alpha = v.a(q[u]) = v.a(q_1[u_1], \dots, q_i[u_i], \dots, q_k[u_k], \dots, q_n[u_n])$$

$$= v.a_1(q_1[u_1], \dots, q_i[u_i], a_2(q_{i+1}[u_{i+1}], \dots, q_k[u_k]), q_{k+1}[u_{k+1}], \dots, q_n[u_n])$$

$$\overline{r}_j \alpha_1 = v.a_1(q_1[u_1], \dots, q_i[u_i], \overline{q}_j[a_2(u_{i+1}, \dots, u_k)], q_{k+1}[u_{k+1}], q_n[u_n])$$

La règle r_j portant sur a_1 de taille $T-1$, on a en appliquant l'hypothèse de récurrence

$$\begin{aligned} \alpha_1 &\stackrel{*}{\vdash} v.q'[a_1(u_1, \dots, u_i, a_2(u_i, \dots, u_k), u_{k+1}, \dots, u_n)] \\ &= v.q'[a(u)] = \beta \end{aligned}$$

$$\text{d'où } \alpha \stackrel{*}{\vdash} \overline{r}_j \alpha_1 \stackrel{*}{\vdash} \beta \text{ ou } \alpha \stackrel{*}{\vdash} \beta$$

avec des états - initiaux q dans Q

- final q' dans Q

- intermédiaires dans Q_r par hypothèse de récurrence et création de \overline{q}_j .

Il suffit alors à tout RATEG généralisé $M_g = (\Sigma, Q, F, R)$ d'associer le

RATEG $M = (\Sigma, Q', F, R')$ avec

$$- Q' = Q \cup \left(\bigcup_{r \in R} Q_r \right)$$

- R' déduit de R en remplaçant toute règle généralisée r par un nombre fini (égal à T) de règles non généralisées obtenues par la construction de récurrence, associées à un nombre fini (égal à $T-1$) de nouveaux états distincts dont la réunion est Q_r .

La preuve de la proposition est alors immédiate :

- $F(M_g) \subset F(M)$ puisque tout mouvement de M_g correspond à un mouvement composé de M menant des mêmes états initiaux à un même état final de $F \subset Q$.

$$- F(M) \subset F(M_g)$$

évident car un RATEG est un RATEG généralisé particulier (les arbres figurant en membres gauche et droit sont réduits à une lettre).

Proposition II.22

La classe des forêts reconnaissables est incluse dans la classe des forêts RATEG, qui coïncide avec la classe des forêts reconnues par RATEG généralisées.

C'est un corollaire immédiat de la remarque sur les automates d'états finis d'arbres et de la proposition précédente.

Proposition II.23

La classe des forêts RATEG est close par union.

Preuve :

Il suffit de la vérifier pour deux.

Soit F_1 et F_2 respectivement reconnues par

$$M_1 = (\Sigma_1, Q_1, F_1, R_1)$$

$$M_2 = (\Sigma_2, Q_2, F_2, R_2)$$

Alors $M = (\Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2, F_1 \cup F_2, R_1 \cup R_2)$ vérifie $F_1 \cup F_2 \subset F(M)$ de façon triviale.

Si de plus Q_1 et Q_2 ou Σ_1 et Σ_2 sont disjoints alors $F_1 \cup F_2 = F(M)$.

Proposition II.24

La classe des forêts RATEG est close par intersection.

La preuve résulte immédiatement du lemme II.25 qui à 2 forêts RATEG et leurs RATEG associés fait correspondre un RATEG reconnaissant leur intersection par la construction suivante :

Soit F et F' reconnues par $M = (\Sigma, Q, F, R)$ et $M' = (\Sigma, Q', F', R')$.

Pour tout couple de règles (r, r') de $R \times R'$ telles que

$$r : a(q[x]) \xrightarrow{t} q_1[a(x)]$$

$$r' : a(q'[x]) \xrightarrow{t'} q'_1[a(x)]$$

nous appliquons l'algorithme d'unification précédemment défini en II.A.II.1

Lorsque l'unifié $t \vee t'$ existe on associera alors à r et r' une règle \bar{r} de la forme

$$\bar{r} : a((q, q')[x]) \xrightarrow{t \ v \ t'} (q_1, q'_1) [a(x)]$$

Soit alors $\bar{M} = (\Sigma, Q \times Q', F \times F', \bar{R})$ le RATEG dont les états sont des couples d'états de M et M' et dont les règles sont déduites de R et R' selon la construction ci-dessus. On vérifie alors que tout mouvement simultané dans M et M' correspond à un mouvement dans \bar{M} et réciproquement.

Lemme II.25

$$\left. \begin{array}{l} U \xrightarrow{\frac{\star}{M}} q[U] \\ \text{et } U' \xrightarrow{\frac{\star}{M'}} q'[U] \end{array} \right\} \Leftrightarrow U \xrightarrow{\frac{\star}{\bar{M}}} (q, q') [U]$$

Preuve du lemme :

(Par induction sur la longueur du mouvement)

- $n = 1$.

$$U = a \in \Sigma_0$$

$$a \xrightarrow{r} q[a] \text{ où } r : a \rightarrow q[a]$$

$$a \xrightarrow{r'} q'[a] \text{ où } r' : a \rightarrow q'[a]$$

A r et R' on associe immédiatement

$$\bar{r} : a \rightarrow (q, q') [a]$$

d'où $a \xrightarrow{\bar{r}} (q, q') [a]$ dans \bar{M} et réciproquement.

- Supposons la propriété vraie jusqu'au rang n et soit

$$\left\{ \begin{array}{l} U \xrightarrow{\frac{\star}{M}} a(q[v]) \xrightarrow{r_{n+1}} q_1[a.v] = q_1[U] \\ \text{et} \\ U \xrightarrow{\frac{\star}{M'}} a(q'[v]) \xrightarrow{r'_{n+1}} q'_1[a.v] = q'_1[U] \end{array} \right.$$

Par hypothèse de récurrence on a

$$U \xrightarrow{\frac{\star}{\bar{M}}} a((q, q') [v])$$

Considérons alors

$$r_{n+1} : a(q[x]) \xrightarrow{t} q_1[a(x)]$$

$$r'_{n+1} : a(q'[x]) \xrightarrow{t'} q'_1[a(x)]$$

D'après les mouvements utilisant r_{n+1} et r'_{n+1} on a :

$$\exists w : v = t.w$$

$$\exists w' : v = t'.w'$$

La simultanéité des mouvements entraîne nécessairement l'existence de l'unifié $t \vee t'$ de t et t' et on déduit alors la règle \bar{r}_{n+1} :

$$\bar{r}_{n+1} : a((q, q')[x]) \xrightarrow{t \vee t'} (q_1 \ q'_1) [a(x)]$$

D'autre part

$$\exists \bar{w} : v = t.w = t'.w' = (t \vee t').\bar{w}$$

et on a le mouvement

$$a((q, q')[v]) \xrightarrow{\bar{r}_{n+1}} (q_1 \ q'_1) [a(v)]$$

soit

$$U \xrightarrow[\bar{M}]{*} a((q, q')[v]) \xrightarrow{\bar{r}_{n+1}} (q_1 \ q'_1) [a(v)]$$

la propriété est donc vraie $\forall n$:

$$\left. \begin{array}{l} U \xrightarrow{*} q[U] \\ \text{et } U \xrightarrow{*} q'[U] \end{array} \right\} \Rightarrow U \xrightarrow{*} (q, q') [U]$$

- Réciproquement :

Supposons la relation vraie pour une longueur n et soit dans \bar{M} :

$$a.U \xrightarrow{*} a(q, q') [U] \xrightarrow{\bar{r}_{n+1}} (q_1 \ q'_1) [a.U]$$

La dernière règle s'écrit

$$\bar{r}_{n+1} : a((q, q')[x]) \xrightarrow{\bar{t}} (q_1 \ q'_1) [a(x)]$$

$$\text{et } \exists \bar{w} : U = \bar{t}.\bar{w}$$

avec \bar{r}_{n+1} déduite de 2 règles r_{n+1} et r'_{n+1} de la forme

$$r_{n+1} : a(q[x]) \xrightarrow{t} q_1[a(x)]$$

$$r'_{n+1} : a(q'[x]) \xrightarrow{t'} q'_1[a(x)]$$

où $t < \bar{t}$ et $t' < \bar{t}$.

Il existe donc w et w' tel que

$$U = \bar{t}.\bar{w} = t.w = t'.w'$$

On peut donc appliquer r_{n+1} et r'_{n+1} , soit dans M et M' :

$$a.U \stackrel{*}{\vdash} a.q[U] \stackrel{r_{n+1}}{\vdash} q_1[a(U)]$$

$$a.U \stackrel{*}{\vdash} a.q'[U] \stackrel{r'_{n+1}}{\vdash} q'_1[a(U)]$$

et la propriété est vraie $\forall n$.

Corollaire II.26

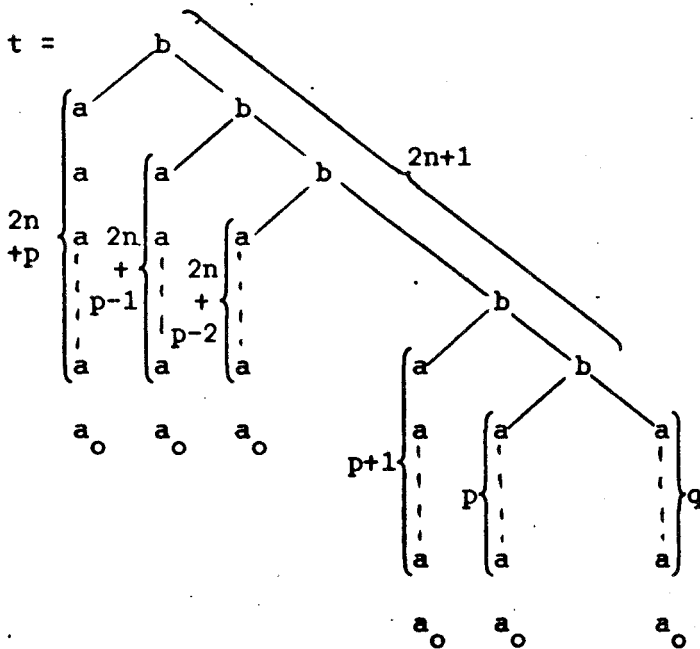
La classe des forêts RATEG est close par intersection avec une forêt reconnaissable.

(Immédiat puisque $\text{Rec} \subset \text{RATEG}$).

Exemple II.12 : Forêt RATEG F intersection de 2 forêts RATEG F_1 et F_2 reconnues par R_1 et R_2 et construction par unification des règles du RATEG R reconnaissant F à partir des règles de R_1 et R_2 .

Soient les 2 forêts F_1 et F_2 construites sur $\Sigma = \{a_0, a(1), b(1, 2)\}$ dont les arbres ont une branche principale constituée d'un nombre impair de "b" et de la forme respective suivante

De façon évidente leur intersection $F = F_1 \cap F_2$ sera constitué d'arbres vérifiant la forme



avec p et q quelconques.

On peut définir assez aisément les RATEG non déterministes

$R_1 = (\Sigma, Q_1, F_1, R_1)$ et $R_2 = (\Sigma, Q_2, F_2, R_2)$ reconnaissant F_1 et F_2 :

- pour R_1 :

$$Q_1 = \{q_0, q_1, q_2, q_3\}$$

$$F_1 = \{q_2\}$$

R_1 :

$$R_1^0 \quad a_0 \rightarrow q_0[a_0]$$

$$R_1^1 \quad a(q_0[1]) \xrightarrow{\langle 1 \rangle} q_0[a(1)] \mid q_1[a(1)]$$

$$R_1^2 \quad b((q_0, q_0) [1, 2]) \xrightarrow{\langle 1, 2 \rangle} q_2[b(1, 2)]$$

$$R_1^3 \quad b((q_1, q_2) [1, 2]) \xrightarrow{\begin{array}{c} (a, \quad b) \\ | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array}} q_3[b(1, 2)]$$

$$R_1^4 \quad b((q_0, q_3) [1, 2]) \xrightarrow{\langle 1, 2 \rangle} q_2[b(1, 2)]$$

(s'appliquant en alternance avec la précédente).

- Pour R_2 :

$$Q_2 = \{k_0, k_1, k_2, k_3\}$$

$$F_2 = \{k_2\}$$

R_2 :

$$R_2^0 \quad a_0 \rightarrow k_0[a_0]$$

$$R_2^1 \quad a(k_0[1]) \xrightarrow{\langle 1 \rangle} k_0[a(1)] \mid k_1[a(1)]$$

$$R_2^2 \quad b((k_0, k_0) [1, 2]) \xrightarrow{\langle 1, 2 \rangle} k_2[b(1, 2)]$$

$$R_2^3 \quad b((k_0, k_2) [1, 2]) \xrightarrow{\langle 1, 2 \rangle} k_3[b(1, 2)]$$

$$R_2^4 \quad b((k_1, k_3) [1, 2]) \xrightarrow{\begin{array}{c} (a, \quad b) \\ | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array}} k_2[b(1, 2)]$$

(en alternance avec la précédente).

On considère alors les règles s'appliquant simultanément à un même noeud α

d'un arbre t de F c'est à dire les règles R_1^i et R_2^i pour $0 \leq i \leq 4$ de

la forme

$$\alpha(q'[x]) \xrightarrow{T_1} q[\alpha(x)]$$

$$\alpha(k'[x]) \xrightarrow{T_2} k[\alpha(x)]$$

On réalise alors l'unification de T_1 et T_2 selon l'algorithme II.A.II.1

d'où les règles suivantes :

$$R^0 \quad a_0 \rightarrow (q_0, k_0) [a_0]$$

$$R^1 \quad a((q_0, k_0) [1]) \xrightarrow{\langle 1 \rangle} (q_0, k_0)[a(1)] \mid (q_0, k_1)[a(1)] \mid (q_1, k_0)[a(1)]$$

$$R^2 \quad b(((q_0, k_0), (q_0, k_0)) [1, 2]) \xrightarrow{\langle 1, 2 \rangle} (q_2, k_2) [b(1, 2)]$$

$$\frac{\vdash}{R^2} b((q_0, k_1)[a^4 a_0], b((q_1, k_0)[a^3 a_0], b((q_0, k_1)[a^2 a_0], b((q_1, k_0)[aa_0], (q_2, k_2)[b(a_0, a^2 a_0)]))))))$$

$$\frac{\vdash}{R^3} b((q_0, k_1)[a^4 a_0], b((q_1, k_0)[a^3 a_0], b((q_0, k_1)[a^2 a_0], (q_3, k_3)[b(aa_0, b(a_0, a^2 a_0)]))))))$$

$$\frac{\vdash}{R^4} b((q_0, k_1)[a^4 a_0], b((q_1, k_0)[a^3 a_0], (q_2, k_2)[b(a^2 a_0, b(aa_0, b(a_0, a^2 a_0)]))))))$$

$$\frac{\vdash}{R^3} b((q_0, k_1)[a^4 a_0], (q_3, k_3)[b(a^3 a_0, b(a^2 a_0, b(aa_0, b(a_0, a^2 a_0)]))))))$$

$$\frac{\vdash}{R^2} (q_2, k_2)[b(a^4 a_0, b(a^3 a_0, b(a^2 a_0, b(aa_0, b(a_0, a^2 a_0)])))))) \\ = (q_2, k_2) [t] \quad \text{et } (q_2, k_2) \in F.$$

Le lecteur généralisera aisément les mouvements obtenus pour $n \geq p$ et q quelconques.

Proposition II.27

La classe des forêts RATEG est close par homomorphisme complet.

La preuve résulte des lemmes II.30 et II.32. On obtient alors pour image par homomorphisme ϕ complet d'une forêt RATEG F une forêt reconnue par un RATEG généralisé M_g . Or on sait associer à M_g un RATEG M tel que $F(M_g) = F(M)$. On a donc $\phi(F) = F(M_g) = F(M)$.

Corollaire II.28

La classe des forêts RATEG est close par démarquage propre.

Immédiat puisque par définition un démarquage propre est un morphisme complet strict qui ne modifie que les labels.

Corollaire II.29

Les images par homomorphisme complet des forêts reconnaissables sont des forêts RATEG.

On a $\text{Rec} \subset \text{RATEG}$ et $\phi(\text{RATEG}) \subset \text{RATEG}$ d'où $\phi(\text{Rec}) \subset \text{RATEG}$.

On retrouve ici le fait que par des morphismes non linéaires on génère des égalités que ne peuvent tester des automates d'états finis d'arbres (d'où la non-clôture par homomorphisme de Rec) mais vérifiables par les RATEG.

Il importe de noter une fois encore que plus que la strictitude c'est la complétude des morphismes qui joue ici un rôle essentiel de conservation dans l'image d'un arbre de toutes les informations qu'il contient. On peut faire un parallèle avec le rôle des morphismes non effaçants en théorie des langages dans le cas des Quasi Realtime Languages par exemple [BG].

Notation

Etant donné un n-uple d'états $q = (q_1, q_2 \dots q_n) \in Q^n$ et une torsion ρ de Θ_n^p nous définissons l'opération notée \cdot de $\Theta_n^p \times Q^n \rightarrow Q^p$:

$$\rho \cdot q = (q_{\rho(1)} \dots q_{\rho(p)})$$

Lemme II.30

L'image par morphisme complet strict d'une forêt RATEG est reconnue par un RATEG généralisé.

Preuve :

Soit F reconnue par $M = (\Sigma, Q, F, R)$ et ϕ complet strict $\phi : \Sigma \rightarrow \Delta$.

A partir de chaque règle r de R et de ϕ nous allons construire une règle \bar{r}_g par le procédé suivant :

Soit r de M :

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$

avec $a \in \Sigma_n$

$$q \in Q^n$$

$$x = (x_1 \dots x_n)$$

$$t \in T(\Sigma)_\delta^n, t = (t_1, t_2 \dots t_n)$$

$$\text{et si } \delta_i \text{ est le degré de } t_i \text{ pour } i \in [n] \quad \delta = \sum_{i=1}^n \delta_i$$

$$\text{Soit } \phi(a) = t_a = \tilde{t}_a \cdot \rho$$

$$\begin{array}{c} / \quad \backslash \\ x_1 \quad x_n \end{array}$$

avec $t_a, \tilde{t}_a \in T(\Delta)_p^1$ et $\rho \in \Theta_n^D$

(Rappelons que les fibres sont conservées par ϕ).

D'autre part ϕ complet strict signifie que

- t_a contient au moins une lettre de Δ
- et que tous les $x_1 \dots x_n$ se retrouvent dans $\phi(a)$.

Soit $\forall i \in [u] \quad \phi(t_i) = t'_i$ de degré δ'_i

et $\phi(t) = t' = (t'_1 t'_2 \dots t'_n) \in T(\Sigma)_\delta^n$,

(où $\delta' = \sum_{i=1}^n \delta'_i$)

Nous construisons alors la règle généralisée \bar{r}_g :

$$\tilde{t}_a(\rho \cdot q[\bar{x}]) \xrightarrow{\rho \cdot \phi(t)} q'[\tilde{t}_a(\bar{x})]$$

avec $\tilde{t}_a \in T(\Sigma)_p^1$

$$\rho \cdot q \in Q^D$$

$$\bar{x} = (x_1, \dots, x_p)$$

$$\rho \cdot \phi(t) = (t'_{\rho(1)} \dots t'_{\rho(p)}) \in T(\Sigma)_{\delta''}^D, \text{ avec } \delta'' = \sum_{i=1}^p \delta'_{\rho(i)}.$$

Soit alors R' l'ensemble des règles déduites de R par cette construction

et soit $M_g = (\Delta, Q, F, R_g)$ le RATEG généralisé associé à M .

Le sous-lemme II.31 montre alors qu'à tout mouvement de M reconnaissant

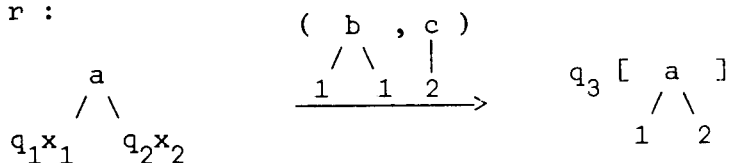
u de $T(\Sigma)_0^1$ correspond un mouvement de M_g reconnaissant $\phi(u)$ et réciproquement.

Et par conséquent

$$F = F(M) \iff \phi(F) = F(M_g)$$

Exemple II.13 : Construction de \bar{r}_g

Soit r :



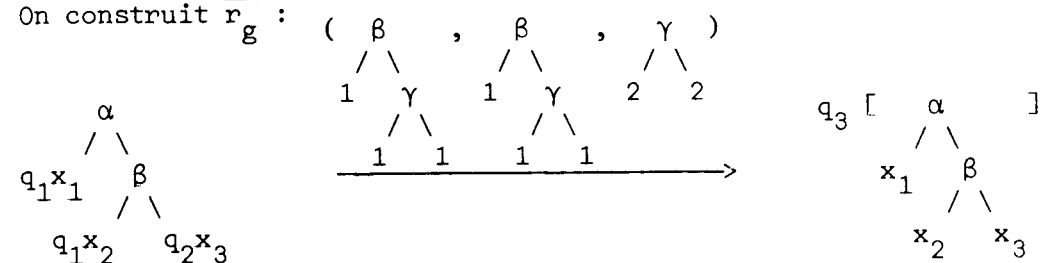
et ϕ :

$$\phi \left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ 1 \quad \beta \\ \quad / \quad \backslash \\ \quad 1 \quad 2 \end{array} = \begin{array}{c} \alpha \\ / \quad \backslash \\ 1 \quad \beta \\ \quad / \quad \backslash \\ \quad 2 \quad 3 \end{array} \cdot \underbrace{\langle 2 ; 1, 1, 2 \rangle}_{\rho}$$

$$\phi \left(\begin{array}{c} b \\ / \quad \backslash \\ 1 \quad 2 \end{array} \right) = \begin{array}{c} \beta \\ / \quad \backslash \\ 1 \quad \gamma \\ \quad / \quad \backslash \\ \quad 2 \quad 2 \end{array}$$

$$\phi \left(\begin{array}{c} c \\ | \\ 1 \end{array} \right) = \begin{array}{c} \gamma \\ / \quad \backslash \\ 1 \quad 1 \end{array}$$

On construit \bar{r}_g :



avec

$$(q_1 \ q_1 \ q_2) = \langle 2 ; 1, 1, 2 \rangle \cdot (q_1, q_2) = \rho \cdot q$$

et

$$\begin{aligned}
 \left(\begin{array}{c} \beta \\ / \quad \backslash \\ 1 \quad \gamma \\ / \quad \backslash \\ 1 \quad 1 \end{array}, \begin{array}{c} \beta \\ / \quad \backslash \\ 1 \quad \gamma \\ / \quad \backslash \\ 1 \quad 1 \end{array}, \begin{array}{c} \gamma \\ / \quad \backslash \\ 2 \quad 2 \end{array} \right) &= \langle 2 ; 1, 1, 2 \rangle \cdot \left(\begin{array}{c} \beta \\ / \quad \backslash \\ 1 \quad \gamma \\ / \quad \backslash \\ 1 \quad 1 \end{array}, \begin{array}{c} \gamma \\ / \quad \backslash \\ 2 \quad 2 \end{array} \right) \\
 &= \rho.\phi(t)
 \end{aligned}$$

Sous-lemme II.31

$$\left. \begin{array}{l} \exists U \in T(\Sigma)_0^1 \text{ tel que } V = \phi(U) \\ \text{et } U \stackrel{*}{\underset{M}{\mid}} \underset{r}{\mid} q'[U] \end{array} \right\} \Leftrightarrow V \stackrel{*}{\underset{M_g}{\mid}} \underset{\bar{r}_g}{\mid} q'[V]$$

Preuve du sous-lemme

(par récurrence sur la longueur du mouvement).

=>.

- Longueur 1

$$U = a(\in \Sigma_0) \stackrel{*}{\underset{M}{\mid}} \underset{r}{\mid} q'[a] \text{ avec } r : a \rightarrow q'[a].$$

On associe de façon triviale à r la règle

$$\bar{r}_g : \phi(a) \rightarrow q'[\phi(a)]$$

$$\text{d'où } \phi(U) = V \stackrel{*}{\underset{M_g}{\mid}} \underset{\bar{r}_g}{\mid} q'[V]$$

- Si la propriété est vraie pour tout mouvement de longueur $\leq l$, soit

$$U = a.v = a(v_1, v_2 \dots v_n) \stackrel{*}{\underset{M}{\mid}} a(q[v]) \stackrel{*}{\underset{r}{\mid}} q'[a(v)]$$

$$\text{avec } q = (q_1, q_2 \dots q_n)$$

$\forall i \in [n]$ le mouvement affectant l'état q_i au sommet de V_i vérifie

l'hypothèse de récurrence soit

$$v_i \stackrel{*}{\underset{M}{\mid}} \underset{r_i}{\mid} q_i[v_i] \Leftrightarrow$$

$$\phi(v_i) \stackrel{*}{\underset{M_g}{\mid}} \underset{\bar{r}_g}{\mid} q_i[\phi(v_i)]$$

Alors

$$\text{si } \phi(a.v) = \phi(a).\phi(v) = t_a.\phi(v) = \tilde{t}_a.\rho.\phi(v)$$

on a

$$U \stackrel{*}{\underset{M}{\vdash}} a(q[v]) \Rightarrow \phi(U) \stackrel{*}{\underset{M_g}{\vdash}} \tilde{t}_a(\rho.q[\rho.\phi(v)])$$

Par construction de \bar{r}_g associée à r on a

$$r : a(q[x]) \xrightarrow{t} q'[a(x)]$$

$$\bar{r}_g : \tilde{t}_a(\rho.q[\bar{x}]) \xrightarrow{\rho.\phi(t)} q'[\tilde{t}_a(\bar{x})]$$

D'autre part le mouvement

$$a(q[v]) \stackrel{r}{\vdash} q'[a(v)]$$

a lieu ssi $\exists w : v = t.w$

On a alors $\phi(v) = \phi(t.w) = \phi(t).\phi(w)$

$$\text{et } \rho.\phi(v) = \rho.\phi(t).\phi(w)$$

On peut par conséquent appliquer la règle \bar{r}_g et on a

$$\tilde{t}_a(\rho.q[\rho.\phi(v)]) \stackrel{\bar{r}_g}{\vdash} q'[\tilde{t}_a(\rho.\phi(v))]$$

c'est à dire

$$\phi(a.v) \stackrel{*}{\underset{M_g}{\vdash}} \tilde{t}_a(\rho.q[\rho.\phi(v)]) \stackrel{\bar{r}_g}{\vdash} q'[\phi(a.v)]$$

$$\text{ou } \phi(U) \stackrel{*}{\vdash} \vdash q'[\phi(U)]$$

et par conséquent la propriété est vérifiée pour tout mouvement.

<=.

Pour un mouvement de longueur 1 on a

$$V \in T(\Delta)_O^1$$

$$\text{tel que } V \stackrel{*}{\underset{M_g}{\vdash}} q'[V]$$

les règles de M_g étant déduites de celles de M on a nécessairement

$$\exists U \in \Sigma_O : \phi(U) = V$$

$$\text{et de façon triviale } U \xrightarrow{r} q'[U]$$

$$\text{soit } U \stackrel{*}{\underset{M}{\vdash}} q'[U]$$

Supposons alors la propriété vraie pour tout mouvement de M_g de longueur

$\leq \ell$ et soit

$$v = a'' \cdot v'' \stackrel{*}{\underset{M}{\vdash}} a''(q''[v'']) \stackrel{\bar{r}_g}{\vdash} q'[a'' \cdot v''] = q'[v]$$

où $q'' \in Q^p$

$$v'' = (v''_1, v''_2 \dots v''_p)$$

$$a'' \in T(\Delta)_p^1$$

Les mouvements affectant les états q''_i aux sommets des v''_i pour tout $i \in [p]$ sont de longueur $\leq \ell$ et par hypothèse de récurrence on a

$$v''_i \stackrel{*}{\vdash} q''_i [v''_i] \Leftrightarrow \exists u'_i : \phi(u'_i) = v''_i \text{ et } u'_i \stackrel{*}{\vdash} q''_i [u'_i]$$

La règle \bar{r}_g appliquée dans

$$a''(q''[v'']) \stackrel{\bar{r}_g}{\vdash} q'[a'' \cdot v'']$$

se déduit d'une règle r de M de la forme

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$

et s'écrit alors

$$\bar{r}_g : a''(\rho \cdot q[x'']) \xrightarrow{\rho \cdot \phi(t)} q'[a''(x'')]$$

avec $\phi(a) = a' = a'' \cdot \rho$ $a \in \Sigma_n$

$$q'' = \rho \cdot q \quad q \in Q^n$$

$$t' \in T(\Sigma)_\delta^n$$

Considérons alors l'arbre u de $T(\Sigma)_\delta^n$ formé des composantes de

$\{u'_i \mid i \in [p]\}$ tel que

$$\rho \cdot \phi(u) = (\phi(u_{\rho(1)}), \dots, \phi(u_{\rho(p)}))$$

$$= (\phi(u'_1) \dots \phi(u'_p)) = (v''_1 \dots v''_p) = v''.$$

Le dernier mouvement de M_g utilise la règle \bar{r}_g donc

$\exists w'' \in T(\Delta)_\delta^p$ tel que

$$v'' = \rho \cdot \phi(t) \cdot w'' = \rho \cdot \phi(u)$$

d'où $\phi(u) = \phi(t) \cdot w''$

On sait que ϕ est complet, donc

$\exists w \in T(\Sigma)_\delta^n$ tel que $\phi(w) = w''$

soit $\phi(u) = \phi(t) \cdot \phi(w) = \phi(t \cdot w)$

et par conséquent $u = t. w$

On pourra donc appliquer la règle r :

Soit $U = a.u$ tel que $\phi(U) = V$. On aura le mouvement suivant

$$U \xrightarrow{*} a(q[u]) \xrightarrow{r} q'[a(u)] = q'[U].$$

La propriété est donc vraie quelle que soit la longueur ℓ du mouvement.

Lemme II.32

L'image par morphisme complet non strict d'une forêt RATEG est reconnue par un RATEG.

Preuve :

On pourra toujours se ramener au cas d'un morphisme dont la non-strictitude porte sur une seule lettre de degré 1 c'est à dire

- ϕ est l'identité sur $\Sigma - \{a\}$
- $\phi(a) = 1$

La preuve se fait alors en 2 étapes ; on construit d'abord un RATEG généralisé M_g avec un nombre infini de règles obtenues à partir de l'algorithme d'unification qui reconnaît la même forêt initiale F puis un 2e RATEG simple \tilde{M} déduit de M_g par ϕ qui reconnaît $\phi(F)$.

On a alors

$$F = F(M) = F(M_g)$$

$$\text{et } \phi(F) = \phi(F(M_g)) = F(\tilde{M})$$

Sous-lemme II.33

A toute séquence de mouvements simples de M utilisant des règles $r, r_{i_1}, r_{i_2}, \dots, r_{i_n}$ avec a en membre gauche des r_{i_j} , on associe un mouvement

simple de RATEG généralisé M_g utilisant une règle r_g dont l'arbre t_u s'obtient par unification des arbres $t, t_{i_1}, t_{i_2} \dots t_{i_n}$ de $r, r_{i_1}, r_{i_2} \dots r_{i_n}$.

i.e. :

$$\frac{aa \dots aa'(q[u])}{\underset{n}{\vee}} \stackrel{*}{M} q'[a \dots a a'.u]$$

où $a' \in \Sigma_p, a' \neq a$

\Leftrightarrow

$$\frac{a \dots a a'(q[u])}{\underset{n}{\vee}} \stackrel{r_g}{M_g} q'[a \dots a a'.u]$$

Preuve du sous-lemme :

(Par induction sur le nombre n de règles r_{i_j} avec a en membre gauche).

- $n = 1$

On a dans M

$$a.a'(q[u]) \stackrel{r}{\vdash} a q''[a'.u] \stackrel{r_{i_1}}{\vdash} q'[a.a'.u]$$

où $r : a'(q[x]) \xrightarrow{t} q''[a'.x]$

$$t \in T(\Sigma)_{\delta}^p, q \in Q^p$$

$$r_{i_1} : a(q''[x_1]) \xrightarrow{t_{i_1}} q'[a(x_1)]$$

$$t_{i_1} \in T(\Sigma)_{\delta'}^1, q'' \in Q$$

L'application successive de r et r_{i_1} entraîne

$$u = t.v$$

$$a'.u = t_{i_1}.w = a'.t.v$$

On réalise alors l'unification de t_{i_1} et $a'.t$ en un arbre unique

$$t_u = t_{i_1} \vee a'.t \in T(\Sigma)_{\delta''}^1 \text{ tel que}$$

$$a'.u = t_{i_1}.w = a'.t.v = t_u.s$$

et on définit r_g :

soit t'_u défini par $t'_u = a'.t'_u$. On a

$$r_g : a.a'(q[x]) \xrightarrow{t'_u} q'[aa'(x)]$$

d'où le mouvement dans M_g :

$$aa'(q[u]) \xrightarrow{r_g} q'[aa'(u)].$$

- Réciproquement

$$\text{soit } aa'(q[u]) \xrightarrow{r_g} q'[a.a'(u)]$$

alors $u = t'_u.s$

Comme t'_u résulte de l'unification de $a'.t$ et t_{i_j} de 2 règles de M on a

$$-(t)_k < (t'_u)_k \quad \forall k \in [\delta] \quad (\delta = \text{degré de } a')$$

donc $\exists v$ tel que

$$a'.u = a'.t'_u.s = a'.t.v$$

$$\text{et } a'(q[u]) \xrightarrow{r} q''[a'(u)]$$

De même

$$t_{i_j} < a'.t'_u$$

donc $\exists w$ tel que

$$a'.u = a'.t'_u.s = t_{i_j}.w$$

$$\text{et } a(\bar{q}''[a'.u]) \xrightarrow{r_{i_j}} q'[a.a' u]$$

- Supposons la propriété vraie jusqu'au rang n c'est à dire :

$$\frac{a \dots a}{\underbrace{\quad}_n} a'(q[u]) \xrightarrow{M} q'[\frac{a \dots a}{\underbrace{\quad}_n} a' u]$$

\Leftrightarrow

$$\frac{a \dots a}{\underbrace{\quad}_n} a'(q[u]) \xrightarrow{M_g} q'[\frac{a \dots a}{\underbrace{\quad}_n} a' u]$$

où r_g et t'_u sont définis comme précédemment.

Considérons dans M le rang $n+1$ du mouvement

$$a(q'[\frac{a \dots a}{\underbrace{\quad}_{n+1}} a' u]) \xrightarrow{\quad} q''[\frac{a \dots a}{\underbrace{\quad}_{n+1}} a' u]$$

utilisant $r_{i_{n+1}}$:

$$a(q'[x]) \xrightarrow{t_{i_{n+1}}} q''[a(x)]$$

avec $t_{i_{n+1}} \in T(\Sigma)_{\delta}^1$ de profondeur $p(t_{i_{n+1}}) \leq p$.

où p est la profondeur maximale des arbres t associés aux règles r de M .

2 cas peuvent se produire : $n < p$ et $n \geq p$

1er cas : $n \geq p$.

$p(t_{i_{n+1}}) \leq p \leq n$ et nécessairement $t_{i_{n+1}}$ est constitué d'une branche de p "a" au plus et $\delta = 1$. Il n'y a plus d'unification à faire et à

$(r_{i_1} \dots r_{i_n} r_{i_{n+1}})$ on associe une nouvelle règle r'_g où $t'_u = t_u$ de r_g soit

$$r'_g : \frac{a \dots a}{n+1} q'(q[x]) \xrightarrow{t'_u} q''[\frac{a \dots a}{n+1} a'(x)]$$

Dans M_g on a alors le mouvement simple

$$\frac{a \dots a}{n+1} q'(q[u]) \xrightarrow[M_g]{} q''[\frac{a \dots a}{n+1} a'(u)]$$

2e cas : $n < p$

a) ou $p(t_{i_{n+1}}) \leq n$

alors $t_{i_{n+1}}$ est une branche de $p(t_{i_{n+1}})$ "a" et $\delta = 1$. On construit alors r'_g et t'_u comme dans le 1er cas et on a la même conclusion.

b) ou $p(t_{i_{n+1}}) > n$

alors nécessairement $t_{i_{n+1}}$ est de la forme

$$t_{i_{n+1}} = \frac{a \dots a}{n} a' t'_{i_{n+1}}$$

On réalise alors l'unification de t_u associé à $(t_{i_1} \dots t_{i_n})$ de

l'étape précédente et $t'_{i_{n+1}}$ en un arbre t'_u puis on déduit la règle

$$r'_g : \frac{a \dots a}{n+1} a'(q[x]) \xrightarrow{t'_u} q''[\frac{a \dots a}{n+1} a'(x)]$$

d'où le mouvement simple dans M_g :

$$\frac{a \dots a}{n+1} a'(q[u]) \xrightarrow{r'_g} [q'' a \dots a a'(u)]$$

Réciproquement : soit dans M_g

$$\frac{a \dots a}{n+1} a'(q[u]) \vdash q''[\frac{a \dots a}{n+1} a'(u)]$$

utilisant r'_g associée à l'arbre t'_u , déduite de règles $(r_{r_i}, \dots, r_{i_{n+1}})$

associées à $t, t_{i_1}, \dots, t_{i_{n+1}}$ et telle que $u = t'_u.s$.

L'unification entraîne

$$\forall k (t)_k < (t'_u)_k$$

et $\forall j \in [n+1]$

$$t_{i_j} < \frac{a \dots a}{j} . a' . t'_u$$

donc - $\exists v : u = t'_u.s = t.v$

$$\text{et } a'(q[u]) \xrightarrow{r} q_1[a(u)]$$

- $\forall j \in [n+1] \exists w_j :$

$$\frac{a \dots a}{j} a' u = \frac{a \dots a}{j} a' . t'_u . s = t_{i_j} . w$$

et

$$a(q_j[\frac{a \dots a}{j} a'(u)]) \xrightarrow{r_{i_j}} q_{j+1}[\frac{a \dots a}{j+1} a'u]$$

d'où dans M le mouvement composé

$$\frac{a \dots a}{n+1} a'(q[u]) \xrightarrow{r} \frac{a \dots a}{n+1} q_1[a'.u]$$

$$\xrightarrow{r_{i_1}} \frac{a \dots a}{n} q_2[a a' u] \quad \xrightarrow{r_{i_2}} \frac{a \dots a}{n-1} q_3[a a a' u]$$

$$\dots \xrightarrow{r_{i_{n+1}}} q''[\frac{a \dots a}{n+1} a' u]$$

Soit alors $M_g = (\Sigma, Q, F, R_g)$ le RATEG généralisé déduit de M avec

$R_g = \{\text{règles de } M \text{ dont le membre gauche } \in \Sigma - \{a\}\}$

u $\{r_g$ déduites de r par la construction du sous-lemme II.33}

On a immédiatement

$$F(M) = F(M_g)$$

avec pour M_g un nombre infini de règles généralisées r_g .

Considérons alors dans une 2e étape les images par le morphisme ϕ complet non strict des règles de M_g :

- A toute règle r dont le membre gauche appartient à $\Sigma - \{a\}$ de la forme

$$r : \alpha(q[x]) \xrightarrow{t} q'[\alpha(x)]$$

on associe

$$\bar{r} : \alpha(q[x]) \xrightarrow{\phi(t)} q'[\alpha(x)]$$

- A toute règle r_g généralisée

$$r_g : \underbrace{a \dots a}_n a'(q[x]) \xrightarrow{t} q'[\underbrace{a \dots a}_n a'(x)]$$

on associe

$$\bar{r}_g : a'(q[x]) \xrightarrow{\phi(t)} q'[a'(x)]$$

D'après la construction de r_g on sait que t reste le même si le nombre n de "a" précédant a' est $> p$. Par conséquent toutes les règles r_g telles que $n > p$ auront une image unique \bar{r}_g :

$$a'(q[x]) \xrightarrow{\phi(t)} q'[a'(x)]$$

Soit alors \bar{R} l'ensemble fini des règles simples ainsi obtenues et

$\bar{M} = (\Sigma - \{a\}, Q, F, \bar{R})$ le RATEG correspondant.

Nous allons montrer que

$$F = F(M_g) \Leftrightarrow \phi(F) = F(\bar{M})$$

Sous-lemme II.34 :

$\exists U \in T(\Sigma)_0^1$ tel que $V = \phi(U)$

et $U \xrightarrow{\frac{*}{M_g}} q'[U]$

$$\Leftrightarrow V \xrightarrow{\frac{*}{\bar{M}}} q'[V]$$

Preuve

(par récurrence sur la longueur du mouvement)

=>.

Longueur 1 :

ou $U \in \Sigma_0 \subset \Sigma - \{a\}$ et immédiatement

$$U \stackrel{M}{\underset{g}{\vdash}} q'[U] \Leftrightarrow U \stackrel{\bar{M}}{\vdash} q'[U]$$

ou $U = \underbrace{a \dots a}_n a_0$ avec $a_0 \in \Sigma_0$. Le mouvement utilise alors une règle

définie de façon unique telle que

$$\underbrace{a \ a \ \dots \ a \ a_0}_n \longrightarrow q'[a \ \dots \ a \ a_0]$$

et on a \bar{r} déduite de r_g :

$$a_0 \longrightarrow q'[a_0]$$

$$\text{d'où } \phi(U) \vdash q'[\phi(U)]$$

Supposons l'équivalence vraie pour tout mouvement de longueur ℓ et soit1er cas :

$$U = \alpha(u_1 \ u_2 \ \dots \ u_n) \stackrel{M}{\underset{g}{\vdash}^*} \alpha(q[u]) \stackrel{r}{\vdash} q'[\alpha(u)]$$

avec $\alpha \in \Sigma - \{a\}$

Par hypothèse de récurrence on a

$$\phi(U) \stackrel{M}{\underset{\bar{M}}{\vdash}^*} \alpha(q[\phi(u)])$$

de plus le mouvement

$$\phi(q[u]) \stackrel{r}{\vdash} q'[\alpha(u)]$$

a lieu ssi

$$\exists w \quad u = t.w$$

On a alors $\phi(u) = \phi(t).\phi(w)$ La règle \bar{r} associée à r soit

$$\bar{r} : \alpha(q[x]) \xrightarrow{\phi(t)} q'[\alpha(x)]$$

peut alors s'appliquer et on a

$$\alpha(q[\phi(u)]) \stackrel{\bar{r}}{\vdash} q'[\alpha(\phi(u))]$$

$$\text{d'où } \phi(U) \stackrel{*}{\vdash}_{\bar{M}} \alpha(q[\phi(u)]) \stackrel{\bar{r}}{\vdash} q[\phi(U)]$$

2e cas

$$U = \underbrace{a \dots a}_p a'(u_1 \dots u_n) \stackrel{*}{\vdash}_{\bar{M}_g} \underbrace{a \dots a}_p a'(q[u]) \stackrel{r_g}{\vdash} q'[\underbrace{a \dots a}_p a'(u)]$$

Par hypothèse de récurrence on a

$$\phi(U) \stackrel{*}{\vdash}_{\bar{M}} \alpha(q[\phi(u)])$$

De plus

$$\underbrace{a \dots a}_p a'(q[u]) \stackrel{r_g}{\vdash} q'[\underbrace{a \dots a}_p a'(u)]$$

a lieu ssi $\exists w : u = t.w$

Alors $\phi(u) = \phi(t).\phi(w)$

On peut donc appliquer la règle \bar{r}_g

$$\bar{r}_g : a'(q[x]) \xrightarrow{\phi(t)} q'[a'(x)]$$

$$\text{soit } a'(q[\phi(u)]) \stackrel{\bar{r}_g}{\vdash} q'[a'(\phi(u))]$$

$$\text{soit } \phi(U) \stackrel{*}{\vdash}_{\bar{M}} a'(q[\phi(u)]) \stackrel{\bar{r}_g}{\vdash} q'[\phi(U)]$$

Alors la propriété est vraie pour tout mouvement.

C'est le même raisonnement que dans le 1er cas en remplaçant α par

$\underbrace{a \dots a}_p a'$ dans le mouvement de \bar{M}_g et la règle r_g et par a' dans le mouvement de \bar{M} et la règle \bar{r}_g .

<=.

$$\text{Soit } V \stackrel{*}{\vdash}_{\bar{M}} q'[V]$$

Pour un mouvement de longueur 1 on a

$$V \in \Sigma - \{a\}$$

e. la règle utilisée étant déduite d'une règle de \bar{M}_g on a nécessairement

$$\exists U \quad V = \phi(U)$$

$$\text{avec } V \stackrel{\bar{r}_g}{\vdash} q'[V]$$

(N.B. la règle de M_g associée peut avoir les 2 formes

$$\bar{r} \quad \alpha(q[x]) \xrightarrow{\phi(t)} q'[\alpha(x)]$$

ou $\bar{r}_g \quad a'(q[x]) \xrightarrow{\phi(t)} q'[a'(x)]$

respectivement déduites de

$$r \quad \alpha(q[x]) \xrightarrow{t} q'[\alpha(x)]$$

$$r_g \quad a \dots a a'(q[x]) \xrightarrow{t} q'[a \dots a a'(x)]$$

dans le 1er cas on aura $U = V = \alpha$

dans le 2e cas on aura $V = a'$ et $U = a \dots a V$.)

Considérons alors la propriété vraie pour un mouvement de longueur ℓ

et soit

$$V = b.v \quad \left| \frac{*}{\bar{M}} \right. b(q''[v]) \quad \left| \frac{r\bar{M}}{\bar{M}} \right. q'[b.v]$$

Le mouvement $V \left| \frac{*}{\bar{M}} \right. b(q''[v])$ étant de longueur $\leq \ell$

on a par hypothèse de récurrence

$$\exists u : v = \phi(u)$$

$$\text{et } u \left| \frac{*}{\bar{M}_g} \right. q''[u]$$

Considérons alors b et $r_{\bar{M}}$:

1er cas $b = \alpha$, $r_{\bar{M}} = \bar{r} : \alpha(q''[x]) \xrightarrow{(t)} q'[\alpha(x)]$

le mouvement a lieu ssi

$$\exists w : v = \phi(t).w$$

or on a $v = \phi(u)$

$$\text{d'où } \phi(u) = \phi(t).w$$

ϕ étant complet $\exists w' : \phi(w') = w$

et donc $u = t.w'$

On peut alors appliquer r dans M_g soit

$$\alpha(q''[u]) \left| \frac{r}{\bar{M}} \right. q'(\alpha(u))$$

et $\exists U = \alpha.u$ tel que

$$U \stackrel{*}{\underset{g}{M}} \alpha(q''[u]) \longrightarrow q'(U)$$

$$\underline{2e \text{ cas}} : b = a', r_{\bar{M}} = \bar{r}_g : a'(q''[x]) \xrightarrow{\phi(t)} q'[a'(x)]$$

$$\text{associée à } r_g : \underbrace{a \dots a}_P a'(q''[x]) \xrightarrow{t} q'[\underbrace{a \dots a}_P q a'(x)]$$

le mouvement a lieu ssi

$$\exists w : v = \phi(t).w$$

$$\text{et } \phi(u) = \phi(t).w$$

$$\phi \text{ étant complet } \exists w' : \phi(w') = w$$

on peut donc appliquer r dans M_g :

$$\exists U = \underbrace{a \dots a}_P a'.u \text{ tel que}$$

$$U \stackrel{*}{\underset{g}{M}} \underbrace{a \dots a}_P a'(q''[u]) \stackrel{*}{\underset{g}{M}} q'[\underbrace{a \dots a}_P a'(u)] = q'[U].$$

La propriété est donc vraie pour tout mouvement.

N.B. Une remarque s'impose concernant la règle \bar{r}_g : elle peut être associée à une infinité de règles r_g où le nombre n de a figurant en membre gauche est $> p$. C'est à dire il y a une infinité d'arbres U ayant même image V lorsque la règle utilisée dans \bar{M} derive d'une telle règle généralisée de M_g .

Proposition II.35

La classe des forêts RATEG n'est pas close par morphisme inverse (même si on se restreint aux démarquages propres inverses).

Idée de la preuve

Il suffit de considérer le contre-exemple suivant.

Soit la forêt RATEG F construite sur

$$\Sigma = \{\neq(1, 2), a(1), \bar{a}\}$$

des arbres de la forme $\neq(t, t)$ où $t = \left. \begin{array}{c} a \\ \vdots \\ \bar{a} \\ a \end{array} \right\} p = a^p \bar{a}$

et soit ϕ défini par

$$\phi : \Delta = \{\neq(1, 2), \alpha(1), \beta(1), \bar{a}\} \rightarrow \Sigma$$

tel que

ϕ est l'identité sur \neq et \bar{a}

$$\begin{array}{ccc} \phi(\alpha) = \phi(\beta) = a & & \\ \left| \quad \quad \quad \right| & & \\ 1 \quad \quad \quad 1 & & 1 \end{array}$$

Alors $\phi^{-1}(F)$ contient des arbres de la forme $\neq(u, v)$ où u et v sont indifféremment constitués de "a" ou "β" mais ont une même longueur.

Or si les règles de RATEG permettent de tester l'égalité de deux sous-arbres, il leur est impossible de vérifier l'égalité du nombre de noeuds de deux sous-arbres. Et manifestement $\phi^{-1}(F)$ n'est pas une forêt RATEG.

Propriétés de décidabilité

Proposition II.36

Le problème du vide est indécidable pour la classe des forêts RATEG.

Preuve

C'est une conséquence du corollaire I.5 du chapitre I. En effet supposons le problème décidable. Comme la classe des RATEG est close par morphisme complet et intersection on a

$\forall K_1, K_2 \in \text{Rec}, \forall \phi_1$ et ϕ_2 complets

$$\phi_1(K_1) \cap \phi_2(K_2) \subset \text{RATEG}$$

On aurait alors la décidabilité du problème

$$\phi_1(K_1) \cap \phi_2(K_2) \stackrel{?}{=} \emptyset$$

ce qui est absurde en vertu du corollaire I.5.

Proposition II.37

Toute forêt RATEG est récursive.

Evident puisqu'il existe des automates qui les reconnaissent.

Proposition II.38

Toute forêt récursivement énumérable est l'image par un démarquage strict d'une forêt RATEG.

Conséquence immédiate du théorème I.1 et de l'inclusion de la clôture des reconnaissables par morphisme complet et intersection dans les RATEG.

Corollaire II.39

La classe des forêts RATEG n'est pas close par morphisme non complet.

Immédiat. Insistons encore sur le fait que la perte d'informations par morphisme non complet fait passer dans la plus large classe des récursivement énumérables.

CHAPITRE III

LE CAS DES RATEG HOMOGENES

Nous avons vu dans la deuxième partie que

$$\text{RATEG} \supset \text{cl}(\eta, \nu, \phi_c, \text{Rec})$$

Nous conjecturons la réciproque à savoir

$$\text{RATEG} \subset \text{c}ll(\eta, \nu, \phi_c, \text{Rec})$$

ce qui entraînerait l'égalité.

Nous montrons cette inclusion dans le cas particulier des RATEG homogènes et plus particulièrement que toute forêt RATEG homogène F peut s'écrire

$$F = \pi[\phi(K) \cap \phi(K')]$$

où π est un démarquage propre, ϕ un morphisme complet strict et K et K' deux reconnaissables.

Intuitivement on appellera forêt RATEG h -homogène une forêt reconnue par un RATEG h -homogène dont les règles seront de la forme

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$

où t sera un arbre h -homogène c'est à dire un arbre dont chaque branche joignant le sommet à une variable sera de longueur h et chaque branche joignant le sommet à un terminal de longueur $\leq h$.

Tout naturellement nous introduirons la notion de RATEG k - h -homogène généralisé dont les règles seront de la forme

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$

avec a k -homogène et t h -homogène.

La proposition III.1 montrera alors l'équivalence de tout RATEG h -homogène avec un RATEG généralisé k - h -homogène.

L'idée de la preuve du théorème III.3 est alors de simuler l'application des règles de RATEG par l'appartenance tantôt à $\phi(K)$ tantôt à $\phi(K')$ selon la profondeur à laquelle se trouvera le noeud.

Dans l'exemple III.2 on verra ainsi que les règles appliquées en des noeuds de profondeur paire sont simulées par l'appartenance à $\phi(K')$, celles appliquées en des noeuds de profondeur impaire à $\phi(K)$.

Le rôle de K et K' est de synchroniser ces simulations sur tout arbre t . Dans le cas homogène cette synchronisation est aisée ; elle consiste à découper t en des "tranches" de profondeur $2h$ qui se chevaucheront sur une profondeur h de sorte que les membres gauches a des règles appliquées en des noeuds sommets de tranches d'ordre impair coïncident avec les arbres t mémorisant les tests d'égalité des règles appliquées en des noeuds sommets de tranches d'ordre pair et vice-versa.

Nous conjecturons qu'une telle synchronisation est possible dans le cas général par deux ou plusieurs forêts reconnaissables.

Mais la difficulté dans le cas général est de synchroniser les simulations des règles. Les règles - qu'elles soient simples ou généralisées - n'étant plus homogènes, on ne peut plus faire chevaucher les arbres figurant en membre gauche et ceux mémorisant les tests d'égalité car leur profondeur est quelconque. D'autre part si on veut remplacer une séquence de mouvements simples par une règle généralisée selon le procédé du lemme III.2 on constate aisément que l'unification donne un arbre unifié $t \vee t'$ de profondeur plus grande que t et t' au lieu de se stabiliser à une même profondeur unique.

Considérons l'exemple suivant des règles non-homogènes

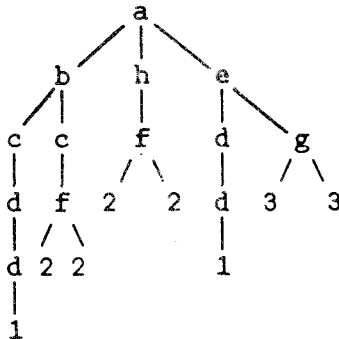
$$\begin{array}{c}
 \langle b, h, e \rangle = t \\
 \begin{array}{c}
 / \quad \backslash \quad | \quad / \quad \backslash \\
 c \quad c \quad 2 \quad 1 \quad 3 \\
 | \quad | \\
 1 \quad 2
 \end{array} \\
 a((q_1, q_2, q_3) [1, 2, 3]) \xrightarrow{\quad} q_4[a(1, 2, 3)]
 \end{array}$$

$$\begin{array}{c}
 \langle f \rangle = t_1 \\
 \begin{array}{c}
 / \quad \backslash \\
 1 \quad 1
 \end{array} \\
 h(q_8[1]) \xrightarrow{\quad} q_2[h(1)]
 \end{array}$$

$$b((q_5, q_5) [1, 2]) \xrightarrow{\langle 1, 2 \rangle = t_2} q_1[b(1, 2)]$$

$$\begin{array}{c}
 \langle d, g \rangle = t_3 \\
 \begin{array}{c}
 | \quad / \quad \backslash \\
 d \quad 2 \quad 2 \\
 | \\
 1
 \end{array} \\
 e((q_2, q_7) [1, 2]) \xrightarrow{\quad} q_3[e(1, 2)]
 \end{array}$$

Le lecteur vérifiera aisément que l'unification de t et $a(t_1, t_2, t_3)$ donnera l'arbre suivant non-homogène.



et donc on ne pourra jamais substituer une règle généralisée k - h -homogène à toute séquence utilisant l'enchaînement de ces règles.

1) Définitions préliminaires

- * Un arbre t est dit h -homogène (ou de profondeur homogène h) ssi
 - toutes les branches joignant son sommet à une variable sont de longueur h
 - toutes les autres branches joignant son sommet à une feuille de Σ_0 sont de longueur $\leq h$.
- * Une règle r d'un RATEG est dite h -homogène ssi les arbres composant t dans

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$
 sont tous h -homogènes.
- * On appellera règle généralisée k - h -homogène une règle généralisée de la forme

$$a(q[x]) \xrightarrow{t} q'[a(x)]$$
 où - $a \in T(\Sigma)_p^1$ est k -homogène
 t a toutes ses composantes h -homogènes.
- * Un RATEG généralisé est dit k - h -homogène ssi toutes ses règles sont k - h -homogènes.

* Une forêt F est RATEG h -homogène s'il existe un RATEG N h -homogène tel que $F = F(M)$.

* Un morphisme est h -homogène ssi l'image de toute lettre de Σ_n est un arbre t h -homogène de $T(\Delta)_n^1$.

Exemple 0.9

$$- \phi \left(\begin{array}{c} a \\ / \quad \backslash \\ 1 \quad 2 \quad 3 \end{array} \right) = \begin{array}{c} \alpha \\ / \quad \backslash \\ \beta \quad \gamma \quad \delta \\ / \quad \backslash \quad | \\ 1 \quad 2 \quad 2 \end{array}$$

- $\forall b \in \Sigma - \{a\}$

$$\phi \left(\begin{array}{c} b \\ / \quad \backslash \\ 1 \quad n \end{array} \right) = \epsilon$$

ϕ est non linéaire, non complet, strict, 2-homogène.

2) Propriétés

Proposition III.1

Quel que soit R RATEG h -homogène et quel que soit $k \in \mathbb{N}$, il existe un RATEG généralisé R' k - h -homogène équivalent à R .

La preuve s'appuie sur le lemme suivant :

Lemme III.2

Les séquences de mouvements simples de R sur un arbre a k -homogène sont en correspondance avec les mouvements simples de R' par l'équivalence suivante :

$\forall a$ k -homogène

$$v.a.q[u] \stackrel{*}{\vdash}_R v.q'[a(u)] \Leftrightarrow v.a.q[u] \vdash_{R'} v.q'[a(u)]$$

Preuve du lemme III.2

Par récurrence sur k , on associe à toute séquence de R sur a k -homogène un arbre T h -homogène et on en déduit une règle généralisée r' k - h -homogène

$$- k = 1$$

$$\text{soit } a(q[u]) \stackrel{r}{\vdash} q'[a(u)]$$

$$\text{avec } r : a(q[x]) \xrightarrow{t} q'[a(x)]$$

On prendra $T = t$ et $r' = r$.

. Supposons le lemme vérifié jusqu'au rang k et soit dans R le mouvement composé suivant

$$v.b.a(q[u]) \stackrel{*}{\vdash}_s v.b.(q'[a(u)]) \stackrel{r}{\vdash} v.q''[b.a(u)]$$

où s est une séquence de R sur a k -homogène

$$- b \in \Sigma_n$$

$$- a = (a_1 \dots a_n) \text{ avec } a_i \in T(\Sigma)_{n_i}^1$$

$$- u = (u_1 \dots u_n) \text{ avec } u_i \in T(\Sigma)_{p_i}^{n_i}$$

L'hypothèse de récurrence s'applique pour tout $i \in [n]$ sur les a_i k -homogènes :

On sait construire T_i

$$\text{et } r'_i : a_i(q_i[x]) \xrightarrow{T_i} q'_i[a_i(x)]$$

de sorte que

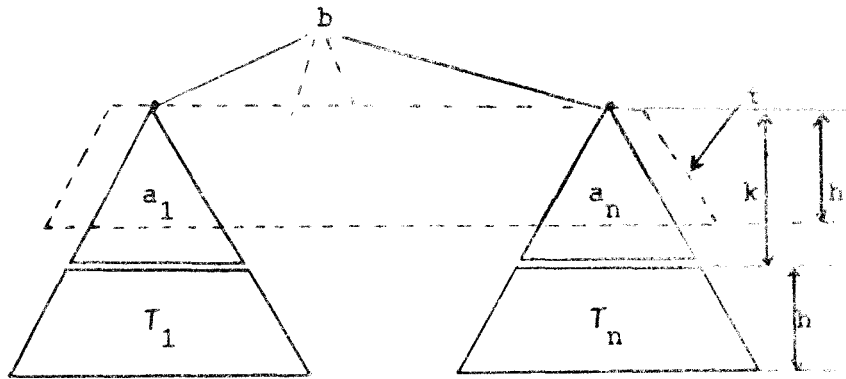
$$a_i(q_i u_i) \stackrel{*}{\vdash}_R q'_i[a_i(u_i)] \Leftrightarrow a_i(q_i u_i) \stackrel{r'_i}{\vdash} q'_i[a_i(u_i)]$$

Considérons alors la règle r :

$$b(q'[x]) \xrightarrow{t} q''[b(x)]$$

$$\text{où } q' = (q'_1 \dots q'_n)$$

L'enchaînement des mouvements dans R implique l'égalité des arbres initiaux de profondeur homogène $\inf(k, h)$ dans t et $T = (a_1.T_1, \dots, a_n.T_n)$ soit schématiquement (si $h < k$).



Nous procédons alors à l'unification de \$t\$ et \$T\$ de profondeur homogène respective \$h\$ et \$k+h\$. L'algorithme d'unification va alors substituer aux variables de \$t\$ des sous arbres de profondeur homogène \$k\$, il en résultera par conséquent un arbre unifié de profondeur homogène \$k+h\$ que l'on écrit \$a.T'\$ où \$a\$ est \$k\$-homogène et \$T'\$ \$h\$-homogène.

On en déduit alors une règle généralisée \$r'\$ \$(k+1)\$-\$h\$-homogène :

$$b.a.(q[x]) \xrightarrow{T'} q''[b.a(x)]$$

où \$q = (q_1, \dots, q_n)\$

et l'on a

$$v.b.a(q[u]) \vdash_{s,r}^* v.q''[b.a(u)] \Leftrightarrow v.b.a(q[u]) \vdash_{r'} q''[b.a(u)]$$

. la propriété est donc vraie pour tout \$k\$.

La preuve de la proposition III.1 est alors immédiate :

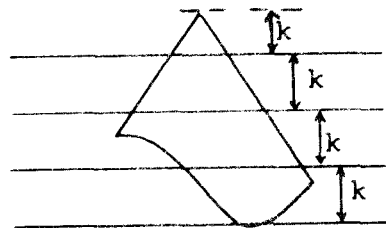
Soit \$t \in F(R)\$. On peut toujours décomposer \$t\$ en sous-arbres \$k\$-homogènes par "tranches" de profondeur \$k\$

depuis le sommet de \$t\$ et par

conséquent \$t \in F(R) \Leftrightarrow\$ il existe

un mouvement composé de reconnaissance

de \$R\$ se décomposant en séquences de mouvements simples de \$R\$ sur les sous-arbres \$k\$-homogènes ainsi découpés dans \$t\$.



Le lemme III.2 permet alors d'affirmer $t \in F(R) \Leftrightarrow$ il existe un mouvement composé de R' résultant des mouvements simples associés à chacune de ces séquences de R et par conséquent $t \in F(R) \Leftrightarrow t \in F(R')$.

On peut déduire de cette proposition un corollaire immédiat.

Corollaire III.3

A tout RATEG h-homogène on sait associer un RATEG généralisé h-h-homogène équivalent.

Exemple III.1

Construction d'une règle généralisée 3-1-homogène à partir de règles 1-homogènes.

Considérons la séquence suivante

$$v.\alpha(q[u]) \xrightarrow[r_1 r_2 r_3 r_4 r_5 r_6]{*} v q_{10}[\alpha(u)]$$

où $\alpha =$  $\in T(\Sigma)_7^1$

$$- q = (q_1, q_2, q_0, q_9, q_1, q_2, q_4) \in Q^7$$

utilisant les règles

$$r_1 : a((q_1, q_2) [1, 2]) \xrightarrow[\begin{matrix} (h, f) \\ | \quad / \backslash \\ 1 \quad 1 \quad 2 \end{matrix}]{*} q_3[a(1, 2)]$$

$$r_2 : c(q_4[1]) \xrightarrow[\begin{matrix} f \\ / \backslash \\ 1 \quad 2 \end{matrix}]{*} q_5[c(1)]$$

$$r_3 : b((q_3, q_5) [1, 2]) \xrightarrow[\begin{matrix} (a, c) \\ / \backslash \quad | \\ 1 \quad 2 \quad 2 \end{matrix}]{*} q_6[b(1, 2)]$$

$$r_4 : e((q_0, q_9) [1, 2]) \xrightarrow{\begin{array}{c} (g, h) \\ / \quad \backslash \\ 1 \quad 2 \quad 3 \end{array}} q_7[e(1, 2)]$$

$$r_5 : d((q_3, q_7) [1, 2]) \xrightarrow{\begin{array}{c} (a, e) \\ / \quad \backslash \quad / \quad \backslash \\ 1 \quad 2 \quad 3 \quad 1 \end{array}} q_8[d(1, 2)]$$

$$r_6 : \#((q_8, q_6) [1, 2]) \xrightarrow{\begin{array}{c} (d, b) \\ / \quad \backslash \quad / \quad \backslash \\ 1 \quad 2 \quad 1 \quad 3 \end{array}} q_{10}[\#(1, 2)]$$

Nous allons construire r' 3-1-homogène.

k=1

Pour r_1 r_2 et r_4 appliquées à a , c et e

on a

$$T_1 = (h, f) \quad T_2 = f \quad T_3 = (g, h)$$

$$\begin{array}{c} | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array} \quad \begin{array}{c} / \quad \backslash \\ 1 \quad 2 \end{array} \quad \begin{array}{c} / \quad \backslash \quad | \\ 1 \quad 2 \quad 3 \end{array}$$

k=2

Considérons r_3 et r_5 . On va unifier

$$(a, c) \text{ et } (a.T_1, c.T_2)$$

$$\begin{array}{c} / \quad \backslash \quad | \\ 1 \quad 2 \quad 2 \end{array}$$

ce qui donne

$$T'_1 = (a, c)$$

$$\begin{array}{c} / \quad \backslash \quad | \\ h \quad f \quad f \\ | \quad / \quad \backslash \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \quad 1 \quad 2 \end{array}$$

$$\text{puis } (a, e) \text{ avec } (a.T_1, e.T_3)$$

$$\begin{array}{c} / \quad \backslash \quad / \quad \backslash \\ 1 \quad 2 \quad 3 \quad 1 \end{array}$$

ce qui donne

$$T'_2 = (a, e)$$

$$\begin{array}{c} / \quad \backslash \quad / \quad \backslash \\ h \quad f \quad g \quad h \\ | \quad / \quad \backslash \quad / \quad \backslash \quad | \\ 1 \quad 1 \quad 2 \quad 3 \quad 4 \quad 1 \end{array}$$

k=3

Pour r_6 on unifie

$$\left(\begin{array}{cc} d & b \\ / \quad \backslash & / \quad \backslash \\ 1 & 2 & 1 & 3 \end{array} \right) \text{ et } (d.T'_1, b.T'_2)$$

ce qui donne

$$T'' = \left(\begin{array}{c} d \\ / \quad \backslash \\ a \quad e \\ / \quad \backslash \quad / \quad \backslash \\ h \quad f \quad g \quad h \\ | \quad | \quad / \quad \backslash \quad | \\ 1 \quad 1 \quad 2 \quad 3 \quad 4 \quad 1 \end{array} , \begin{array}{c} b \\ / \quad \backslash \\ a \quad c \\ / \quad \backslash \quad | \\ h \quad f \quad f \\ | \quad | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \quad 1 \quad 2 \end{array} \right)$$

d'où l'arbre

$$T = \left(\begin{array}{cccccc} h & , & f & , & g & , & h & , & h & , & f & , & f \\ | & & / \quad \backslash & & / \quad \backslash & & | & & | & & / \quad \backslash & & / \quad \backslash \\ 1 & & 1 \quad 2 & & 3 \quad 4 & & 1 & & 1 & & 1 \quad 2 & & 1 \quad 2 \end{array} \right)$$

et la règle

$$r' : \alpha(q[x]) \xrightarrow{T} q_{10}[\alpha(x)]$$

et on vérifie aisément l'équivalence

$$v.\alpha(q[u]) \xrightarrow[r_1 r_2 r_3 r_4 r_5 r_6]{*} v.q_{10}[\alpha(u)] \Leftrightarrow v.\alpha(q[u]) \xrightarrow{r'} v.q_{10}[\alpha(u)]$$

Théorème III.4

Quelle que soit F forêt Rateg h-homogène

il existe K, K' forêts reconnaissables

ϕ complet strict

π démarquage propre

tels que $F = \pi[\phi(K) \cap \phi(K')]$

Preuve du théorème III.4

Avec le corollaire III.3 on sait construire $R = (\Sigma, Q, F, R)$ Rateg généralisé h-h-homogène reconnaissant F .

Définissons alors de nouveaux symboles à partir de Σ et Q puis ϕ, π et

K et K' :

Alphabets $\bar{\Sigma}$ et Δ

- $\forall a \in \Sigma$, tel qu'il existe r :

$$r : \alpha(q'[x]) \xrightarrow{t} q[\alpha(x)]$$

où α est h-homogène de sommet $a \in \Sigma_m$, et t h-homogène nous définissons un nouveau symbole a^q de même degré m que a .

Et soit $\bar{\Sigma}$ l'ensemble des a^q ainsi définis.

Nous noterons alors $\bar{\alpha}$ l'arbre h-homogène obtenu à partir du membre gauche α de r en remplaçant son sommet a par a^q .

- A toute règle généralisée h-h-homogène r de R :

$$r : \alpha(q'[x]) \xrightarrow{t} q[\alpha(x)]$$

où $t \in T(\Sigma)_n^p$ nous associons une nouvelle lettre \bar{r} de degré n égal au degré inférieur de t .

Nous notons $\Delta = \{\bar{r} \mid r \in R\}$

Homomorphisme ϕ

ϕ est défini sur $\Sigma \cup \bar{\Sigma} \cup \Delta$ à image dans $\bar{\Sigma} \cup \Sigma$.

- $\forall \bar{r}$ de degré n associé à r

$$r : \alpha(q[x]) \xrightarrow{t} q_o[\alpha(x)]$$

$$\text{où } \alpha = a(\alpha_1 \otimes \alpha_2 \dots \otimes \alpha_m) \in T(\Sigma)_p^1$$

$$t = (t_1, \dots, t_p) \in T(\Sigma)_n^p$$

$$q = (q_1, \dots, q_p)$$

nous définissons

$$\phi \left(\begin{array}{c} \bar{r} \\ / \quad \backslash \\ x_1 \quad x_n \end{array} \right) = \bar{\alpha} \cdot \bar{t}$$

$$= a^{q_o}(\alpha_1 \otimes \alpha_2 \dots \otimes \alpha_m)(\bar{t}_1, \dots, \bar{t}_p)$$

où $\forall i \in [p]$ le sommet α_i de t_i est remplacé par a^{q_i} dans \bar{t}_i

\bar{t} ayant la même torsion que t .

- $\forall \bar{r}$ de degré 0 associé à r

$r : \alpha \xrightarrow{t} q[\alpha]$ où $\alpha \in T(\Sigma)_p^1$ et $t \in T(\Sigma)_o^p$

$$\phi(\bar{r}) = \bar{\alpha} \cdot \bar{t}$$

(Rappelons que si $\alpha \in T(\Sigma)_o^1$ t est réduit à la torsion vide)

- ϕ est l'identité sur $\Sigma \cup \bar{\Sigma}$.

Par définition ϕ est un homomorphisme complet strict.

Démarquage π

π est défini sur $\Sigma \cup \bar{\Sigma}$ à image dans Σ :

$$- \forall a^q \in \bar{\Sigma} \quad \pi \left(\begin{array}{c} a^q \\ / \quad \backslash \\ x_1 \quad x_n \end{array} \right) = \begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_n \end{array}$$

- π est l'identité sur Σ

Par définition π est complet strict et ne modifie que les labels :

c'est un démarquage propre.

On vérifie facilement que $\pi(\bar{\alpha}) = \alpha$.

Forêts K et K'

Les arbres de K et K' sont construits sur l'alphabet Δ , les arbres de K' étant de profondeur $\geq h$ avec un sous arbre initial h -homogène dont le sommet $\in \bar{\Sigma}$ et les autres labels à Σ .

K est manifestement reconnaissable. On vérifie aisément que K' l'est aussi avec les règles

$$- \forall \bar{r} \in \Delta_m \quad \begin{array}{c} \bar{r} \\ / \quad \backslash \\ q_o x_1 \quad q_o x_m \end{array} \rightarrow q_o \begin{array}{c} \bar{r} \\ / \quad \backslash \\ x_1 \quad x_m \end{array}$$

$$- \forall a \in \Sigma_m \quad \begin{array}{c} a \\ / \quad \backslash \\ q_o x_1 \quad q_o x_m \end{array} \rightarrow q_1 \begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_m \end{array}$$

$$- \forall i \in [h-2] \quad \begin{array}{c} a \\ / \quad \backslash \\ q_i x_1 \quad q_i x_m \end{array} \rightarrow q_{i+1} \begin{array}{c} a \\ / \quad \backslash \\ x_1 \quad x_m \end{array}$$

$$- \forall a^q \in \bar{\Sigma}_m \quad \begin{array}{c} a \\ / \quad \backslash \\ q_{h-1}x_1 \quad q_{h-1}x_m \end{array} \rightarrow q_h \begin{array}{c} a^q \\ / \quad \backslash \\ x_1 \quad x_m \end{array} \text{ avec } q_h \text{ état final.}$$

Schématiquement les arbres de K et K' auront les formes respectives



Et on constate aisément que les images par ϕ de K et K' seront des arbres dont la longueur de chaque branche sera multiple de h.

La démonstration du théorème précédent résulte alors immédiatement du lemme suivant :

Lemme III.5

$\forall T \in F$ reconnu par R

$\exists t \in K, \exists t' \in K' :$

$$T = \pi[\phi(t) \cap \phi(t')]$$

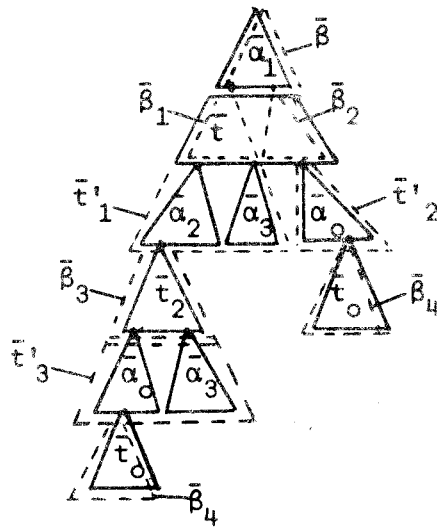
Preuve du lemme III.5

Soit $T \in F$, reconnu par la séquence

$$T \xrightarrow{*} q[T] \text{ où } q \in F$$

utilisant des règles généralisées h-h-homogènes.

Intuitivement nous allons découper T en tranches de profondeur h de façon à simuler par ϕ les règles utilisées à partir d'un t de K pour les rangs impairs et d'un t' de K' pour les rangs pairs ce qu'on traduira par un schéma de la forme suivante pour $\phi(t) \cap \phi(t')$:



Nous notons

- $r_{n,k}$ la règle h-h-homogène de la forme

$$\alpha_{n,k}(q'_{n,k}[x]) \xrightarrow{t_{n,k}} q'_{n,k}[\alpha_{n,k}(x)]$$

appliquée au $k^{\text{ème}}$ sous arbre à partir de la gauche et à la profondeur $1+nh$ dans T ,

et

- $a_{n,k}$ le sommet de $\alpha_{n,k}$.

L'enchaînement des mouvements impose alors que pour toute j^{e} composante

$t_{n,k}^j$ de $t_{n,k}$, on y applique en son sommet $a_{n+1,l}$ la règle $r_{n+1,l}$ telle que

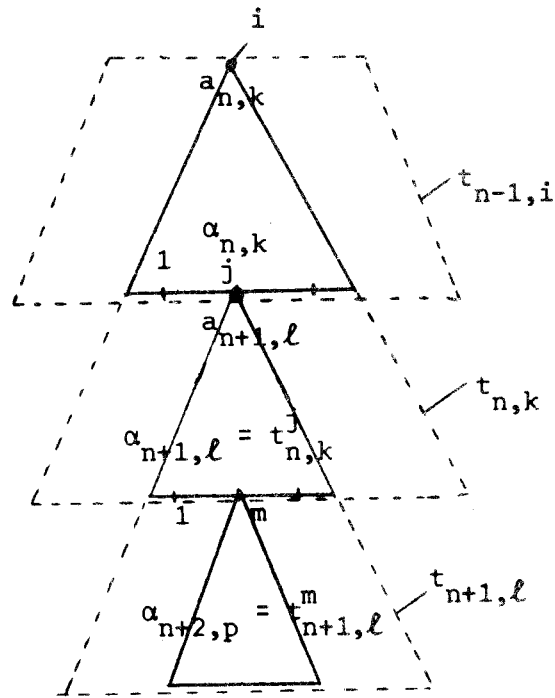
- $q_{n+1,l} = q'_{n,k}^j$, $j^{\text{ème}}$ composante de $q'_{n,k}$

et

- $\alpha_{n+1,l} = t_{n,k}^j$

(Rappelons que s'il s'agit d'une règle terminale on a

$$\alpha_{n+1,l} = t_{n,k}^j \in T(\Sigma)_0^1.)$$



On peut alors définir t de K et t' de K' de façon unique par :

$$\begin{array}{c}
 t = \bar{r}_1 \\
 \begin{array}{cc}
 / & \backslash \\
 \bar{r}_{3,1} & \bar{r}_{3,k_3} \\
 / & \backslash \\
 / & \backslash \\
 \bar{r}_{(2n+1),1} & \bar{r}_{(2n'+1),k_{2n'+1}} \\
 / & \backslash \\
 \bar{r}_{(2p+1),1} & \bar{r}_{(2p+1),k_{2p+1}}
 \end{array}
 \end{array}$$

en prenant les $r_{n,k}$ de rang impair

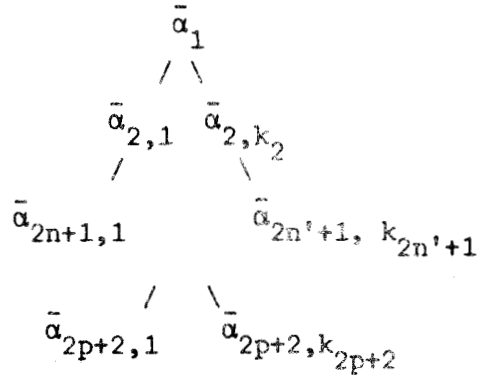
et

$$\begin{array}{c}
 t' = \bar{\alpha}_1 \\
 \begin{array}{cc}
 / & \backslash \\
 \bar{r}_{2,1} & \bar{r}_{2,k_2} \\
 / & \backslash \\
 / & \backslash \\
 \bar{r}_{2n,1} & \bar{r}_{2n',k_{2n'}} \\
 / & \backslash \\
 \bar{r}_{2p+2,1} & \bar{r}_{2p+2,k_{2p+2}}
 \end{array}
 \end{array}$$

en prenant les $r_{n,k}$ de rang pair

(N.B. On note k_n le nombre de sommets de T à la profondeur $nh+1$).

On vérifie alors immédiatement que $\phi(t) \cap \phi(t')$ est de la forme



et en appliquant π on retrouve T puisque

$$\forall i, k \quad \pi(\bar{\alpha}_{i,k}) = \alpha_{i,k}$$

Réciproquement

Soit t et $t' \in K$ et K' et soit

$$T = \pi[\phi(t) \cap \phi(t')]$$

Si l'intersection est non vide elle est nécessairement un arbre de la forme ci-dessus.

Les feuilles de l'arbre sont des $\bar{\alpha}_{n,k} \in T(\Sigma)_o^1$ correspondant à l'application des règles $\bar{r}_{n,k}$ de la forme

$$\alpha_{n,k} \longrightarrow q_{n,k}[\alpha_{n,k}]$$

Considérons alors les $\bar{\alpha}_{n,k}$ $\bar{\alpha}_{n,k+1}$ $\bar{\alpha}_{n,h}$ sous-arbres de

$$\bar{\alpha}_{n-1,i} \text{ dans } \phi(t) \cap \phi(t').$$

Les sommets $a_{n,k}$ $a_{n,k+1}$ \dots $a_{n,h}$ étant marqués des états

$q_{n,k}$ $q_{n,k+1}$ \dots $q_{n,h}$ on pourra appliquer la règle

$$r_{n-1,i} : \alpha_{n-1,i}(q[x]) \xrightarrow{t} q_{n-1,i}[\alpha_{n-1,i}(x)]$$

avec $q = (q_{n,k}, q_{n,k+1}, \dots, q_{n,h})$

et $t = (\alpha_{n,k}, \alpha_{n,k+1}, \dots, \alpha_{n,h})$

et on a dans T

$$\alpha_{n-1,i}(\alpha_{n,k}, \alpha_{n,k+1}, \dots, \alpha_{n,h}) \frac{*}{r_{n,k}, r_{n,k+1}, \dots, r_{n,h}} \alpha_{n-1,i}(q[\alpha_{n,k}, \dots, \alpha_{n,h}])$$

$$\frac{}{r_{n-1,i}} q_{n-1,i}[\alpha_{n-1,i}(\alpha_{n,k}, \dots, \alpha_{n,h})]$$

Supposons alors pour que pour tout sous arbre $u_{n,j}$ de T dont le sommet est au rang $nh+1$ on ait

$$u_{n,j} \frac{*}{r_{n,j}} q_{n,j}[u_{n,j}]$$

Alors considérons $\alpha_{n-1,i}$ de successeurs

$$u_{n,k}, u_{n,k+1}, \dots, u_{n,h}$$

On va appliquer la règle

$$r_{n-1,i} : \alpha_{n-1,i}(q[x]) \xrightarrow{t} q_{n-1,i}[\alpha_{n-1,i}(x)]$$

avec $q = (q_{n,k}, q_{n,k+1}, \dots, q_{n,h})$

et $t = (\alpha_{n,k}, \alpha_{n,k+1}, \dots, \alpha_{n,h})$

et par conséquent on aura dans R le mouvement

$$\alpha_{n-1,i}(u_{n,k}, \dots, u_{n,h}) \frac{*}{r_{n-1,i}} \alpha_{n-1,i}(q[u_{n,k}, \dots, u_{n,h}])$$

$$\frac{}{r_{n-1,i}} q_{n-1,i}[\alpha_{n-1,i}(u_{n,k}, \dots, u_{n,h})]$$

et l'enchaînement des règles est vrai pour tout rang $(n-1)h+1$.

On aura donc

$$T \frac{*}{R} q_{1,1}[T]$$

En imposant alors pour les arbres de K un sommet \bar{r}_i correspondant à l'affectation d'un état final q_i de F au sommet de α_i on aura alors

$$q_{1,1} = q_1 \in F$$

et par conséquent $T \in F$.

3) Exemple III.2 : Forêt Rateg h-homogène et sa caractérisation comme image par un démarquage propre de l'intersection des images homomorphes de 2 forêts reconnaissables

Nous reprenons l'exemple "classique" de la forêt F sur

$\Sigma = \{a_o, a(1), b(1, 2)\}$ des arbres de la forme

$$b(a^{n+p} a_o \otimes b(a^{n+p-1} a_o \otimes b(a^{n+p-2} a_o \otimes b(a^{p+1} a_o \otimes b(a^p a_o \otimes a^q a_o)) \dots)))$$

dont la branche principale est constituée de $(n+1)''b''$ avec $(n+1)$ impair

(p et q quelconques). Cette forêt est reconnue par le Rateg R défini par

$$R = (\Sigma, Q, F, \mathcal{R}) \text{ où}$$

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

$$F = \{q_1\}$$

et \mathcal{R} ensemble fini des règles suivantes :

$$R_o \quad a_o \rightarrow q_o[a_o]$$

$$a(q_o[1]) \xrightarrow{\langle 1 \rangle} q_o[a(1)]$$

$$b((q_o \ q_o) [1, 2]) \xrightarrow{\langle 1, 2 \rangle} q_1[b(1, 2)]$$

$$R_7 \quad b((q_o \ q_1) [1, 2]) \xrightarrow{\begin{array}{c} (a, \quad b) \\ | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array}} q_2[b(1, 2)]$$

$$R_8 \quad b((q_o \ q_2) [1, 2]) \xrightarrow{\begin{array}{c} (a, \quad b) \\ | \quad / \quad \backslash \\ 1 \quad 1 \quad 2 \end{array}} q_1[b(1, 2)]$$

De façon évidente la première règle R_o les 2 dernières R_7 et R_8 sont

1-1-homogènes. Remplaçons alors les autres par :

$$R_1 \quad a(q_o[1]) \xrightarrow{a_o} q_o[a(1)]$$

$$R_2 \quad a(q_o[1]) \xrightarrow{\begin{array}{c} (a) \\ | \\ 1 \end{array}} q_o[a(1)]$$

$$R_3 \quad b((q_o \ q_o) [1, 2]) \xrightarrow{(a_o \ a_o)} q_1[b(1, 2)]$$

$$R_4 \quad b((q_o \ q_o) [1, 2]) \xrightarrow{\begin{array}{c} (a, \ a_o) \\ | \\ 1 \end{array}} q_1[b(1, 2)]$$

$$R_5 \quad b((q_o \ q_o) [1, 2]) \xrightarrow{\begin{array}{c} (a_o, \ a) \\ | \\ 1 \end{array}} q_1[b(1, 2)]$$

$$R_6 \quad b((q_0 \ q_0) [1, 2]) \xrightarrow{\begin{array}{c} (a, a) \\ | \quad | \\ 1 \quad 2 \end{array}} q_1[b(1, 2)]$$

Alors le Rateg $R' = (\Sigma, Q, F, R')$ où R' est l'ensemble $\{R_i \mid 0 \leq i \leq 8\}$ est un Rateg 1-1-homogène reconnaissant F .

Selon le procédé décrit dans la démonstration précédente du théorème III.4 nous définissons

$$\bar{\Sigma} = \{a_0^{q_0}, a^{q_0}(1), b^{q_1}(1, 2), b^{q_2}(1, 2)\}$$

$$\text{et } \Delta = \{\bar{R}_0, \bar{R}_1, \bar{R}_3, \bar{R}_2(1), \bar{R}_4(1), \bar{R}_5(1), \bar{R}_6(1, 2), \bar{R}_7(1, 2, 3), \bar{R}_8(1, 2, 3)\}$$

D'autre part les arbres α membres gauches des R_i étant 1-homogènes, les arbres $\bar{\alpha}$ obtenus en remplaçant le sommet a par la lettre associée a^q de $\bar{\Sigma}$ seront ici réduits à a^q .

Par l'homomorphisme ϕ décrit précédemment on aura ici

$$\phi(\bar{R}_0) = a_0^{q_0}$$

$$\phi(\bar{R}_1) = a^{q_0}(a_0^{q_0})$$

$$\phi(\bar{R}_3) = b^{q_1}(a_0^{q_0}, a_0^{q_0})$$

$$\phi(\bar{R}_2) = a^{q_0}(a_0^{q_0}(1))$$

$$\begin{array}{c} | \\ 1 \end{array}$$

$$\phi(\bar{R}_4) = b^{q_1}(a_0^{q_0}(1), a_0^{q_0})$$

$$\begin{array}{c} | \\ 1 \end{array}$$

$$\phi(\bar{R}_5) = b^{q_1}(a_0^{q_0}, a_0^{q_0}(1))$$

$$\begin{array}{c} | \\ 1 \end{array}$$

$$\phi(\bar{R}_6) = b^{q_1}(a_0^{q_0}(1), a_0^{q_0}(2))$$

$$\begin{array}{c} / \quad \backslash \\ 1 \quad 2 \end{array}$$

$$\phi(\bar{R}_7) = b^{q_2}(a_0^{q_0}(1), b^{q_0}(2, 3))$$

$$\begin{array}{c} / \quad | \quad \backslash \\ 1 \quad 2 \quad 3 \end{array}$$

$$\phi(\bar{R}_8) = b^{q_1}(a_0^{q_0}(1), b^{q_2}(2, 3))$$

$$\begin{array}{c} / \quad | \quad \backslash \\ 1 \quad 2 \quad 3 \end{array}$$

ϕ étant l'identité sur $\Sigma \cup \bar{\Sigma}$

De même $\pi : \Sigma \cup \bar{\Sigma} \rightarrow \Sigma$ se réduira ici à l'identité sur Σ

et $\pi(\begin{matrix} a^q \\ / \quad \backslash \\ 1 \quad m \end{matrix}) = a$ pour tout a^q de $\bar{\Sigma}_m$.

Notons qu'en plus on aura

$$\pi(\begin{matrix} \bar{a}^q \\ / \quad \backslash \\ 1 \quad m \end{matrix}) = \pi(\begin{matrix} a^q \\ / \quad \backslash \\ 1 \quad m \end{matrix}) = a = \alpha .$$

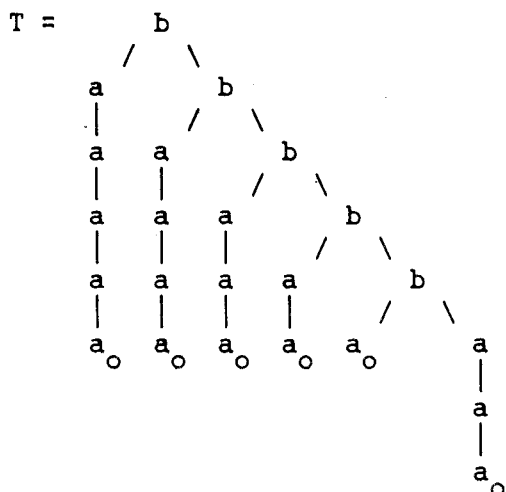
Les forêts K et K' étant construites sur Δ vérifient dans ce cas respectivement :

- les arbres de K ont un sommet correspondant à l'application d'une règle de R' menant dans l'état final q_1 , ici donc : $R_{3,4,5,6}$ ou R_8 .
- Les arbres de K' ont "un sous-arbre initial 1-homogène dont le sommet est dans $\bar{\Sigma}$ " soit ici ont pour sommet un a^q de $\bar{\Sigma}$.

Par ϕ on obtient alors des arbres dont les branches ont une longueur multiple de 1 ce qui est ici (et toujours) vérifié !...

Le cas des règles $R_{3,4,5,6}$ correspondant à des arbres contenant un seul symbole "b" ne présente aucun intérêt. Nous allons nous attacher au cas suivant où $n = 4, p = 0, q = 2$. Le lecteur vérifierait facilement que pour $p = 0, 1$ ou $p \geq 1$ et $q = 0, 1$ ou $q \geq 1$ on appliquerait R_3, R_4, R_5 ou R_6 - et on pourrait aisément généraliser la construction pour n quelconque

Considérons plus particulièrement l'arbre T suivant de F :

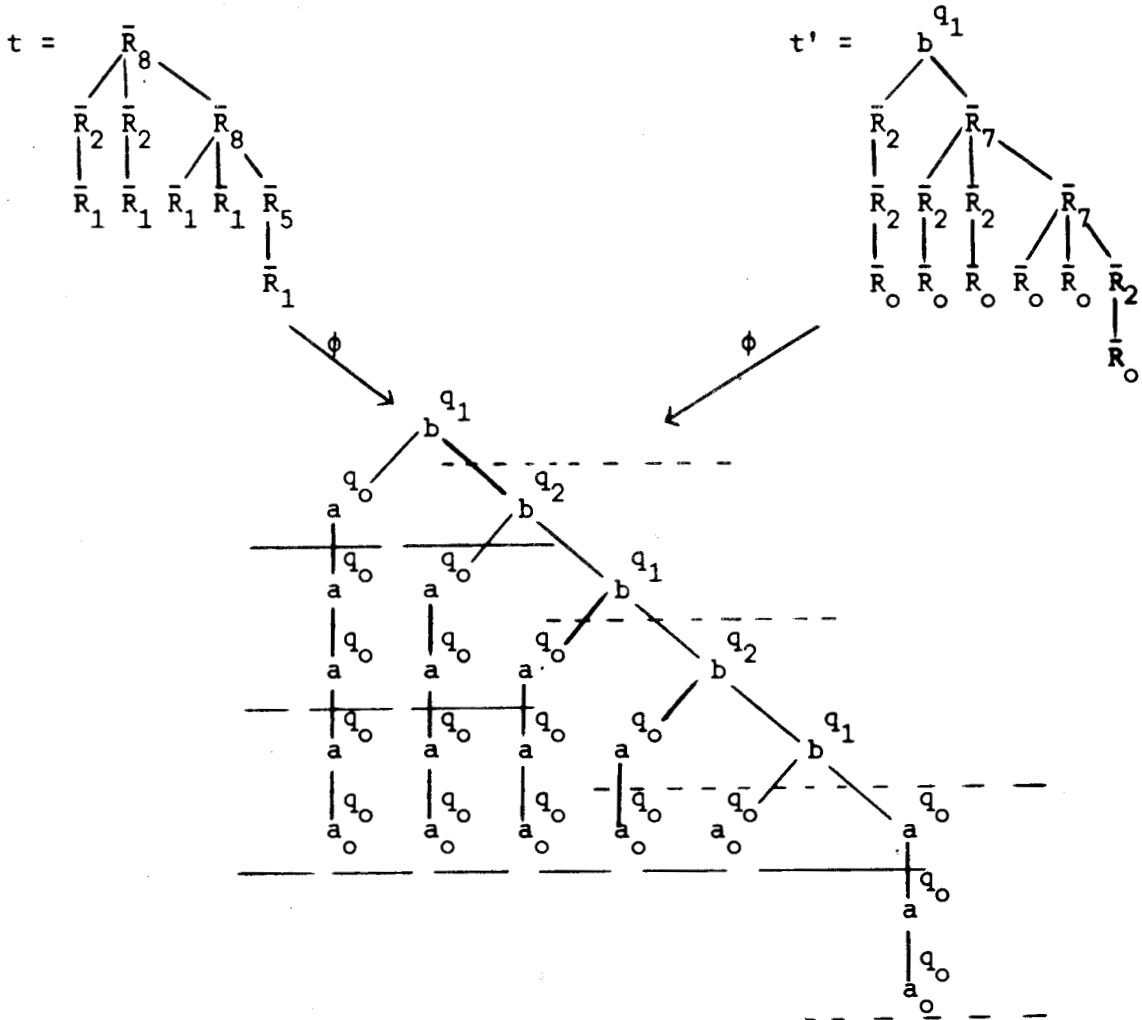


On peut alors définir de façon unique $t \in K$, $t' \in K'$ tel que

$$T = \pi[\phi(t) \cap \phi(t')]$$

On prend en effet pour $t \in K$ les sommets \bar{R}_i associés aux règles R_i appliquées en rang impair dans le mouvement de R' reconnaissant T et pour $t' \in K'$ le sommet b^{q_1} puis les sommets \bar{R}_i associés aux règles R_i appliquées en rang pair dans le même mouvement.

On vérifie alors $\phi(t) = \phi(t')$ et $T = \pi[\phi(t)]$.



4) Cas des morphismes complets homogènes

Proposition III.6

La classe des forêts RATEG homogènes est close par union, intersection et morphisme complet homogène.

Preuve

C'est une conséquence des propositions II.23, II.24 et II.27 pour lesquelles le fait d'imposer l'homogénéité des arbres t testant les égalités dans les règles ne change rien à la démonstration. Le lecteur vérifiera aisément que l'unification de deux arbres h et h' -homogènes donne - s'il existe - un unifié $\text{sup}(h, h')$ -homogène et que le lemme II.25 reste vrai. De même la construction du lemme II.30 donne un RATEG généralisé h - h -homogène et assure que l'image par morphisme complet strict homogène d'une forêt RATEG homogène est une forêt RATEG homogène.

Enfin le cas d'un morphisme ϕ complet non strict homogène ne présente pas d'intérêt (cas d'un alphabet dont les lettres sont de degré 0 ou 1 uniquement et tel que l'image par ϕ de toute forêt se réduit à l'ensemble des terminaux).

Corollaire III.7

La classe des RATEG homogènes est close par démarquage propre.

Evident puisqu'un démarquage propre est un morphisme complet strict 1-homogène.

Corollaire III.8

La classe des RATEG homogènes contient la clôture par union, intersection et morphisme complet homogène des reconnaissables.

Evident puisque Rec sont des forêts RATEG reconnues par des RATEG α -homogènes.

Nous pouvons alors récapituler les résultats obtenus par 3 inclusions :

$$\text{RATEG} \supset \text{Cl}(\text{Rec}, u, n, \phi_c)$$

$$\text{RATEG homogènes} \supset \text{Cl}(\text{Rec}, u, n, \phi_{c,h})$$

$$\text{RATEG homogènes} \supset \text{Cl}(\text{Rec}, u, n, \phi_c)$$

L'inclusion inverse

$$\text{RATEG homogènes} \subset \text{Cl}(\text{Rec}, u, n, \phi_{c,h})$$

n'est pas vraie. Le lecteur vérifiera aisément que le découpage des arbres ne peut se faire comme dans le théorème III.4 qu'à l'aide d'un morphisme ϕ non homogène.

BIBLIOGRAPHIE

- [AD1] A. ARNOLD et M. DAUCHET
"Théorie des magmaïdes I" RAIRO Informatique Théorique 12-3, 235-257
 (1978)
- [AD2] A. ARNOLD et M. DAUCHET
"Théorie des magmaïdes II" RAIRO Informatique Théorique 13-2, 135-156
 (1979).
- [AD3] A. ARNOLD et M. DAUCHET
"Bitransductions de forêts" Third International Colloquium on Automata
 Languages and Programming, Edinburg University Press (1976) 74-86.
- [AD4] A. ARNOLD et M. DAUCHET
*"Caractérisation algébrique des ensembles d'arbres récursivement
 énumérables"* 1er Colloque AFCET-SMF, Paris (1978) T2, 1-14.
- [AD5] A. ARNOLD et M. DAUCHET
"Sur l'inversion des morphismes d'arbres" 5th Automata, Languages
 and Programming (1978) 26-35
- [AD6] A. ARNOLD et M. DAUCHET
"Morphismes et bimorphismes d'arbres" à paraître dans Theoretical
 Computer Science.
- [AU1] A.V. AHO et J.D. ULLMAN
"Syntax directed translations and the pushdown assembler" J. Comput.
 System Sci, 3, (1969), 37-56.
- [AU2] A.V. AHO et J.D. ULLMANN
"Properties of syntax directed translations". J. Comput System Sci 3,
 (1969), 319 - 334.
- [B] B.S. BAKER
"Tree Transductions and families of tree-languages" 5th ACM Proc on
 Theory of Computing (1973).

- [BJ] D. BJORNER
 "The synthesis of finite-state syntax-directed top-down and bottom-up transducers". 11th. IEEE Symp. on Switching and automata theory (1970) 122-132.
- [BG] R.V. BOOK et S.A. GREIBACH
 "Quasi Realtime Languages" Mathematical Systems Theory 2 (1969).
- [BN] L. BOASSON et M. NIVAT
 "Sur diverses familles de langages fermées par transduction rationnelle" Acta Informatica 2 (1973) 180-188.
- [CH] J. CHAUCHE
 "Transducteurs et arborescences. Etudes et réalisations de systèmes appliqués aux grammaires transformationnelles". Doctorat d'Etat Grenoble (1974).
- [COUS] COUSINEAU
 "Les arbres à feuilles indicées : un cadre algébrique pour l'étude des structures de contrôle" Thèse de Doctorat d'Etat, Institut de Programmation, Université de Paris VI.
- [D] M. DAUCHET
 "Transductions de forêts. Bimorphismes de magmoïdes". Doctorat d'Etat Lille I (1977).
- [DM1] M. DAUCHET et J. MONGY
 "Images de noyaux reconnaissables par diverses classes de transformations" 4ème CLAAP (1979) 79-101.
- [DM2] M. DAUCHET et J. MONGY
 "Transformations de noyaux reconnaissables, capacité générative des bimorphismes de forêts" FCT' 79, Akademie-Verlag Berlin 99-104.
- [EW] S. EILENBERG et J.B. WRIGHT
 "Automata in general algebras" Inf and Control 11 (1967) 217-231.

- [EN1] J. ENGELFRIET
"Top down tree transducers with regular look-ahead" Math. Systems Theory 10, 289-303 (1977).
- [EN2] J. ENGELFRIET
"Bottom-up and top-down tree transformations. A comparison" Math. Systems Theory 9 (1975) 198-231.
- [GP] S. GINSBURG et B. PARTEE
"A mathematical model of transformational grammars". Inf and Control 15 (1969) 297-334.
- [GL] M. GROSS et A. LENTIN
"Notions sur les grammaires formelles" Gauthier Villars (1967)
- [JG] J.R. GUARD
"Automated logic for semi-automated mathematics" Scientific Report n° 1 AFCRL (64).
- [H] G.P. HUET
"A unification algorithm for typed λ -calculus" Theoretical Computer Science 1(1975).
- [L] M. LATTEUX
"Cônes rationnels commutatifs" J.C.S.S. 18, 3 p 307-333 (1979)
- [LI] E. LILIN
"Une généralisation des transducteurs d'états finis d'arbres : les S-transducteurs" Thèse 3e cycle Lille I (1978).
- [LO] B. LOHRO
"De la définition des langages de programmation : méthode des attributs sémantiques". Doctorat és Sciences. Université Paul Sabatier de Toulouse (1974).
- [NI] M. NIVAT
"Transductions des langages de Chomsky". Annales de l'Institut Fourier de Grenoble 18 (1968) 339-456.

- [PAQ] C. PAIR et A. QUERE
"Définition et étude des bilangages réguliers" Inf. and Control 13
 (1968) 565-593.
- [PR] P. PETERS et S. RICHIE
"On restricting the base component of transformational grammars"
 Inf & Control 18(1971)
- [R] J.A. ROBINSON
"A machine-oriented logic based on the resolution principle"
 J.A.C.M. 12(1) (1965) 23-41.
- [R1] W.C. ROUNDS
"Trees, transducers and transformations". Ph. D. Dissertation,
 Stanford University (1968).
- [R2] W.C. ROUNDS
"Context-free grammars on trees". 1st ACM Proc. on theory of
 computing (1969) 143-148.
- [SC] P.P. SCHREIBER
"Tree transducers and syntax-connected transductions" In "Les Arbres
 en algèbre et en programmation", Colloque de Lille (1976).
- [TA] M. TAKAHASHI
"Primitive transformations of regular sets and recognizable sets"
 In "Automata, languages and programming" (Nivat Ed.) North-Holland,
 Amsterdam (1973) 475-480.
- [TH1] J.W. THATCHER
*"Characterizing derivation trees of context free grammars through a
 generalization of finite automata theory"*. J. Comput. System Sci. 1
 (1967) 317-322.
- [TH2] J.W. THATCHER
"Generalized² sequential machine maps" J. Comput. System Sci 4 (1970) 339-367.
- [TH3] J.W. THATCHER
"Tree automata : an informal survey" In "Currents in the theory of
 computing" (A.V. Aho, Ed.) Prentice Hall (1973) 143-178.

