

LIFL

U.A. 369 du C.N.R.S.

cy
15

50376
1986
33

LABORATOIRE D'INFORMATIQUE FONDAMENTALE DE LILLE

50376
1986
33

THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE
FLANDRES ARTOIS

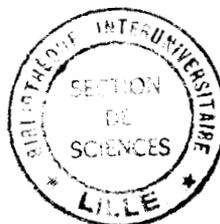
pour obtenir

Le titre de DOCTEUR
en Informatique

Rationalité et reconnaissabilité
dans des graphes

par

BOSSUT Francis



soutenue le 24 Avril 1986

Rationalité et reconnaissabilité dans des graphes

par

BOSSUT Francis

Thèse soutenue le 24 avril 1986, devant:

<u>Jury</u> :	MM.	G.	COMYN	Président
		A.	ARNOLD	Rapporteur
		D.	PERRIN	Rapporteur
		M.	NIVAT	
		M.	DAUCHET	Directeur de thèse
		G.	JACOB	
		M.	LATTEUX	
<u>Invité</u> :	MM.	J.	CELEYRETTE	
		E.	MAZET	

J'exprime à Messieurs les membres du Jury ma sincère gratitude :

à Monsieur le Professeur Maurice Nivat qui me fait l'insigne honneur de juger ce travail;

à Monsieur le Professeur André Arnold dont le rôle de rapporteur attentif et enthousiaste m'a profondément touché;

à Monsieur le Professeur Dominique Perrin dont j'apprécie hautement le soutien qu'il accorde à mon étude;

à Gérard Comyn qui n'a cessé de s'enquérir de l'avancement de mon travail et souligne l'intérêt qu'il y porte en présidant ce Jury;

à Max Dauchet instigateur et chef d'orchestre de cette thèse à qui j'exprime ma plus cordiale reconnaissance pour son exceptionnelle patience, sa grande qualité de relation et ses inspirations décisives;

à Michel Latteux qui m'a transmis le goût de l'informatique théorique et dont la précision et la clarté du discours ont été les modèles pour la rédaction de cette thèse;

à Gérard Jacob qui m'a initié aux magmoïdes et qui me témoigne de son intérêt en participant à ce Jury;

à Messieurs Celeyrette ex-Président de l'Université de Lille III et Mazet directeur de l'U.F.R de Mathématiques de Lille III desquels je dépends.

La dactylographie de cette thèse a été réalisée avec diligence et compétence par Madame Bénédicte Vandroeme que je tiens à remercier et à complimenter.

Cette thèse est l'aboutissement d'une étroite collaboration avec mon collègue et ami Bruno Warin, comme moi assistant en informatique à l'Université de Lille III.

Notre travail consista à élaborer de nouvelles définitions et à en éprouver l'intérêt et la solidité par l'obtention plus ou moins aisée de résultats techniques. C'est au cours de nombreuses et animées discussions propices à la confrontation des idées que les concepts et résultats présentés dans cette thèse ont vu le jour.

Cette double paternité rend parfaitement illusoire l'attribution à l'un plutôt qu'à l'autre telle définition de base ou tel lemme technique.

Francis Bossut.

Dites! L'erreur plombant les fronts; les bagnes
de la croyance où le savoir marchait au pas;

...

Dites! Quels temps versés au gouffre des années,
et quels cerveaux chargés de noble lassitude
a-t-il fallu pour faire si peu de certitude ?

Emile Verhaeren.

A mon Père,

Francis Bossut.

TABLE DES MATIERES

INTRODUCTION ... p. 5

CHAPITRE 0 : PRELIMINAIRES SUR LE CADRE FORMEL DE L'ETUDE : LES
MAGMOÏDES ... p. 46

- 0.1. DEFINITION DES MAGMOÏDES
- 0.2. LES MORPHISMES DE MAGMOÏDES
- 0.3. LES MAGMOÏDES LIBRES

CHAPITRE I : LE MAGMOÏDE DES DAGS PLANAIRES ... p. 52

- I.1. LE MAGMOÏDE DES DAGS PLANAIRES : $\hat{DP}(\Sigma)$
- I.2. REPRESENTATION GRAPHIQUE DES ELEMENTS DE $\hat{DP}(\Sigma)$
- I.3. EXISTENCE D'UN REPRESENTANT CANONIQUE DES ELEMENTS
DE $\hat{DP}(\Sigma)$
- I.4. FORME NORMALE DUALE D'UN ELEMENT DE $\hat{DP}(\Sigma)$
- I.5. FEUILLAGE D'UN DAG PLANAIRE
- I.6. EXTENSIONS DES OPERATIONS DU MAGMOÏDE, AUX TAILLIS

CHAPITRE II : LES AUTOMATES DE DAGS PLANAIRES ... p. 74

II.1. DEFINITIONS ET CLASSIFICATIONS

- II.1.1. Automates descendants
- II.1.2. Exemple et critiques des définitions
- II.1.3. Dualité entre automates ascendants et
automates descendants
- II.1.4. Théorème de "remontée (resp. descente) du
contrôle bas (resp. haut)"
- II.1.5. Automates d'arbres et automates de dags
planaires
- II.1.6. Automates déterministes

II.2. PROPRIETES DE CLOTURE

- II.2.1. Clôture par union
- II.2.2. Clôture par intersection
- II.2.3. Clôture par produit tensoriel
- II.2.4. Clôture par étoile du produit tensoriel
- II.2.5. Clôture par produit de composition
- II.2.6. Non clôture par étoile du produit de
composition

- II.2.7. Non clôturation par homomorphisme de magma
- II.2.8. Clôturation par homomorphisme connexe
- II.2.9. Clôturation par homomorphisme inverse
- II.2.10. Problème ouvert : clôturation de REC par complémentaire

CHAPITRE III : LES GRAMMAIRES REGULIERES DE DAGS PLANAIREs ... p. 145

III.1. DEFINITIONS ET EXEMPLES

- III.1.1. Grammaires linéaires à droite
- III.1.2. Grammaires régulières
- III.1.3. Critique des définitions

III.2. THEOREMES D'IDENTITE DES CLASSES DE TAILLIS :
TAILLIS RECONNAISSABLE = TAILLIS REGULIERS =
TAILLIS LINEAIRES A DROITE

- III.2.1. Théorème d'équivalence entre grammaires régulières et grammaires linéaires à droite
- III.2.2. Théorème d'équivalence entre régularité et reconnaissabilité dans les dags planaires

III.3. SCHEMA RECAPITULATIF

CHAPITRE IV : LES DAGS DE DERIVATION ... p. 176

IV.1. DEFINITIONS

IV.2. THEOREME : LES TAILLIS DE DAGS DE DERIVATIONS
SONT REGULIERS

- IV.2.1. Exemple introductif
- IV.2.2. Généralisation de la construction
- IV.2.3. Démonstration du théorème

IV.3. L'INDECIDABILITE DE L'EGALITE DE DEUX TAILLIS
REGULIERS

CHAPITRE V : LA CONNEXITE DANS LES DAGS PLANAIRES ... p. 199

V.1. L'AUTOMATE DE DAGS PLANAIRES DG

V.2. PROPRIETES DE L'AUTOMATE DG EN RAPPORT AVEC LA
CONNEXITE

V.3. CONCLUSION

CHAPITRE VI : LA RATIONALITE DANS LES DAGS PLANAIRES ... p. 216

VI.1. LES MAGMOÏDES TYPES

VI.2. LES MORPHISMES DE MAGMOÏDES TYPES

VI.3. LES MAGMOÏDES TYPES LIBRES

VI.4. RATIONALITE DANS LES DAGS PLANAIRES : DEFINITIONS

VI.5. JUSTIFICATION DE LA DEFINITION DE RATIONALITE
DANS LES DAGS PLANAIRES

CHAPITRE VII : LE THEOREME RAT = REG ... p. 284

VII.1. ENONCE DU THEOREME

VII.2. DEMONSTRATION DU THEOREME

VII.2.1. Enoncé du lemme

VII.2.2. Démonstration du lemme par induction

VII.2.3. Démonstration de l'inclusion de RAT
dans REC

VII.2.4. Démonstration de l'inclusion de REC
dans RAT

VII.3. CONCLUSION

INTRODUCTION

A. PRESENTATION

- A.1. Buts de la thèse
- A.2. Démarche de la thèse
- A.3. Présentation de l'introduction

B. CORPS DE L'INTRODUCTION

- B.1. Des arbres aux dags
- B.2. Des exemples sur les dags
- B.3. La rationalité et la reconnaissabilité dans les mots
et dans les arbres
- B.4. Contenu de notre travail
- B.5. Axes de développement

A. PRESENTATION

Le but de notre thèse est de :

1 - Poser de façon critique un cadre formel pour les DAGS (Directed Acyclic Graphs : graphes orientés sans circuit), cadre dans lequel nous pourrions "pratiquement" calculer sur ces objets.

2 - Définir les notions de rationalité et de reconnaissabilité dans l'esprit de ce qui existe dans les mots et dans les arbres.

3 - Prouver des propriétés et construire des algorithmes posant des jalons pour une utilisation réaliste de ces objets.

Notre démarche est sous-tendue par deux principes :

1 - L'intérêt de travailler dans les structures profondes, fussent-elles complexes, des objets.

2 - L'importance conceptuelle de la rationalité et de la reconnaissabilité dans les fondements syntaxiques de l'informatique théorique.

Dans l'introduction :

1 - Après quelques exemples illustrant l'intérêt de travailler dans les structures profondes des objets, nous situons notre étude sur les dags comme un prolongement naturel de l'étude des arbres.

2 - Nous parcourons une série de quatre exemples, destinés à introduire intuitivement la notion de dags planaires.

3 - Dans notre travail nous identifions les mots, les arbres et les dags planaires sans circuit à des objets syntaxiques d'un magma libre engendré par un alphabet (Arnold et Dauchet [1], [2]) respectivement simple (c'est-à-dire bigradué 1-1) gradué (c'est-à-dire bigradué 1-n) et bigradué quelconque. Nous rappelons rapidement les résultats de la théorie des langages (mots) et des forêts (arbres) qui trouveront un écho dans notre théorie des taillis (dags planaires).

4 - Nous donnons une présentation informelle de notre travail assortie d'une discussion critique des définitions, mise en évidence des alternatives, des difficultés rencontrées et des solutions proposées. Cette partie sert de guide à qui voudrait se plonger dans une lecture technique du corps de la thèse.

5 - Nous concluons en proposant des axes de développement à ce travail.

B. CORPS DE L'INTRODUCTION

B.1. Des arbres aux DAGS

L'idée que mieux vaut faire des opérations simples sur la structure profonde des objets (fut-elle complexe) plutôt que des opérations compliquées sur la structure apparente simple, s'illustre avec succès dans les arbres :

- en compilation :

- . Nivat [3], Aho et Ullman [10]
- . Compilation par attribut (Courcelle [4], Lohro [5] pour le système DELTA)
- . méta-compilation (Gaudel [6] pour le système PERLUETTE)

- dans les outils modernes de développement de programme (Kahn [7] pour MENTOR).

Dans ces exemples, l'objet manipulé est l'arbre structurel du programme et non le programme vu comme un mot. Il n'y a pas de miracle, le gain de simplicité vient du déploiement par l'analyse syntaxique de l'information de la structure de programme. Les arbres sont utilisés par ailleurs dans de nombreux autres domaines de l'informatique : types d'objets (Goguen [8]), arbres de tri, combinatoire (Pair et Gaudel [9], Aho et Ullman [10]).

La diversité des centres d'intérêt justifie donc une étude syntaxique ([11], [12], [13], [14], [15]); c'est-à-dire une étude de ces objets pour eux-mêmes, indépendamment de leur signification dans tel ou tel domaine.

Notre étude des graphes s'inscrit donc dans une volonté commune à de nombreux informaticiens d'horizons différents, qui est de maîtriser des structures complexes (Claus, Ehrig, Rozenberg [18]; Janssens, Rozenberg [16]; Raoult [17]; Ehrig, Nagl, Rozenberg [19]). En effet les graphes apparaissent adaptés à décrire la structure profonde d'objets complexes : réseau, schéma de programmes parallèles, ... Mais leur utilisation syntaxique et systématique est freinée par le manque de cadre formel simple pour les définir et les manipuler pratiquement. La complexité des formalismes et des techniques opératoires définies jusqu'à présent s'illustre bien dans l'extension des arbres aux graphes des systèmes de réécriture (faire une réécriture $g \rightarrow d$ consiste à remplacer dans un contexte un sous-objet g : sous-mot, sous-arbre, sous-graphe, par un autre d). La généralisation de cette notion aux graphes pose de sérieux problèmes formels comme par exemple la définition de sous-graphe correctement inséré dans un autre. Ehrig [23] a bien spécifié le problème en termes de catégories; Raoult [17] et Courcelles ([20], [21]) ont dans le même esprit simplifié l'approche, mais malgré cela il reste malaisé de calculer pratiquement dans les graphes.

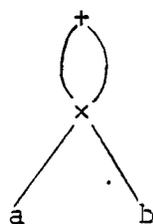
Situons donc nos choix par rapport à cette complexité formelle pour l'heure encore redhibitoire, d'une utilisation calculatoire des graphes. Nous ne considérerons que des DAGS (Directed Acyclic Graphs) construits sur un alphabet bigradué où les lettres représenteront les noeuds des graphes, la coarité d'une lettre représentera le nombre d'arcs entrant pour ce noeud et la arité, le nombre d'arcs sortant. Une fois payé le prix de cette restriction, nous disposons d'un cadre formel commun avec celui des arbres : les magmoïdes (Arnold, Dauchet [1] et [2]). Cette similitude entre arbres et dags est du point de vue formalisme, la clé de ce travail. Les objets manipulés - les dags - sont les objets du magmoïde libre engendré par un alphabet bigradué donné. Signalons que nous escamoterons dans cette thèse le garde-fou conceptuel que pourrait être l'approche catégorique (Eilenberg et Wright [22], Ehrig [23]). Quels que soient les mérites unificateurs de cette approche, et bien qu'elle aurait pu nous dispenser d'explicitier quelques constructions standards, son intérêt, ici, eut été limité : la plupart de nos résultats "fins" échappant totalement à ce cadre global.

B.2. DES EXEMPLES SUR LES DAGS

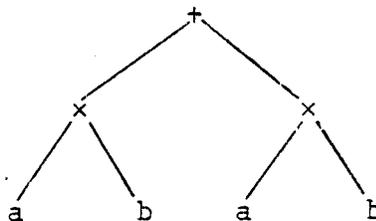
Nous allons donner quelques exemples présentant les dags en tant que structures déployant des calculs. Ces exemples ne sont évidemment pas destinés à étayer l'intérêt pratique de notre travail mais prétendent simplement introduire intuitivement ces objets en les chargeant d'une interprétation immédiate.

B.2.1. Il est bien connu que les arbres suffisent pour décrire la sémantique des calculs mais pas leur complexité. On constate que seuls des DAGS peuvent intégrer dans leur formalisme le partage de ressources ou de résultats.

Ainsi le dag



est sémantiquement équivalent à l'arbre



si on interprète l'opérateur \times comme la fonction suivante :

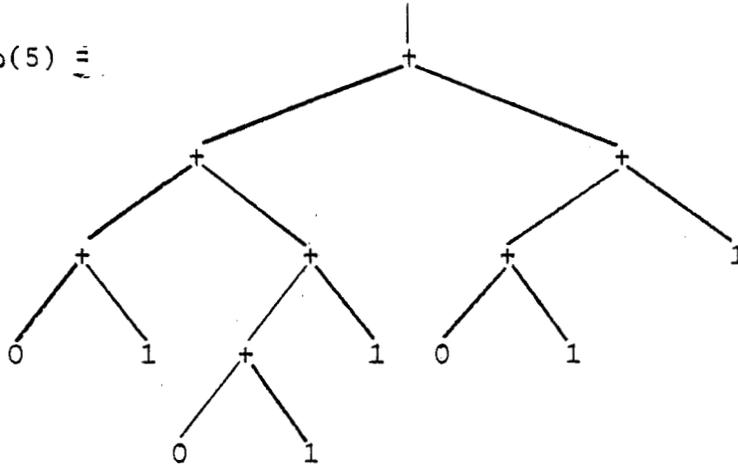
$$X(x, y) = (x \times y, x \times y).$$

Cette idée a conduit à l'introduction des DAGS dans de nombreuses études liées à l'implémentation et à la complexité (Aho, Ullman [20]), en λ -calcul (Raoult [17]) et dans les mécanismes d'unification ([25], [26], [24]).

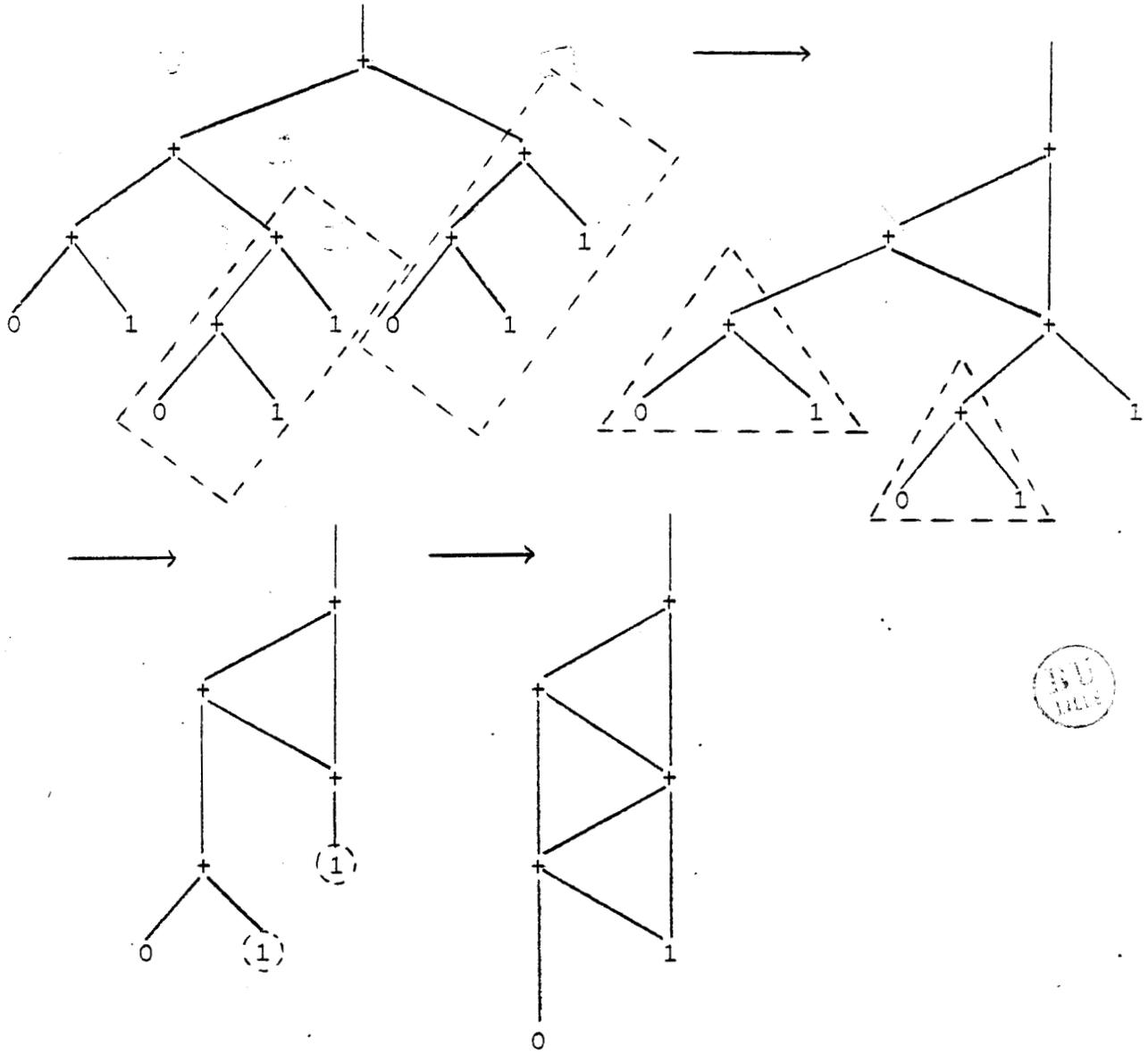
On peut appliquer cette idée, par exemple, dans l'optimisation du calcul des termes de la suite de Fibonacci. La suite de Fibonacci est une suite récurrente à laquelle est associée la fonction récursive $\text{Fib}(n) = +(\text{Fib}(n-1), \text{Fib}(n-2))$ avec $\text{Fib}(0) = 0$; $\text{Fib}(1) = 1$.

Déployons l'arbre de $\text{Fib}(5)$:

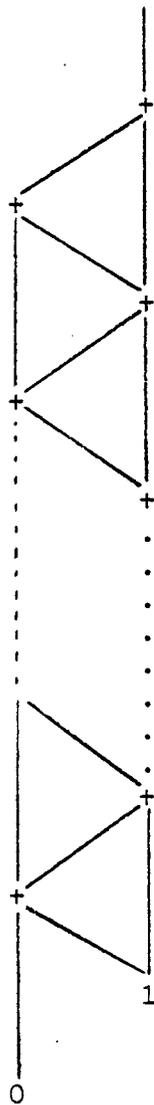
Fib(5) =



Essayons de rendre compte, à l'aide de dags, des partages de résultats possibles dans ce calcul.



La forme générale du dag de calcul d'un terme de la suite de Fibonacci serait :



Les dags obtenus traduisent une optimisation des calculs, pratique désormais courante en compilation.

B.2.2. De même que l'on parle d'arbre de dérivation pour les mots d'un langage algébrique, on peut introduire la notion de dags de dérivation pour les mots générés par une grammaire à structure de phrase. Cette représentation spatiale des dérivations dans une grammaire à structure de phrases permettrait de prolonger la relation d'équivalence définie sur les dérivations des grammaires algébriques. Deux dérivations seront dites équivalentes si et seulement si elles ont mêmes dags de dérivations. Cette approche a déjà fait l'objet d'études (Stanat [28]).

Illustrons ce point de vue sur l'exemple suivant :

Soit la grammaire $G = (\Sigma, N, P)$ où

$\Sigma = \{a, b, c, \dots, z\}$;

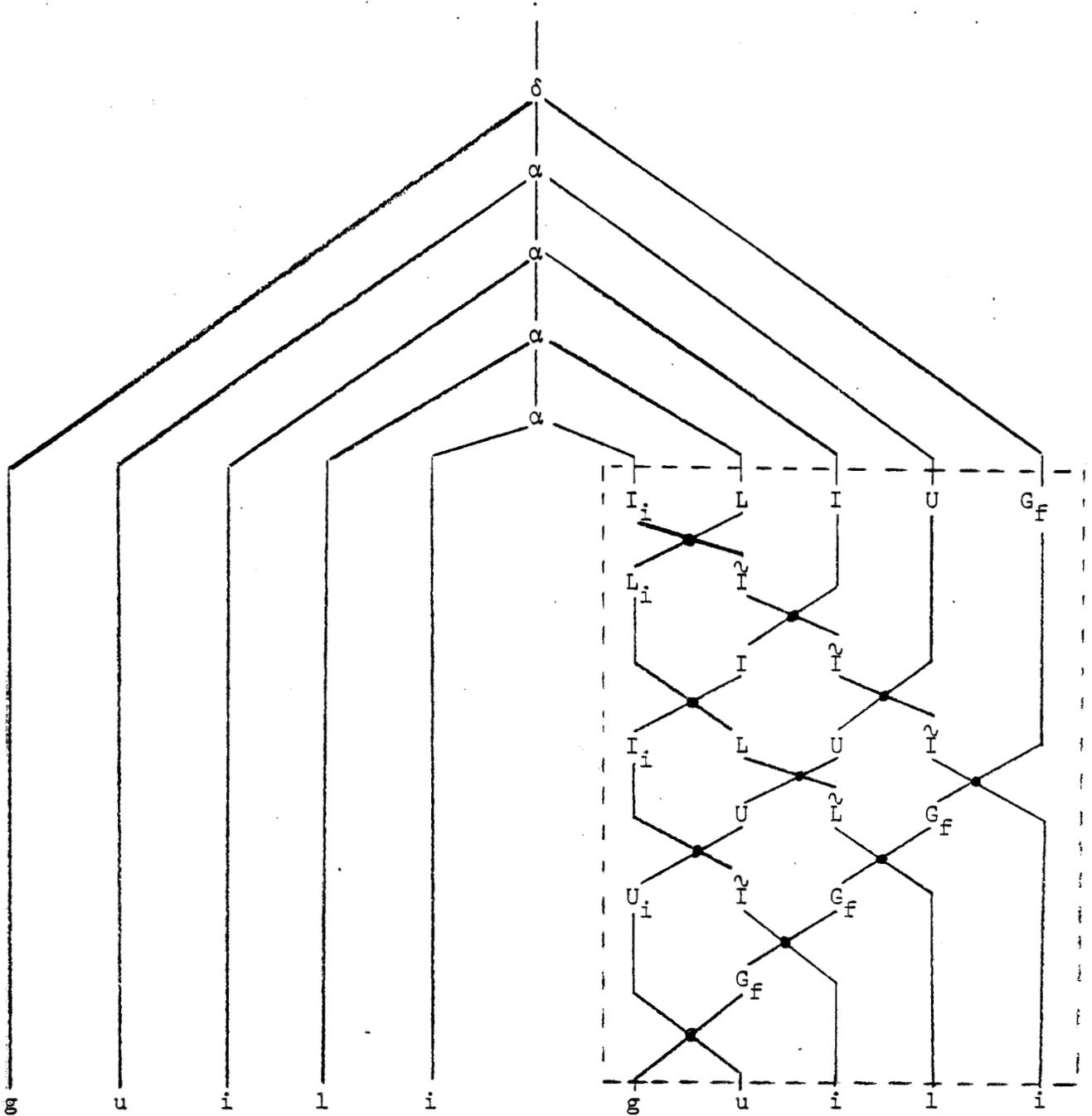
$N = \{A, \dots, Z\} \cup \{A_i, \dots, Z_i\} \cup \{A_f, \dots, Z_f\} \cup \{\tilde{A}, \dots, \tilde{Z}\} \cup \{\alpha, \delta\}$

P comprend les familles de règles suivantes :

- i) $\forall x \in \Sigma \quad \delta \rightarrow x\alpha\bar{X}_f$
- ii) $\forall x \in \Sigma \quad \alpha \rightarrow x\alpha X \text{ ou } xX_i$
- iii) $\forall x, y \in \Sigma \quad X_i Y \rightarrow Y_i \bar{X}$
- iv) $\forall x, y \in \Sigma \quad \tilde{X} Y \rightarrow Y \tilde{X}$
- v) $\forall x, y \in \Sigma \quad \tilde{X} Y_f \rightarrow Y_f X$
- vi) $\forall x, y \in \Sigma \quad \tilde{X}_i Y_f \rightarrow yx$

Le langage engendré par cette grammaire à partir de l'axiome δ est l'ensemble des mots mm ; $m \in \Sigma^*$. Le nombre de règles, la relative lourdeur du formalisme rend la preuve difficile à écrire et surtout éloignée de l'intuition que l'on peut se faire du phénomène. Examinons quelles allures possèdent les dags de dérivations issus des dérivations dans cette grammaire.

Le mot "guiliguili" est obtenu par cette grammaire à partir de la dérivation dont le dag est :



On induit de façon assez naturelle que la forme des dags obtenus est conservée pour tous les mots de même longueur, et que ce schéma s'agrandit "régulièrement" lorsque la taille des mots croît. Il apparaît donc un caractère unificateur dans cette famille de dags : le lien "géométrique" entre les parties supérieures de ces dags est évident et les parties inférieures ressemblent toutes à un "maillage régulier".

B.2.3. Regardons maintenant comment un certain nombre de "calculs" peuvent être considérés comme l'interprétation de familles de dags possédant un caractère de régularité dans leurs formes.

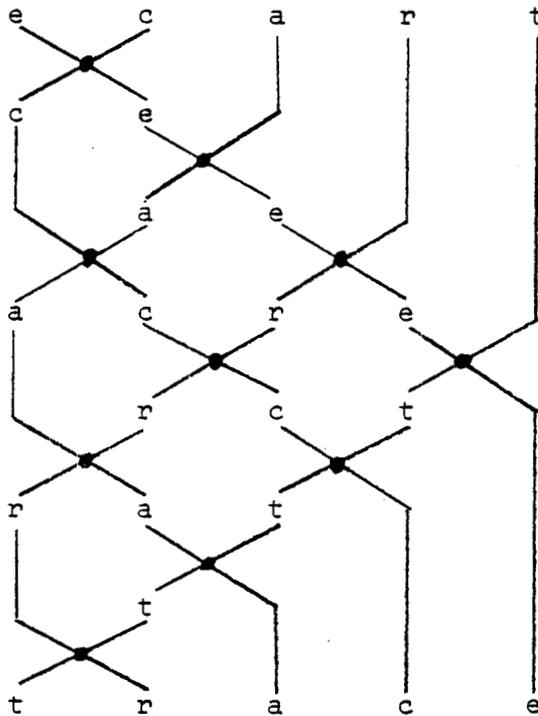
B.2.3.1. Reprenons la famille de dags mise en évidence lors de l'exemple B.2.2. et attribuons au symbole \times la fonction de permuter deux lettres quelconques :

quelques soient a et b appartenant à un alphabet donné :

$$\times(a, b) = (b, a).$$

Les dags de cette famille représentent alors l'ensemble des schémas de calcul de la transformation qui, à un mot, associe son miroir. De plus, cette famille de dags est caractéristique d'un algorithme particulier effectuant cette transformation.

Exemple :

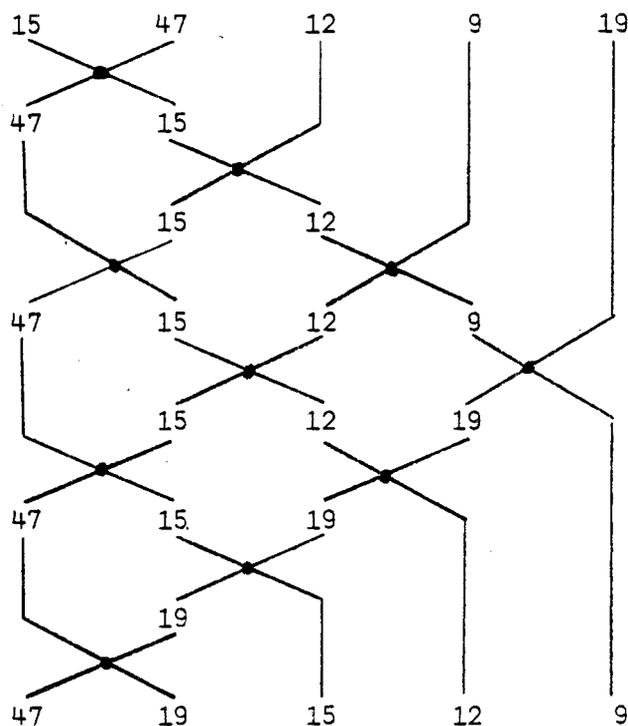


B.2.3.2. Remarquons que la même famille de dags, pour laquelle on accorderait au symbole \times l'interprétation suivante :

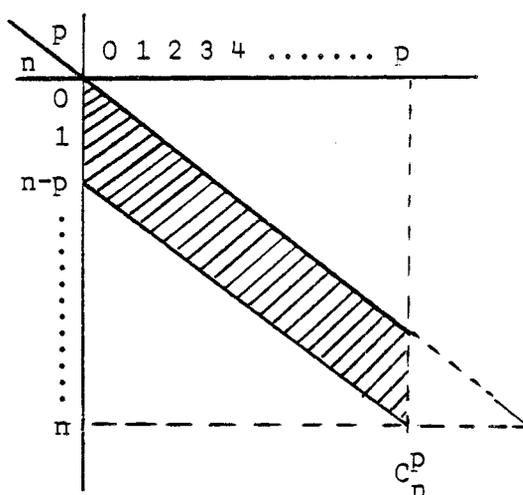
$$\forall n, p \in \mathbb{R} \quad \times(n, p) = \begin{array}{l} \text{si } n \geq p \text{ alors } (n, p) \\ \text{sinon } (p, n) \\ \text{fsi} \end{array}$$

représenterait alors l'ensemble de schémas de calcul du tri d'une série de nombres réels selon une méthode que l'on pourrait assimiler au "tri-bulles".

Exemple :

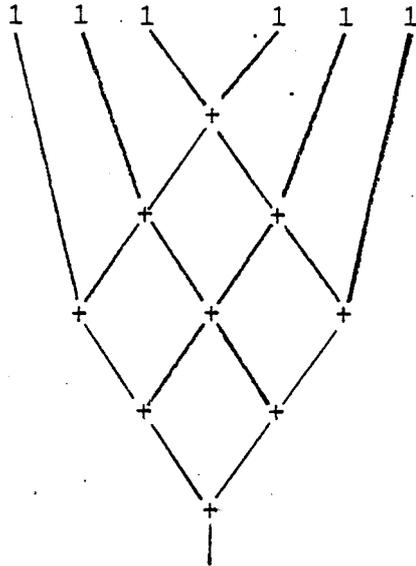


B.2.3.3. Ce dernier exemple s'appuie sur une méthode de calcul de C_n^p par un algorithme qui "construit" la partie minimale du triangle de Pascal qui permette de déterminer le C_n^p désiré. L'élaboration du triangle de Pascal repose sur l'application de la formule $C_n^p = C_{n-1}^p + C_{n-1}^{p-1}$; à partir de cela on peut aisément mettre en évidence la partie utile du triangle de Pascal au calcul d'un C_n^p



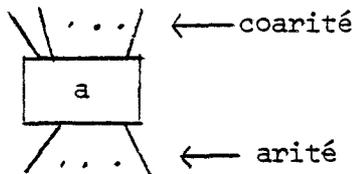
Représentons cette succession de calculs par un graphe planaire dont le noeuds seront interprétés tantôt comme l'opérateur classique d'addition, et d'autres fois comme un opérateur d'addition qui duplique la somme obtenue.

Cela donnerait pour le calcul de C_7^3 .

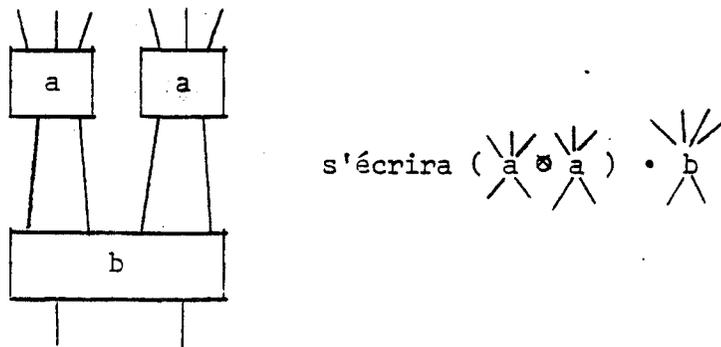


On remarque, aussi que cet exemple, que l'ensemble des schémas de calcul des C_n^p respectent une forme de "régularité".

Dorénavant, nous abandonnons toute interprétation pour des considérations purement syntaxiques. Nous ne manipulerons que des lettres bigraduées, i.e. des lettres auxquelles sont associées un certain nombre d'entrées appelé co-arité et un certain nombre de sorties, appelé arité. Ces lettres représenteront de noeuds ou sommets de nos graphes et se dessineront :



Elles seront composées en parallèle grâce aux opérateurs \otimes et \odot ou en série par l'opérateur \cdot , pour obtenir les objets désirés. Ainsi le dag planaire



Et nous étudierons les propriétés "formelles" de ces objets, i.e. valables quelque soit la signification (interprétation) donnée aux lettres. Cette démarche est celle suivie dans l'étude des structures plus simples de la théorie des langages (voir par exemple Hopcroft, Ullman [29]) et des forêts (voir par exemple : Brainerd [31], Eilenberg [32], Engelfriet [33], Maibaum [34], Rounds [35], [36]).

B.3. LA RATIONALITE ET LA RECONNAISSABILITE DANS LES MOTS ET LES ARBRES

B.3.1. Le cas des mots

L'étude de ces notions sur les mots se situe dans le cadre formel d'un monoïde libre. Un monoïde est un triplet formé d'un ensemble M , d'une loi de composition interne associative et d'un élément distingué de M que nous noterons ε . Cet élément ε est l'élément neutre de la loi de composition interne appelée concaténation. Soit A un ensemble fini appelé alphabet, dont les éléments sont appelés lettres ; on notera A^* le monoïde libre sur A . Pratiquement A^* est construit de la manière suivante : c'est l'ensemble des suites finies de lettres ou mots. On note un mot par simple juxtaposition de lettres : $a_1 a_2 \dots a_n$; la concaténation de deux mots $a_1 a_2 \dots a_n$ et $b_1 b_2 \dots b_p$ est le mot $a_1 a_2 \dots a_n b_1 b_2 \dots b_p$. On appelle langage (sur l'alphabet A) une partie du monoïde libre A^* . On peut définir sur les langages une grande quantité d'opérations. Outre les opérations booléennes classiques (union finie, intersection finie, complémentation), on utilise aussi l'opération produit et étoile. Le produit (de concaténation) de deux langages L et K est le langage :

$$LK = \{uv \in A^* / u \in L \text{ et } v \in K\}.$$

L'étoile d'un langage L de A^* , notée L^* , est le sous-monoïde de A^* engendré par L . Les opérations produit, union, et étoile permettent de spécifier certains langages définis par des propriétés combinatoires. Ainsi, si $A = \{a, b\}$, l'ensemble des mots dans lesquels ne figurent jamais deux "a" consécutifs peut s'exprimer sous la forme : $(ab + b)^* \cdot (a + \epsilon)$. On peut tenter de déterminer la puissance de ces opérations ; on définit alors l'ensemble des langages rationnels comme le plus petit ensemble de langages de A^* qui contienne les langages réduits à un mot et qui soit clos par union, produit et étoile.

Parallèlement, on s'intéresse à une classe d'algorithmes, qui agissent sur les mots d'un monoïde libre, que l'on appelle "automate d'états finis". Un automate d'états finis est un algorithme qui :

- reçoit une donnée constituée d'une suite de symboles,
- lit cette donnée sans retourner en arrière,
- dispose d'un nombre fini d'états pour qualifier l'information qu'il retient sur ce qu'il a déjà lu,
- produit (éventuellement : transducteur séquentiel) au fur et à mesure son résultat sous forme d'une nouvelle suite de symboles (Perrin [57]).

Le formalisme adopté, dans le cas des automates d'états finis de mots est :

- un automate d'états finis et un quintuplet $C = \langle A, Q, q_0, F, \square \rangle$ où
 - A est un alphabet fini,
 - Q est un ensemble fini d'états,
 - q_0 état de Q , appelé état initial,
 - F sous-ensemble de Q , tout état appartenant à F est appelé état final,
 - \square est une relation de $Q \times A$ dans Q . Cette relation est étendue par associativité à A^* de la façon suivante :

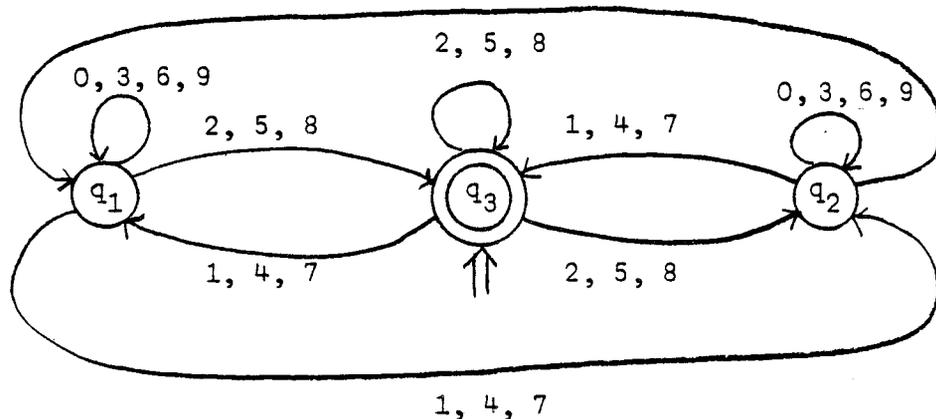
$$q \square \lambda = q \text{ et}$$

$$\forall w \in A^* \quad \forall a \in A \quad q \square (wa) = (q \square w) \square a.$$

On dit qu'un langage L est reconnu par l'automate C si et seulement si $L = \{w \in A^* / q_0 \square w \in F\}$. Tout langage reconnu par un automate est dit reconnaissable, l'ensemble des langages reconnaissables est habituellement appelé REC.

Notons, qu'en préservant le langage reconnu, on parvient toujours à rendre déterministe et minimal un automate d'états finis de mots (i.e. la relation \square devient une application de $Q \times A$ dans A et le nombre d'états est le plus petit possible). L'informaticien dispose donc, pour un processus de reconnaissance, de l'algorithme de complexité minimale.

Par exemple, l'ensemble des multiples de 3, considérés comme des mots du monoïde libre engendré par l'alphabet $\{0, 1, 2, \dots, 9\}$ est un langage reconnaissable. Un examen attentif de l'automate ci-dessous convaincra sans doute le lecteur qu'il reconnaît bien les multiples de 3.



Un résultat fondamental de la théorie des langages rapproche ces deux notions de reconnaissabilité et de rationalité : le théorème de Kleene. Ce théorème affirme que tout langage rationnel est reconnaissable et vice-versa. De plus, le passage de la caractérisation reconnaissable (automate) à la caractérisation rationnelle (expression rationnelle) s'effectue par résolution d'un système régulier d'équations algébriques.

Ainsi l'expression rationnelle représentant le langage des multiples de trois est extraite de la résolution de (S) :

$$(S) \begin{cases} N_1 = \bar{1} + N_1 \bar{0} + N_2 \bar{2} + N_3 \bar{1} \\ N_2 = \bar{2} + N_1 \bar{1} + N_2 \bar{0} + N_3 \bar{2} \\ N_3 = \bar{3} + N_1 \bar{2} + N_2 \bar{1} + N_3 \bar{0} \end{cases}$$

dans lequel N_i désigne le langage reconnu par l'automate avec q_3 comme état initial et q_1 comme état final et $\bar{0}$, $\bar{1}$, $\bar{2}$ représentent respectivement les ensembles $\{0, 3, 6, 9\}, \{1, 4, 7\}, \{2, 5, 8\}$.

- En remarquant que la solution d'une équation de la forme

$$X = XA + B \quad (\text{où } A, B \text{ sont des langages})$$

est le langage BA^* .

- En appliquant à ce système un procédé de résolution de type "substitution".

On obtient une expression rationnelle représentative de N_3 :

$$(\bar{0} + (\bar{1} \bar{0}^* \bar{1} + \bar{2})) \cdot (\bar{0} + \bar{2} \cdot \bar{0}^* \cdot \bar{1})^* \cdot (\bar{2} \cdot \bar{0}^* \cdot \bar{2} + \bar{1}) + \bar{1} \cdot \bar{0}^* \cdot \bar{2})^*$$

Bien que cette dualité rationnelle-reconnaissable soit encore vraie dans les arbres, et les dags, on ne retrouvera pas dans les arbres, et moins encore dans les dags les facilités et la souplesse d'utilisation de ces concepts dans les mots.

B.3.2. Le cas des arbres

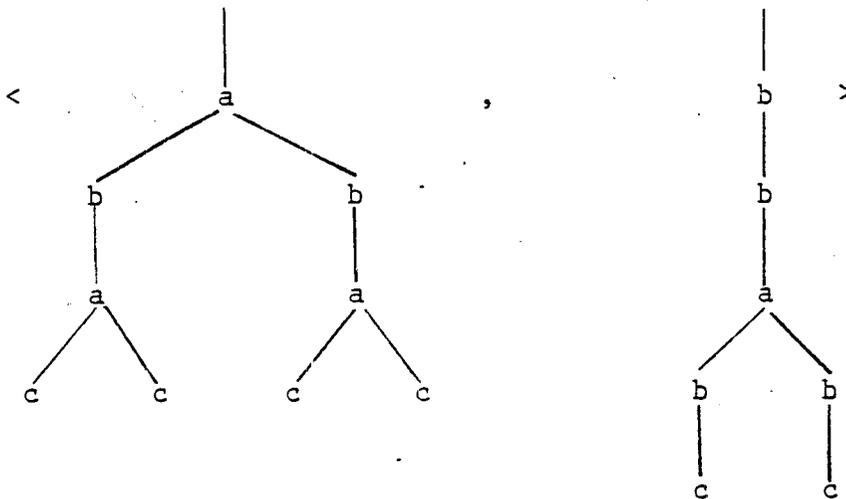
La formalisation par une structure algébrique de la notion d'arbres conduit à la structure de Σ -magmas (Nivat [93], de Σ -algèbre (Eilenberg [32], Thatcher J.W. [15]) ou à celle de magmaïde (Arnold, Dauchet [1], [2]). Dans le Σ -magma libre et la Σ -algèbre libre (les seuls qui nous intéressent ici) on ne considère qu'un seul opérateur portant sur des lettres graduées : la composition. On définit les objets (termes ou arbres) de la façon suivante :

- si $a \in \Sigma_0$ alors a est un terme, où Σ_0 est le sous-ensemble de Σ des lettres de arité 0
- si t_1, \dots, t_n sont des termes et T est une lettre de degré n (en fait T est considéré comme un opérateur à n opérandes) alors $T(t_1, t_2, \dots, t_n)$ est un terme.

Dans les magmoïdes on introduit, outre l'opération de composition énoncée ci-dessus, un opérateur noté \otimes de mise en parallèle. Un élément du magmoïde est alors une suite finie d'arbres donc un objet à plusieurs entrées et à plusieurs sorties (on voit par là comment la structure de magmoïde pourra aussi rendre compte des DAGS). Plus formellement :

- Un magmoïde est un quintuplet $M = \langle M, \cdot, \otimes, e, e_0 \rangle$ - qui est noté M suivant l'usage consistant à identifier une structure algébrique et son support - où M est un ensemble, \cdot une opération binaire partielle sur M , \otimes une opération binaire sur M , e et e_0 deux éléments distingués de M . De plus, M vérifie cinq axiomes que l'on trouvera en (Arnold, Dauchet [1]).

Ainsi la suite d'arbres



s'écrit dans le magmoïde libre sur Σ , noté $\tilde{T}(\Sigma)$

$$\langle a \cdot (b \otimes b) \cdot (a \otimes a) \cdot (c \otimes c \otimes c \otimes c), b \cdot b \cdot a \cdot (b \otimes b) \cdot (c \otimes c) \rangle$$

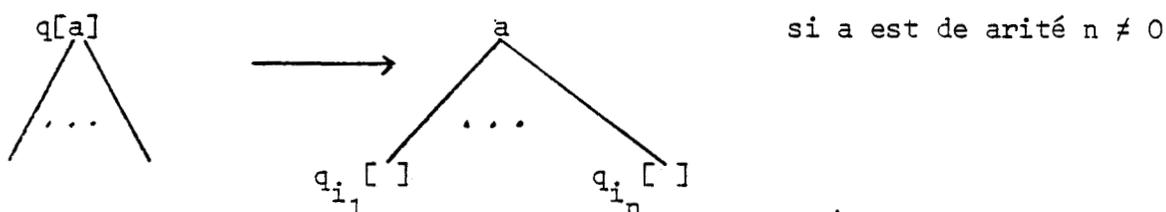
Remarque : Les arbres finis considérés comme élément du magmoïde libre n'utilise pas toute la puissance de cette structure et se limite à en être une partie.

Les automates d'états finis d'arbres reprennent les mêmes principes (Thatcher [15], Engelfriet [33]) que les automates d'états finis de mots, mais appliqués à la reconnaissance de lettres graduées. De plus, un arbre n'étant pas un objet "symétrique", deux points de vue duaux sont possibles : les automates des-

endants et les automates ascendants. Dans le cadre du magmaïde, rappelons succinctement quelques définitions

Un automate descendant d'états finis d'arbres est un quadruplet $M = \langle \Sigma, Q, I, R \rangle$ où

- Σ est un alphabet gradué fini
- Q , un ensemble fini d'états
- $I \subseteq Q$, un ensemble d'états distingués, appelés états initiaux
- R , un ensemble fini de règles de la forme

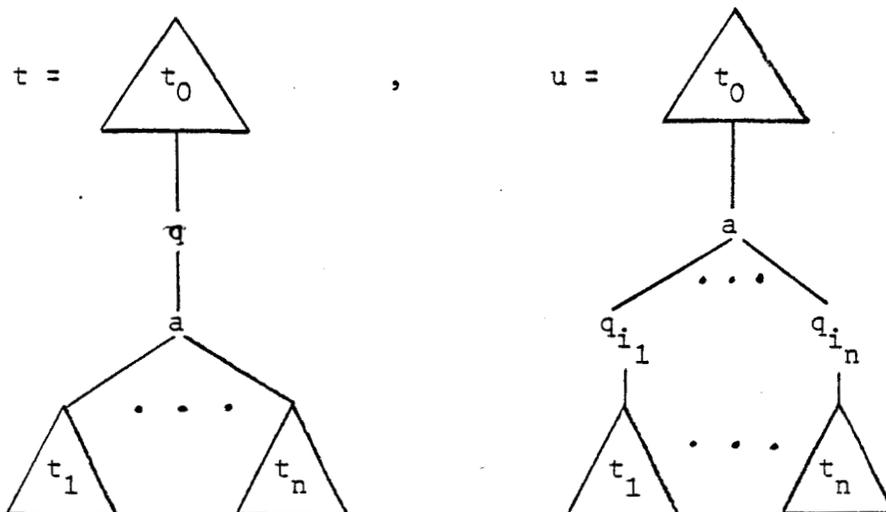


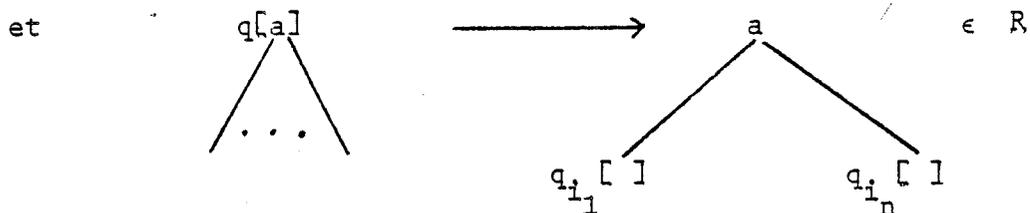
ou

$q[a] \longrightarrow a$ si a est de arité nulle.

En assimilant, pour des raisons de facilité d'écriture, Q à des lettres graduées de arité 1, on définit la relation $\stackrel{1}{\underset{M}{\mid}}$ sur $T(\Sigma \cup Q)_0^1$ par :

$\forall t, \forall u \in T(\Sigma \cup Q)_0^1, t \stackrel{1}{\underset{M}{\mid}} u$ ssi on a :





On dit que $t \in T(\Sigma)_0^1$ est reconnu par M ssi $\exists q \in I$ tel que $q \cdot t \stackrel{*}{\mid}_M t$, où $\stackrel{*}{\mid}_M$ désigne la clôture transitive de \mid_M . On note $F(M)$ la forêt reconnue par M . L'automate ascendant s'obtient en "inversant le sens des flèches" des règles de R et l'ensemble I , alors noté F , est appelé ensemble des états finaux. La relation \mid_M est modifiée de façon à effectuer la reconnaissance de "bas en haut". On dit que $t \in T(\Sigma)_0^1$ est reconnu par M , ascendant, si et seulement si, $\exists q \in F$ tel que $t \stackrel{*}{\mid}_M q[t]$.

Ainsi une forêt est reconnaissable si et seulement si elle peut être reconnue par un automate d'états fini d'arbres qu'il soit ascendant ou descendant.

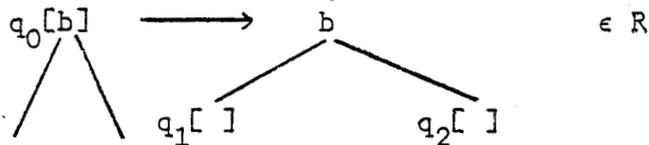
Les résultats élémentaires sur la classe des forêts reconnaissables notée REC, sont :

- les clôtures par union, intersection, homomorphisme inverse, homomorphisme linéaire, complémentaire (Thatcher [15]),
- dans le cas ascendant, la construction effective d'un automate minimal (Thatcher [15]).
- la réduction du non-déterminisme des automates ascendants, cf exemple ci-dessous et (Thatcher [15]).

Exemple : Soit la forêt reconnaissable $FR = \left\{ \begin{array}{c} b \\ / \quad \backslash \\ a \quad a \end{array} , \begin{array}{c} b \\ / \quad \backslash \\ a' \quad a' \end{array} \right\}$, alors il n'existe

pas d'automate descendant déterministe $M = \langle \Sigma, Q, f, R \rangle$ (i.e. f est réduit à un seul état et deux règles distinctes de R ont des membres gauches différents) tel que M reconnaisse FR . En effet :

- I réduit à un seul état, soit q_0 cet état, on a nécessairement

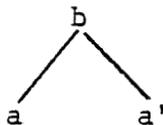


- et ainsi pour reconnaître FR, on a nécessairement

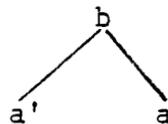
$$q_1[a] \rightarrow a ; q_2[a] \rightarrow a$$

$$q_1[a'] \rightarrow a ; q_2[a'] \rightarrow a'$$

mais alors l'automate reconnaît aussi les arbres :



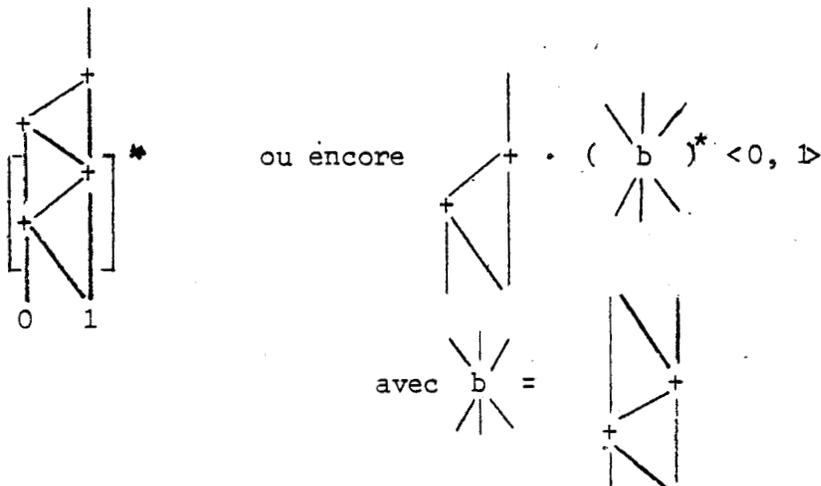
et



□

d) la propriété fondamentale du feuillage : il y a identité entre la classe des langages algébriques sans mot vide et la classe des feuillages des forêts reconnaissables (Thatcher [15]).

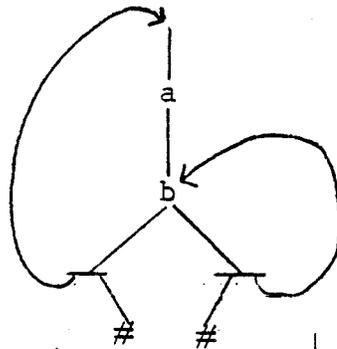
La notion de rationalité définie dans les langages (spécification de langages à l'aide des opérateurs d'union (+), de composition (•) et d'itération de composition (*) équivalente à la reconnaissabilité) ne peut être reprise telle quelle dans les magmoïdes que dans le cas des "fibres carrées", i.e. d'objets ayant une arité et une coarité égales (A. Arnold [44]). L'exemple B.2.1. rentre dans ce cadre, et on peut le décrire par :



l'étoile est dans ce cadre défini sur des objets "n-n", (ici n = 3).

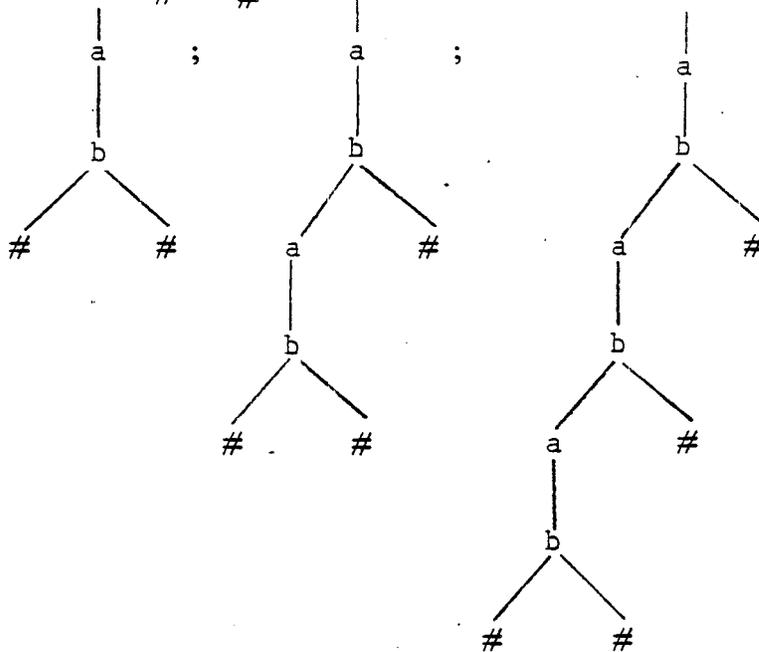
Mais il n'est pas possible en général de décrire une forêt reconnaissable à l'aide des trois opérateurs usuels : +, * et •. Le passage de la caractérisation reconnaissable (automate) à la caractérisation rationnelle (expression rationnelle) nécessite, comme l'illustre l'exemple ci-après, de "typer" l'opérateur * [VI.5.1. dans la thèse].

Exemple : Soit la forêt composée des arbres de la forme



où  symbolise un choix

ainsi les arbres



;... sont des arbres de cette forêt.

Cette forêt est reconnue par l'automate descendant $M = \langle \Sigma, Q, R, F \rangle$ où

- $\Sigma = \{ b, a, \# \}$
- $Q = \{ q_0, q_1, q_2 \}$
- $I = \{ q_0 \}$

et l'ensemble de règles R : $q_0[a] \rightarrow a$

$q_1[b] \rightarrow$; $q_1[b] \rightarrow$;

$q_1[b] \rightarrow$; $q_1[b] \rightarrow$

$q_2 \cdot \# \rightarrow \#$

En notant N_i , la forêt obtenue à partir de l'état q_i , on obtient le système d'équations

$$N_0 = a \cdot N_1$$

$$N_1 = b \cdot (N_0, N_1) + b \cdot (N_0, N_2) + b \cdot (N_2, N_1) + b \cdot (N_2, N_2)$$

$$N_2 = \#$$

ce qui donne (sans typer *)

$$N_0 = [a \cdot [b \cdot (e + \#, e + \#)]^*]^* \quad \text{où } e \text{ est l'élément neutre du produit de composition}$$

mais dans cette expression, on ne sait pas à quel endroit faire opérer l'opérateur *, en réalité il faut exprimer :

$$N_0 = [a \cdot [b \cdot (e + \#, e + \#)]^*]^*$$

d'où l'introduction de l'opérateur "étoile typée"

$$N_0 = [a \cdot [b \cdot (t_2 + \#, t_1 + \#)]^{*t_1}]^{*t_2}$$

B.4. Le contenu de notre travail

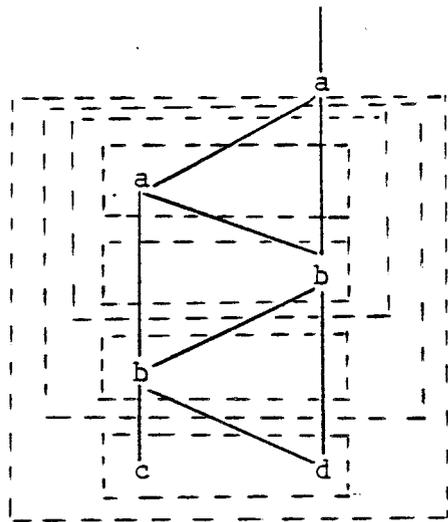
B.4.1. Le magmaïde des dags planaires : $\overset{\vee}{\text{DP}}(\Sigma)$

On peut voir l'étude successive des mots, puis des arbres comme autant de pas vers l'étude générale des objets syntaxiques d'un magmaïde libre quelconque. En effet, si M est un magmaïde, la fibre 1-1 de ce magmaïde muni de l'opérateur produit de composition et de l'élément e est un monoïde libre : cadre formel de l'étude des langages de mots. L'étude des arbres s'inscrit également dans l'étude des objets d'un magmaïde puisqu'un Σ -magma peut être muni d'une structure de magmaïde. Les éléments du magmaïde libre $\overset{\vee}{\text{DP}}(\Sigma)$ engendré par l'alphabet Σ bigradué sont obtenus comme combinaisons de produit de composition et de produit tensoriel de lettres bigraduées.

Exemple : Si $\Sigma_2^1 = \{a\}$, $\Sigma_2^2 = \{b\}$, $\Sigma_0^1 = \{c\}$, $\Sigma_0^2 = \{d\}$

$$\delta = a \cdot (((a \otimes e) \cdot (e \otimes b)) \cdot (b \otimes e)) \cdot (c \otimes d) \in \overset{\vee}{\text{DP}}(\Sigma).$$

Comme pour les mots et les arbres, on s'évertue à donner à ces objets syntaxiques une présentation plus intuitive sous forme de dessins. Ainsi, on assimilera le produit de composition à une mise en "série" de deux schémas, et le produit tensoriel à une mise "en parallèle" de deux schémas. On obtient alors comme représentation graphique de l'élément δ du magmaïde mentionné plus-haut :



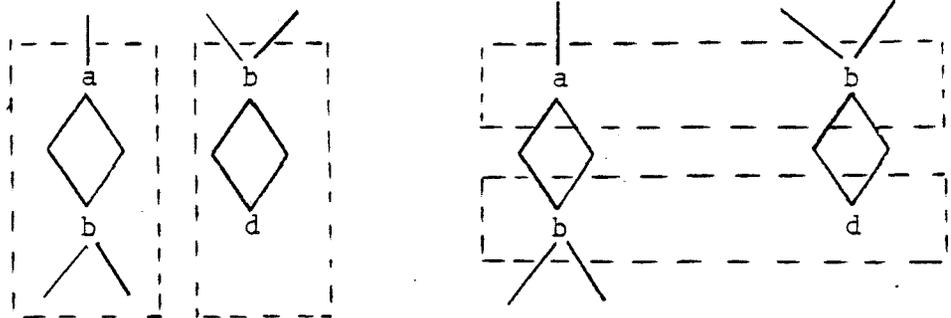
Ce type de dessin s'apparente à un graphe orienté (de haut en bas) et sanscircuit, soit "directed acyclic graph" ou DAG. Par la suite, les éléments du magmaïde libre $\hat{DP}(\Sigma)$ seront systématiquement appelés dag.

Une des préoccupations principales lorsque l'on a à manipuler des objets syntaxiques, est d'exploiter les propriétés du cadre formel pour simplifier l'écriture littérale de ces objets. La suppression des systèmes de parenthèses issue de l'associativité d'opérateur est une manoeuvre extrêmement courante et conduit, par exemple, à l'abandon total des parenthèses dans les écritures de mots. Dans les dags, certains jeux de parenthèses ne peuvent être effacés sans introduire d'ambiguïté.

Exemple : L'écriture $a \circ a \cdot b \circ b$ peut être celle de $a \circ (a \cdot b) \circ b$, ou celle de $(a \circ a) \cdot (b \circ b)$.

On peut, par un système de priorités, comme il est d'usage dans les expressions arithmétiques, privilégier un opérateur par rapport à l'autre et ainsi écarter l'une des deux interprétations de $a \circ a \cdot b \circ b$. Dans le magmaïde, on accorde une priorité supérieure au produit de composition. Ainsi, on peut enlever les parenthèses dans l'écriture de $a \circ (a \cdot b) \circ b$, mais elles demeurent inévitables dans l'écriture de $(a \circ a) \cdot (b \circ b)$. Si les mots, structure à une dimension, peuvent être spécifiés sans équivoque par une écriture linéaire non parenthésée, on ne peut supprimer, dans l'écriture linéaire des dags (résultats de la composition d'opérateurs qui agissent de façon "perpendiculaire"), certaines informations (les parenthèses) indiquant sur quoi agissent ces opérateurs. Par contre, une représentation en dimension deux rend parfaitement compte de la composition des dags sans "solidariser" explicitement les opérands des opérateurs.

Exemple : $(a \cdot b) \circ (b \cdot d)$ et $(a \circ b) \cdot (b \circ d)$, qui sont deux écritures d'un même élément de $\hat{DP}(\Sigma)$, ont respectivement les représentations graphiques suivantes :



On constate que la suppression des encadrés en pointillés (transposition en dimension 2 des parenthèses) fait apparaître deux dessins semblables.

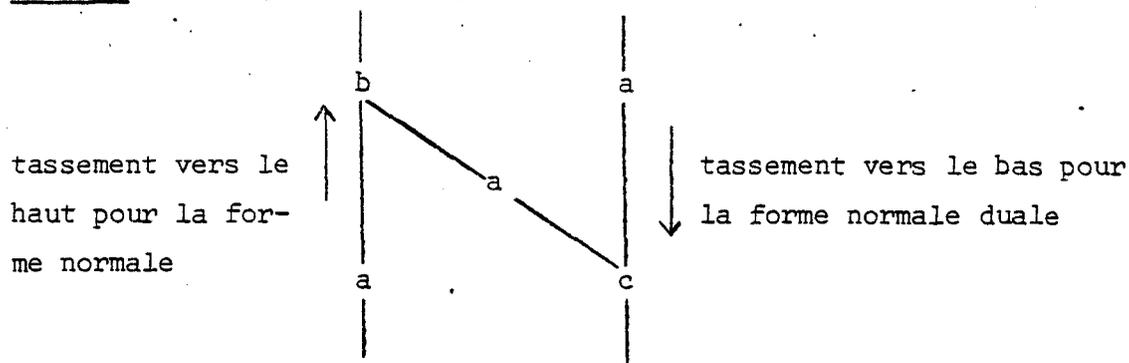
Nous montrerons en I.2. que l'axiomatique des magmoïdes se confond dans la représentation graphique de ses éléments avec une notion plus naturelle et plus directement perceptible : l'équivalence "structurelle" de deux dessins. On a par conséquent une représentation unique en dimension 2 de tout élément du magmoïde $\hat{DP}(\Sigma)$.

On peut néanmoins se poser la question de savoir si deux écritures linéaires distinctes représentent ou non un même élément du magmoïde. Cette question revêt un intérêt majeur pour des manipulations algorithmiques des éléments de $\hat{DP}(\Sigma)$. Or, ce problème tombe dans le cadre de ceux abordés par Knuth et Bendix dans leur fameux article: "Simple Words in Universal Algebras". Ils proposent un algorithme (partiel) capable de répondre à cette question ; cet algorithme repose sur :

- la définition d'un ordre bien fondé sur l'ensemble des écritures considérées,
 - l'orientation des identités axiomatiques dans le sens décroissant.
- Ces identités deviennent alors les règles d'un système de réécriture.

Dans certains cas, pour un choix judicieux de l'ordre, le système de réécriture ainsi défini est confluent (Huet [39]). Le problème de l'égalité de deux mots revient alors à comparer les réécritures irréductibles de ces deux mots par le système : si elles sont égales, les deux mots représentent le même élément. Cette dernière réécriture d'un mot est un terme privilégié pour l'ensemble des termes qui lui sont égaux ; on la distinguera en l'appelant forme normale. Cette méthode appliquée aux objets bigradués nous a permis de mettre en évidence dans les dags une forme normale ; cette forme normale est obtenue en "remontant" le plus haut possible les lettres et en faisant disparaître les étages composés uniquement d'éléments neutres. De l'interprétation imagée du fonctionnement du système de réécriture employé naît l'idée d'un autre système de réécriture au comportement symétrique par rapport au précédent (i.e. qui effectue un "tassement" vers le "bas" des lettres). Ceci conduit à l'élaboration d'une autre forme normale que nous appellerons forme normale duale.

Exemple : Le dag dont la représentation est :



a comme forme normale $(b \otimes a) \cdot (a \otimes a \otimes e) \cdot (e \otimes c)$
 et comme forme normale duale $(b \otimes e) \cdot (e \otimes a \otimes e) \cdot (a \otimes c)$.

B.4.2. Les automates de dags planaires

Comme dans les arbres, nous définissons, dans les dags planaires, les automates d'états finis en distinguant le cas descendant [II.1.1.] du cas ascendant [II.1.3.], mais ici contrairement aux arbres la symétrie parfaite entre le "haut" et le "bas" d'un dag nous assure de la dualité entre automate descendant et ascendant [II.1.3.] et signifie que toute propriété portant sur les automates descendants se retrouvera symétriquement dans le cas ascendant.

Un automate d'états finis descendant de dags planaires est un quintuplet $M = \langle \Sigma, Q, I, F, R \rangle$ où

- * Σ est un ensemble fini de lettres bigraduées de coarité et arité > 0 .
- * Q est un ensemble fini d'états bigradués de coarité et de arité éga-
lent à 1 et $\Sigma \cap Q = \emptyset$. On note Q^{\otimes} le monoïde libre (Q, \otimes) d'élément
neutre e_0 .
- * $I \subseteq Q^{\otimes}$, langage rationnel, appelé indifféremment contrôle haut ou con-
trôle initial.
- * $F \subseteq Q^{\otimes}$, langage rationnel, appelé indifféremment contrôle bas ou contrô-
le final.
- * R est un ensemble fini de règles $\subseteq Q^{\otimes} \times \Sigma \times Q^{\otimes}$ écrites sous la forme

$$(q_1 \otimes q_2 \otimes \dots \otimes q_n) \cdot c \rightarrow c \cdot (q'_1 \otimes \dots \otimes q'_n)$$

où $c \in \Sigma_n^n$.

On définit classiquement les mouvements de l'automate [II.1.1.2. à II.1.1.5.] comme une relation sur $\hat{D}P(\Sigma \cup Q)$, noté $\frac{*}{M}$ et le taillis reconnu par M , noté $T(M)$, est défini par :

$$T(M) = \{ \delta \in \overset{\vee}{DP}(\Sigma) / \exists q_i \in I, \exists q_f \in F \text{ avec } q_i \cdot \delta \stackrel{*}{\underset{M}{|}} \delta \cdot q_f \}$$

Exemple : L'automate d'états finis descendant de dags planaires $M = \langle \Sigma, Q, I, F, R \rangle$
où

$$* \Sigma = \left\{ \begin{array}{c} | \\ a \\ | \end{array} , \begin{array}{c} \diagup \\ b \\ \diagdown \end{array} \right\}$$

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

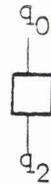
$$* I = F = q_0 \otimes (q_1 \otimes q_2)^{\otimes} \otimes q_0 \quad (\text{langage rationnel d'états})$$

* et R

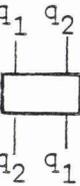
$q_0 \cdot a \rightarrow a \cdot q_1$ ou encore, écrit graphiquement



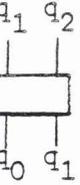
$q_0 \cdot a \rightarrow a \cdot q_2$ ou encore, écrit graphiquement



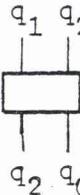
$(q_1 \otimes q_2) \cdot b \rightarrow b \cdot (q_2 \otimes q_1)$ ou encore, écrit graphiquement



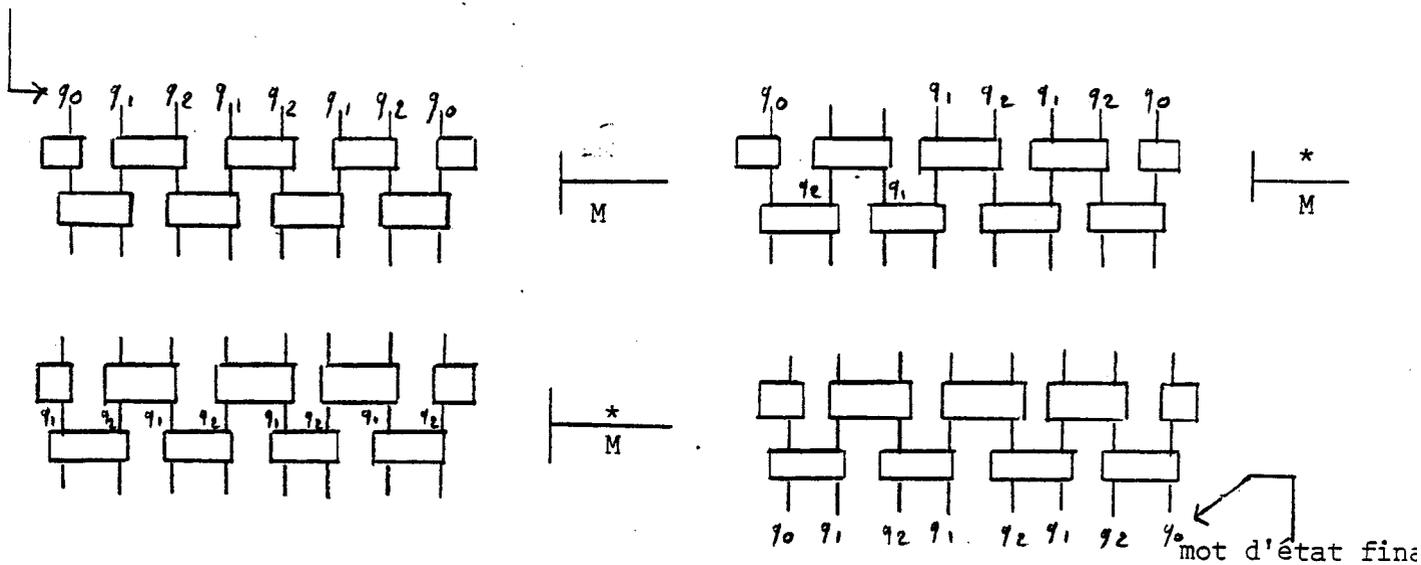
$(q_1 \otimes q_2) \cdot b \rightarrow b \cdot (q_0 \otimes q_1)$ ou encore, écrit graphiquement



$(q_1 \otimes q_2) \cdot b \rightarrow b \cdot (q_2 \otimes q_0)$ ou encore, écrit graphiquement



reconnait les dags "en forme de mur (de briques)" dont un exemple de reconnaissance peut se représenter graphiquement :



Le mot d'état commençant (ou contrôlant en haut) la reconnaissance est appelé le mot d'état initial (ou haut), et le mot d'état finissant (ou contrôlant en bas) la reconnaissance, est appelé le mot d'état final (ou bas).

Le cas ascendant est obtenu en "inversant le sens des flèches" des règles de R et $T(M) = \{\delta \in \hat{DP}(\Sigma) / \exists q_i \in I, \exists q_f \in F \text{ avec } \delta \cdot q_f \stackrel{M}{\sim} q_i \cdot \delta\}$.

Nous rencontrons deux problèmes formels au niveau de la définition des automates de dags planaires. Le premier problème concerne le statut des "conditions initiales" et des "conditions finales". Le deuxième, lié au premier, concerne les dags de degré 0-0.

1. Pour la notion de contrôle initial (resp. final), nous avons le choix entre, reprendre la notion d'état initial (resp. final) des automates de mots, ou de la généraliser.

Option a : On se donne $I \subseteq Q$ et $F \subseteq Q$ et les conditions d'une reconnaissance sont de commencer à partir d'états éléments de I pour aboutir en fin de reconnaissance à des états éléments de F.

option b : On se donne $I \subseteq Q^{\otimes}$ et $F \subseteq Q^{\otimes}$, donc des langages d'états, avec la contrainte qu'ils soient rationnels et les conditions d'une reconnaissance sont de commencer à partir d'un mot d'états élément de I pour aboutir à un mot d'états élément de F.

Nous adoptons la deuxième option, plus générale. Un résultat dans la thèse [V.2.] montrera que les deux options coïncident pour des taillis composés de dags connexes. En conséquence, le choix de ce point de la définition est secondaire, il ne porte que sur le contrôle de la mise en parallèle de dags déconnectés (non connexes).

Mais, outre ces justifications théoriques, nous pouvons illustrer sur un exemple, même simple, l'apport du choix de la seconde option.

Exemple : Soit le taillis fini $T = \{ \overset{|}{a} \otimes \overset{|}{a} \}$. Essayons de construire l'automate $M = \langle \Sigma, Q, I, F, R \rangle$ qui reconnaisse T . Alors il vient

$$q_0 \cdot a \rightarrow a \cdot q_f \in R$$

et, en se limitant à définir $I \subseteq Q$ et $F \subseteq Q$ (option a), il vient nécessairement

$$I = \{q_0\} \quad \text{et} \quad F = \{q_f\}$$

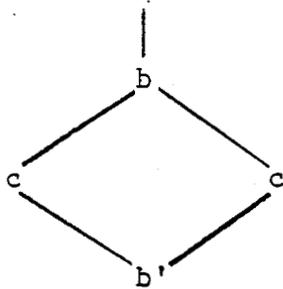
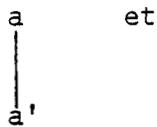
mais alors M reconnaît en plus les dags

$$\overset{|}{a} \otimes \overset{|}{a} \otimes \overset{|}{a}, \overset{|}{a} \otimes \overset{|}{a} \otimes \overset{|}{a} \otimes \overset{|}{a}, \dots$$

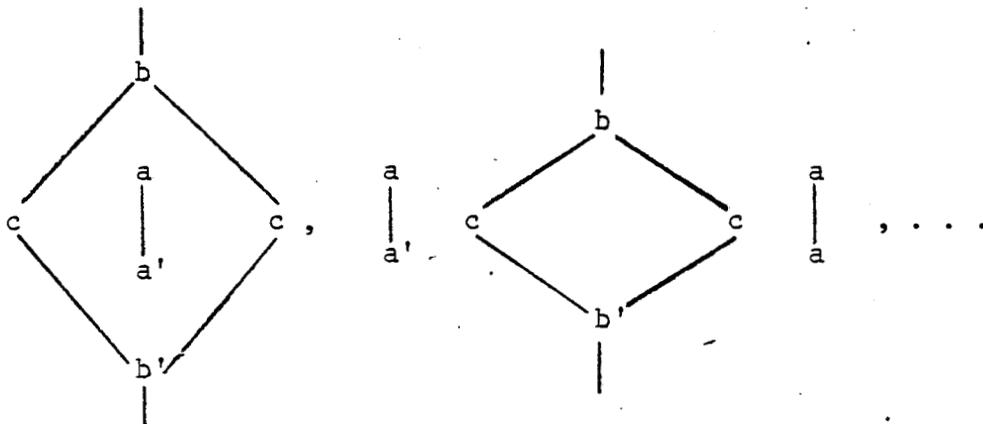
Par contre, en définissant (option b) $I = \{q_0 \otimes q_0\}$ et $F = \{q_f \otimes q_f\}$ on a : $T(M) = T = \{ \overset{|}{a} \otimes \overset{|}{a} \}$. L'exemple choisi, ci-dessus, pour mettre en défaut (l'option a) joue sur une caractéristique clé dans l'étude des dags : la CONNEXITE. Un dag D est connexe si et seulement si deux sommets distincts quelconques peuvent être reliés par une suite d'arcs ou encore plus formellement si et seulement si soit $D = e_0$, soit on ne peut écrire $D = d_1 \otimes d_2$ avec $d_1 \neq e_0$ et $d_2 \neq e_0$.

2. D'autre part, nous choisissons de ne considérer des taillis reconnaissables que sur des alphabets bigradués ne comprenant ni lettre de coarité nulle, ni lettre de arité nulle. Cette restriction est expliquée en [II.1.2.2.2.] et peut se résumer en deux points :

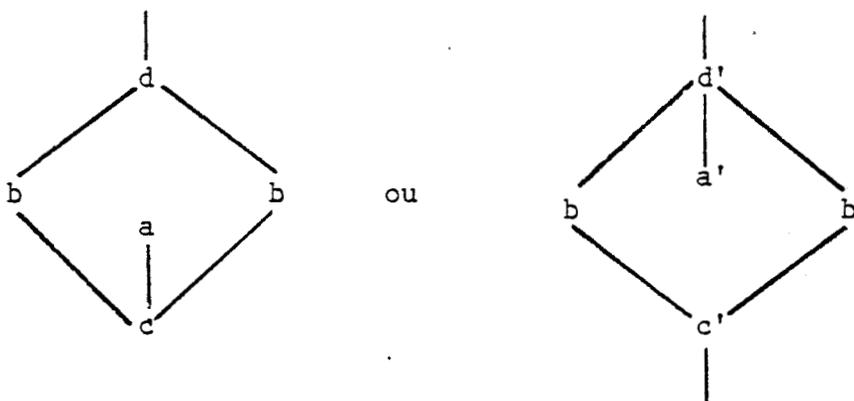
* Le nombre d'états "contrôlant" un dag lors d'une reconnaissance est directement lié à la coarité et à l'arité de ce dag. Ainsi la reconnaissance d'un dag tel que $\overset{|}{a}$, où a est de coarité nulle et $\overset{|}{a}$ est de arité nulle, ne serait subordonnée à aucun état à l'extérieur et de tels dags pourraient proliférer, sans aucun contrôle, dans n'importe quel dag reconnu par ailleurs. Ainsi un automate reconnaissant séparément.:



reconnaîtrait aussi les dags



* Il est vrai que les dags de la forme

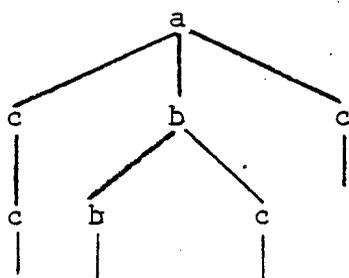


auraient pu être reconnus, en rompant avec la symétrie, en admettant dans l'alphabet la coarité nulle ou (exclusif) l'arité nulle. Mais nous avons exclu ce cas puisqu'il ne change pas le fond de notre démarche, mais compliquerait formalismes et preuves.

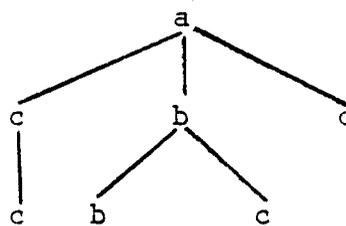
Remarque : L'exclusion des lettres de coarité ou de arité nulle paraît dissocier les arbres et les dags : un arbre est un élément de $T(\Sigma)_0^1$, c'est-à-dire un élément ayant une coarité égale à 1 et une arité nulle. Mais cette dissociation n'est qu'une différence de formalisme [II.1.5.], et il est immédiat d'identifier un dag planaire construit sur un alphabet d'arbres (lettres de coarité égale à 1) à un arbre, et vice-versa.

Exemple :

dag planaire

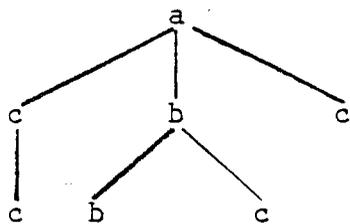


arbre

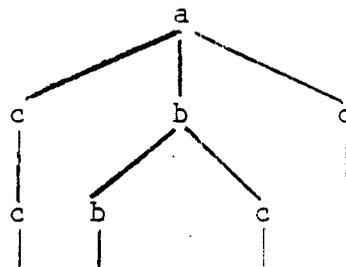


et

arbre



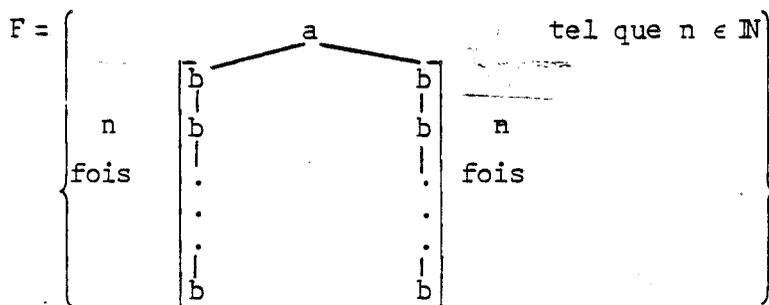
dag planaire



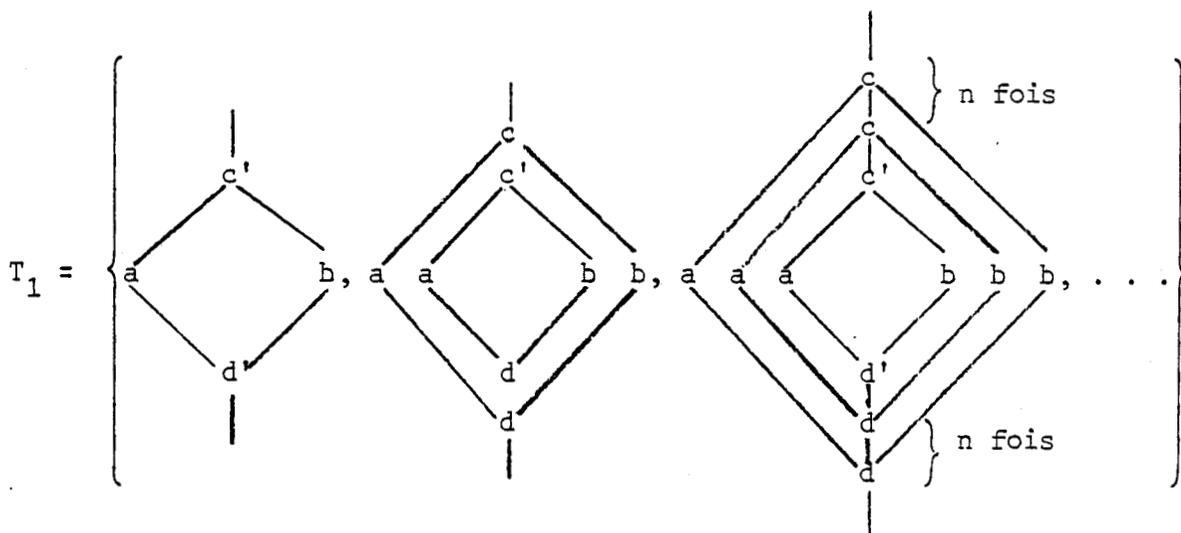
Ainsi, l'identification est faite à l'arité près des noeuds des feuillages.

Après la justification de nos choix dans la définition des automates d'états finis de dags planaires, abordons maintenant les propriétés charnières des taillis reconnus par de tels automates, i.e. les taillis RECONNAISSABLES. Rappelons auparavant que la dualité parfaite entre les automates descendants et les automates ascendants nous assure que toutes propriétés vraies dans le cas descendant se retrouve (symétriquement) dans le cas ascendant, et vice-versa :

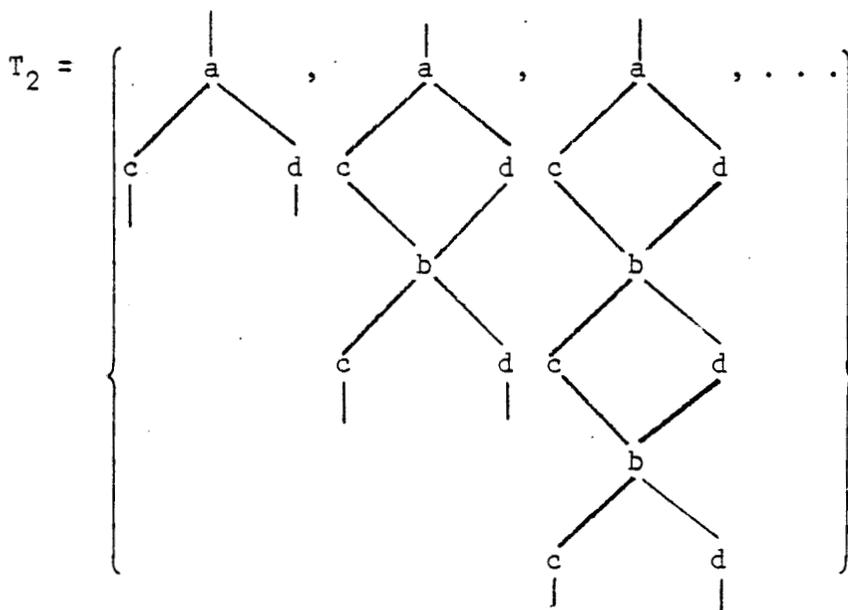
1. Comme pour les automates d'arbres, un automate de dags planaires ne pourra "établir" qu'un "comptage borné". Par exemple, la forêt (donc le taillis) :



n'est pas une forêt reconnaissable, ni, considérée en tant que taillis, un taillis reconnaissable. Cependant un automate de dags planaires permet un "comptage synchronisé". Par exemple, le taillis :



est un taillis reconnaissable. Dans le même ordre d'idée, le taillis



est un taillis reconnaissable. Ce qui nous amène à remarquer immédiatement que les taillis reconnaissables ne sont pas clos par homomorphisme de magmoïde puisque $F = h(T_2)$ où h est l'identité sauf pour $h(b) = e \otimes e$.

2. Il convient de signaler un résultat important [II.1.4.] : à tout automate de dags planaires, on peut associer un automate $M' = \langle \Sigma, Q', I', F', R' \rangle$ tel que

$$T(M) = T(M')$$

$$I' = Q'_0 \text{ où } Q'_0 \subseteq Q$$

En d'autres termes, si $I' = Q'_0$, l'appartenance à I' du mot initial d'états se réduit à l'appartenance de chacun des états de ce mot, à l'ensemble Q'_0 , ce qui peut se faire au niveau de l'utilisation d'une règle état par état. Ainsi l'expression du contrôle haut d'une reconnaissance se "résume" à l'appartenance d'un état à un ensemble d'états, comme dans les mots et les arbres.

Par dualité [II.1.3.], on peut construire $M'' = \langle \Sigma, Q'', I'', F'', R'' \rangle$ tel que

$$T(M) = T(M'')$$

$$F'' = Q''_f \text{ où } Q''_f \subseteq Q$$

et dans ce cas le contrôle bas se "résume" à l'appartenance d'un état à un ensemble d'états. Mais c'est seulement dans le cas où M reconnaît un taillis de dags connexes que l'on peut assurer de pouvoir "résumer" à la fois [V.2.4.] :

- le contrôle haut à l'appartenance d'un état à un ensemble d'états,
- le contrôle bas à l'appartenance d'un état à un ensemble d'états.

Remarque : La construction des automates M' ou M'' à partir de l'automate M est obtenue par démultiplication de l'ensemble d'états et l'introduction d'un non-déterminisme considérable [II.1.4.].

3. L'étude du déterminisme dans les automates de dags planaires permet de situer (en partie) la "complexité" des calculs (les reconnaissances) de ces automates. En effet, nous montrons [II.1.6.3.] que le non-déterminisme n'est pas réductible, i.e. on ne peut pas toujours construire un automate déterministe. En conséquence :

- les reconnaissances des dags planaires sont d'une complexité beaucoup plus grande que celle des automates de mots ou d'arbres,
- cette non-réduction du non-déterminisme nous ferme la voie classique (Hoscropft, Ullman [29]) de démonstration de clôture par complémentaire de REC. La clôture de REC par complémentaire reste un problème ouvert [II.2.9.].

Remarque : Le lecteur trouvera en [II.1.6.1.1. et II.1.6.1.2.] deux définitions (proches) d'automate déterministe. La première guidée par notre souci de généralisation des théories existantes (langages et forêts) nous conduit à définir :

les automates fortement déterministes.

La seconde guidée par notre souci de prendre en compte pleinement la notion de déterminisme nous conduit à définir :

les automates faiblement déterministes.

Mais le non-déterminisme n'étant pas réductible ni à l'une [II.1.6.3.] ni à l'autre [II.1.6.4.] définition, le distingo proposé n'est pas un problème de fond.

4. Enfin, l'étude des propriétés de clôture dans REC confirme l'importance des deux aspects dégagés jusqu'à présent :

- la CONNEXITE,
- le NON-DETERMINISME (et donc la complexité).

REC est clos [II.2.1. à II.2.9.] par :

- union, intersection, produit tensoriel, étoile du produit tensoriel, produit de composition, homomorphisme inverse et homomorphisme connexe.

REC n'est pas clos [II.2.6. et II.2.7.] par :

- étoile du produit de composition, homomorphisme de magmaïde.

Rappelons que la clôture de REC par complément reste un problème ouvert [II.2.10.].

B.4.3. La rationalité dans les dags planaires

Comme dans les mots, et les arbres, nous allons considérer certaines familles de dags obtenues par combinaison d'opérations ensemblistes. Cette démarche vise naturellement à faire coïncider dans les dags les notions de reconnaissabilité et de rationalité. Plusieurs difficultés s'ajoutent à celles déjà observées dans les arbres. Dans les expressions rationnelles d'arbres, on est astreint à un marquage de certaines arités afin de les différencier les unes des autres. Dans les dags, il sera nécessaire de particulariser arité et coarité, ce qui nous oblige à travailler avec des lettres bitypées. Ceci entraîne la définition d'un nouveau cadre d'étude : le magmaïde typé libre $\hat{DP}(\Sigma, T)$. Les opérateurs ensemblistes choisis seront, à l'image de ceux retenus pour les expressions rationnelles de mots et d'arbres, l'union et l'extension aux ensembles des opérations constructrices des objets ainsi que leurs opérateurs itérés. Hélas, on ne peut adopter cette façon standard de faire pour l'opérateur produit tensoriel. En effet, une extension pure et simple de \otimes aux ensembles introduirait la possibilité d'exprimer une mise en parallèle synchrone non bornée. Ainsi $(a \otimes b)^*$ décrirait le taillis $\{a^n \otimes b^n / n \in \mathbb{N}\}$ qui n'est pas reconnaissable. On crée donc un nouvel opérateur \otimes , dérivé du produit tensoriel, et son opérateur itératif associé noté \otimes (placé en exposant). Cet opérateur \otimes s'interprète comme une mise en parallèle asynchrone, c'est-à-dire que la présence simultanée des termes l'un à côté de l'autre n'est plus obligatoire. De manière plus formelle, l'interprétation de $a \otimes b$ est l'ensemble suivant de $\hat{DP}(\Sigma, T)$:

$$\{ \underbrace{e \otimes \dots \otimes e}_n \otimes b / n \in \mathbb{N} \} \cup \{ a \otimes e \otimes \dots \otimes e / \underbrace{p}_p \in \mathbb{N} \} \cup \{ a \otimes b \}.$$

Un dernier obstacle reste à surmonter : comment traduire dans les expressions rationnelles de dags les contrôles haut et bas qui interviennent dans les automates d'états finis. On doit se résoudre à filtrer les éléments décrits par l'expression rationnelle en leur imposant un type d'entrée et un type de sortie à valoir dans des langages rationnels de T^* (si T est l'ensemble des types utilisés). La notion d'expression rationnelle de dag et leur interprétation en terme de taillis sont rigoureusement définies en [VI.4.1.2.].

Il est évident que tout dag dans lequel figure l'opérateur produit tensoriel peut être décrit par une expression rationnelle dans laquelle on fera appel, pour simuler l'opérateur \otimes , à l'opérateur \otimes dont l'action sera dirigée par un typage approprié des lettres.

Exemple : L'interprétation de l'expression rationnelle :

$$(q_0 \ q_0; a_{q_1}^{q_0} \otimes b_{q_1}^{q_0}, q_1 \ q_1)$$

recouvre l'ensemble des dags dont le type d'entrée est $q_0 \ q_0$, le type de sortie $q_1 \ q_1$ et qui appartiennent à l'interprétation de

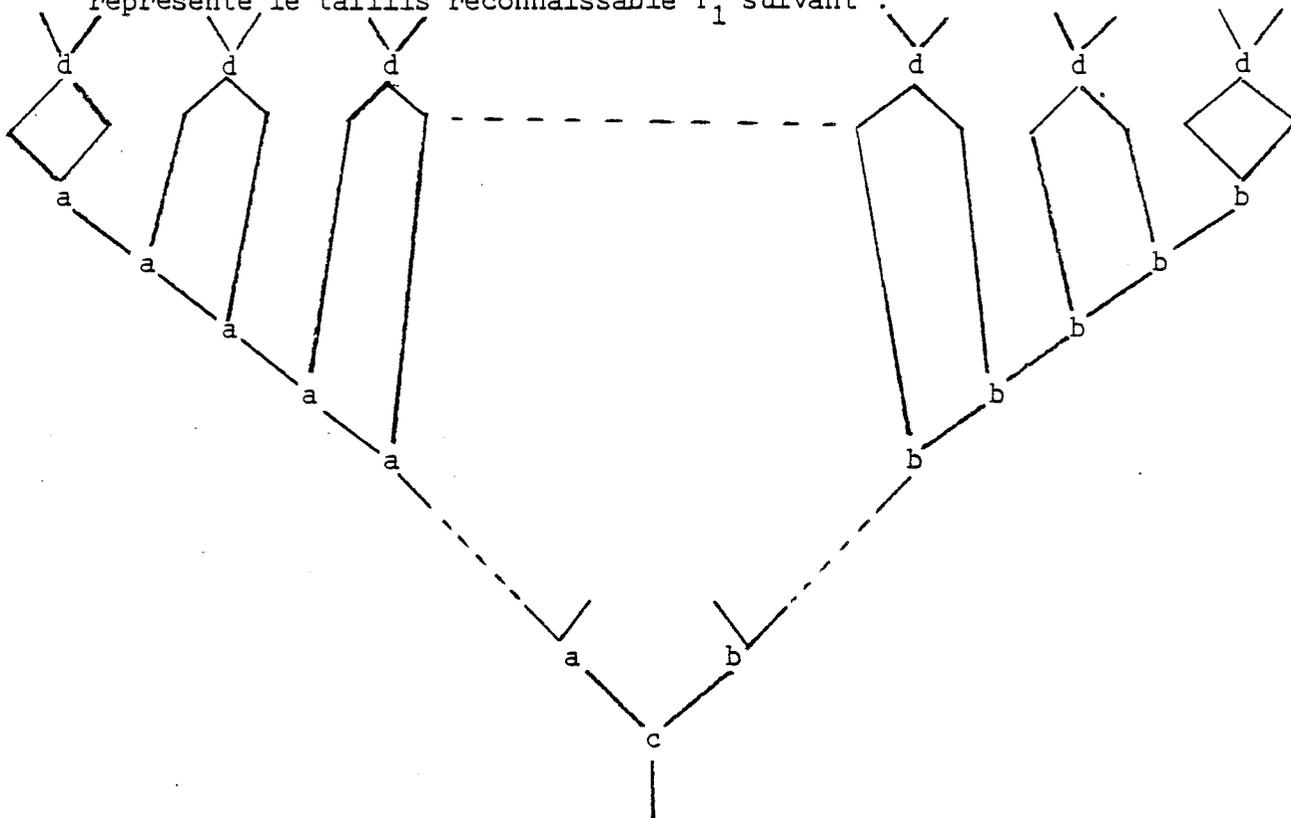
$$a_{q_1}^{q_0} \otimes b_{q_1}^{q_0} : \text{ soit donc } a_{q_1}^{q_0} \otimes b_{q_1}^{q_0}, \text{ ce qui donne par détypage } a \otimes b.$$

La notion de rationalité, telle que nous venons de la définir, remplit bien sa mission qui était de se superposer à la notion de reconnaissabilité. On peut trivialement exhiber une expression rationnelle exprimant un taillis reconnaissable à partir de la connaissance de l'automate : il suffit d'effectuer un typage des lettres reproduisant exactement les règles de l'automate. On peut aussi s'ingénier à trouver des expressions rationnelles équivalentes fournissant une description plus riche (en appauvrissant en contre-partie les types).

Exemple : L'expression rationnelle :

$$\left(t'^*, \left[\begin{array}{c} t' \quad t' \\ \quad \diagdown \quad / \\ \quad \quad d \\ \quad / \quad \diagdown \\ t \quad t \end{array} \right]^{\circledast} \cdot \left[\begin{array}{c} t \quad t \quad t \quad t \\ \quad \diagdown \quad / \quad \diagdown \quad / \\ \quad \quad a \quad \quad b \\ \quad / \quad \diagdown \quad / \quad \diagdown \\ t \quad t \quad t \quad t \end{array} \right]^* \cdot \left[\begin{array}{c} t \quad t \\ \quad \diagdown \quad / \\ \quad \quad c \\ \quad / \\ t \end{array} \right], t'^* \right)$$

représente le taillis reconnaissable T_1 suivant :



Si la preuve de l'inclusion de RAT dans REC est une simple formalité, la réciproque est d'une toute autre complexité technique.

Difficulté 1 :

Pour des expressions rationnelles présentant de nombreux opérateurs autres que l'opérateur d'union (+), on assiste, à une prolifération de règles entraînant un non-déterminisme chronique, lorsque l'on construit l'automate équivalent.

Par exemple l'ensemble des dags qui se terminent par un dag de la famille T_1 (exemple précédent) se décrit rapidement par l'expression rationnelle :

$$\left(t'^*, \left[\left(\begin{array}{cc} t' & t' \\ & d \\ b' & t' \end{array} + \begin{array}{cc} t' & t' \\ & a \\ & t' \end{array} + \begin{array}{cc} t' & t' \\ & b \\ & t' \end{array} + \begin{array}{cc} t' & t' \\ & c \\ & t' \end{array} \right]^{\circledast} \right) \cdot \underbrace{\left[\begin{array}{cc} t' & t' \\ & d \\ t & t \end{array} \right]^{\circledast} \cdot \left[\begin{array}{cc} t & t \\ & a \\ & t \end{array} \right]^{\circledast} \cdot \left[\begin{array}{cc} t & t \\ & b \\ & t \end{array} \right]^{\circledast} \cdot \left[\begin{array}{cc} t & t \\ & c \\ & t \end{array} \right]^{\circledast}, t'^* }_{\text{dag} \in T_1}$$

L'automate reconnaissant de tels dags est plus compliqué à trouver. En effet, pour reconnaître ces dags, il faudrait prendre des "paris" sur la fonction tenue par les lettres.

Difficulté 2 :

Une nouvelle difficulté surgit lorsque l'on cherche à montrer que la mise à l'étoile (du produit de composition) d'une expression simple conserve son caractère reconnaissable.

Cette difficulté se concentre autour du passage d'un dag de l'interprétation de l'expression au suivant. On est contraint à ce niveau d'intégrer en cours de reconnaissance la vérification du contrôle bas pour l'un et du contrôle haut pour l'autre. La solution réside dans une analyse fine du comportement de l'association des opérateurs * et \circledast . On s'aperçoit, en particulier, lorsque l'on passe à l'étoile, que les emplacements relatifs des composantes connexes des dags assemblés les unes aux autres ne dépendent plus que des correspondances de types ; excepté pour celles qui occupent les bords de la structure.

Exemple : Dans l'interprétation de l'expression $(a \odot b \odot c \odot d \odot f)^*$ on trouve les dags suivants :

$a \odot b \odot d \odot c \odot f$

$a \odot d \odot c \odot b \odot f$

$a \odot c \odot d \odot b \odot f \dots \text{etc.} \dots$

D'autre part, on sait que toute composante connexe reconnaissable peut l'être en utilisant un automate pourvu simplement d'états initiaux et finaux.

De ces deux observations, on déduit que le passage d'un dag de l'expression simple au suivant peut être négocié de façon séparée sur chacune des composantes connexes, et ceci par une simple substitution d'un final par un état initial.

Cette preuve est longuement détaillée et commentée par la partie VII de cette Thèse.

B.4.4. Les grammaires régulières de dags planaires

Evoquons, pour achever le survol de ce travail, un troisième angle d'approche des taillis étudiés : l'aspect génératif. Les grammaires linéaires à droite et régulières de dags exposées dans la partie III présentent quelques particularités par rapport aux notions équivalentes dans les mots ou les arbres. Les parties gauches de règles peuvent éventuellement être un produit tensoriel de plusieurs variables et leur partie droite doit être connexe. Cette dernière contrainte interdit toute génération parallèle et synchrone de termes dissociés sur une hauteur arbitrairement grande.

On établit l'égalité entre l'ensemble des taillis réguliers (REG) et l'ensemble des taillis reconnaissable (REC) par un raisonnement calqué sur la démonstration de $REC = REG$ dans les mots. Il suffit de transformer les règles de grammaire en transition d'automate par changement du statut des variables en états (ou vice-versa).

L'emploi de telles grammaires se révèle intéressant notamment pour démontrer que le taillis des dags de dérivation d'une grammaire à structure de phrases est régulier (donc rationnel - donc reconnaissable). Une présentation sommaire des dags de dérivation a été donné lors de l'exemple B.2.2. de cette introduction ; la partie IV de cette thèse leur est entièrement consacrée. On découvrira en particulier, que la reconnaissabilité des taillis de dérivation a comme conséquence l'indécidabilité de l'égalité de deux taillis reconnaissables.

B.4.5. Axe de développement de notre travail

Le formalisme adopté pour la manipulation des dags permet d'établir de nombreux résultats qui ressemblent dans leur esprit à ceux obtenus sur les mots et les arbres. D'autre part, certains algorithmes, notamment les automates d'états finis de dags, ont été implémentés et testés sur machine (Caridroit [42]). Tout cela nous conforte dans l'idée que le magmaïde est l'un des outils de prédilection dans les dags. On peut alors envisager de persévérer dans cette voie d'étude suivant deux directions académiques :

1. Le modèle de la théorie des langages

La théorie des langages nous fournit quantités de notions que l'on peut transposer dans les dags. Soulevons quelques questions classiques à titre d'exemples :

- Le problème ouvert du complémentaire d'un taillis reconnaissable.
- Les transducteurs et bimorphismes.
- Les DAGS infinis.
- Les grammaires algébriques de DAGS.

Et l'on pourrait encore allonger la liste. Une telle démarche pourrait se justifier de part les résultats qu'elle dévoilerait. Mais elle permettrait également de mettre en évidence des phénomènes profonds à travers diverses structures. Ainsi, les bimorphismes se composent dans les mots (Nivat [38]); dans les arbres, il ne se composent pas mais toute composition de n bimorphismes se ramène à la composition de 2 (Dauchet [12]). Il serait, par conséquent, intéressant d'examiner la puissance de la composition des bimorphismes dans les dags.

2. Utilisation des dags dans un cadre interprété

L'emploi de dags semble bien adapté à l'analyse de la complexité des programmes comme nous avons pu le constater sur des exemples de cette introduction. On peut remarquer également que les programmes pilotés par les données s'exprimeraient

bien à l'aide de dags. De même, des schémas de programmes itératifs parallèles incluant des systèmes de rendez-vous (FORK... JOIN) généreraient des traces de calculs convenablement représentées par des dags. Et, si l'ensemble des traces de calculs d'un schéma de programme itératif est une forêt rationnelle (Cousineau [45]), l'ensemble des traces de calculs d'un schéma de programme itératif parallèle pourrait être vu comme rationnel au sens des taillis de dags. L'apport structurel des dags sur les arbres élargit considérablement la gamme des interprétations possibles et, par voie de conséquence, atteste des nombreuses applications qui pourraient tirer parti d'un "bon" formalisme sur ce type d'objets.

CHAPITRE 0

PRÉLIMINAIRE SUR LE CADRE
FORMEL DE L'ETUDE : LES MAGMOÏDES

PLAN DU CHAPITRE 0

0.1. - DEFINITION DE MAGMOÏDES

0.2. - LES MORPHISMES DE MAGMOÏDES

0.3. - LES MAGMOÏDES LIBRES

0.1. - DEFINITION DES MAGMOIDES

Un magmoïde est un quintuplet $\langle M, \cdot, \otimes, e, e_0 \rangle$, qui sera noté M selon l'usage consistant à identifier une structure algébrique et son support ; M est ici un ensemble, \cdot une opération binaire partielle sur M , \otimes une opération binaire sur M , e et e_0 deux éléments distingués de M . M vérifie les axiomes suivants :

A1. $\forall p, q \in \mathbb{N}$, il existe une partie M_q^D de M appelée fibre p - q de M . Les fibres sont deux à deux disjointes et M est la réunion de toutes ses fibres.

M est donc un ensemble bigradué. Un élément de M_q^D sera dit de coarité p et de arité q .

A2. $\forall p, q, p', q' \in \mathbb{N} \quad \forall m \in M_q^D \quad \forall m' \in M_{q'}^{D'}$

$m \cdot m'$ est défini ssi $q = p'$. On a alors $m \cdot m' \in M_{q'}^{D'}$.

Cette opération \cdot est dénommée produit de composition.

A'2. Le produit de composition est associatif.

A3. $\forall p, q, p', q' \in \mathbb{N} \quad \forall m \in M_q^D \quad \forall m' \in M_{q'}^{D'} \quad m \otimes m' \in M_{q+q'}^{D+p'}$.

Cette opération \otimes est dénommée produit tensoriel.

A'3. Le produit tensoriel est associatif.

A4. $\forall m_1, m_2, m'_1, m'_2 \in M$ si $(m_1 \cdot m_2) \otimes (m'_1 \cdot m'_2)$ est défini alors
 $(m_1 \cdot m_2) \otimes (m'_1 \cdot m'_2) = (m_1 \otimes m'_1) \cdot (m_2 \otimes m'_2)$.

A5. $e \in M_1^1, e_0 \in M_0^0$. En posant, pour $p > 0$ $e_p = \underbrace{e \otimes \dots \otimes e}_{p \text{ fois}}$.

On a : $\forall m \in M_q^D \quad e_p \cdot m = m$ et $m \cdot e_q = m$

et $\forall m \in M \quad e_0 \otimes m = m \otimes e_0 = m$

Exemple :

Soit l'ensemble $R(E) = \bigcup_{i,j} R_j^i(E)$

, avec $R_j^i(E) = \{\text{ensemble des relations de } E^i \text{ dans } E^j\}$, muni des lois suivantes :

- • La composition des relations.

$\forall r \in R_j^i(E) \forall r' \in R_k^j(E)$, $r \cdot r'$ est une relation de $R_k^i(E)$ définie par :

$\forall x \in E^i \forall z \in E^k$ (x, z) appartient au graphe de $r \cdot r'$ ssi

$\exists y \in E^j$ tel que (x, y) appartient au graphe de r et (y, z) appartient au graphe de r' .

- Le produit tensoriel des relations.

$\forall r \in R_q^p(E) \forall r' \in R_s^r(E)$ $r \otimes r'$ est une relation de $R_{q+s}^{p+r}(E)$ définie par :

$\forall (x_1, x_2, \dots, x_p, x_{p+1}, \dots, x_{p+r}) \in E^{p+r} \forall (y_1, y_2, \dots, y_q, y_{q+1}, \dots, y_{q+s}) \in E^{q+s}$

$((x_1, \dots, x_{p+r}), (y_1, \dots, y_{q+s}))$ appartient au graphe de $r \otimes r'$ ssi

$((x_1, \dots, x_p), (y_1, \dots, y_q))$ appartient au graphe de r et

$((x_{p+1}, \dots, x_{p+r}), (y_{q+1}, y_{q+s}))$ appartient au graphe de r' .

On notera Id^i la relation $\in R_1^i(E)$ dont le graphe est l'ensemble des couples (x, x) , $x \in E^i$, et I_0 la seule relation définie de l'ensemble vide sur lui-même.

Alors $\langle R(E), \otimes, \cdot, \text{Id}^1, I_0 \rangle$ est un magmaïde.

0.2. - LES MORPHISMES DE MAGMOIDES

Soient $\langle M, \cdot, \otimes, e, e_0 \rangle$ et $\langle M', \cdot, \otimes, e', e'_0 \rangle$ deux magmoïdes.

Une application $\underline{\Phi}$ de M dans M' est un morphisme de magmoïde ssi :

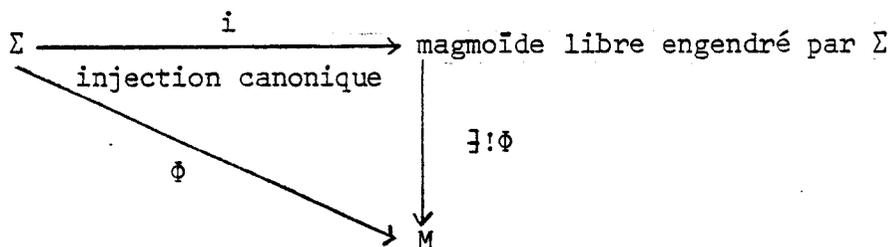
- Φ respecte la graduation : $\forall m \in M_q^D \quad \Phi(m) \in M_q^{D'}$
- Φ préserve les éléments neutres : $\Phi(e) = e'$ et $\Phi(e_0) = e'_0$
- Φ est compatible avec les opérations \cdot et \otimes :

$$\begin{aligned} \forall m \in M_q^D, m' \in M_r^D \quad & \Phi(m \cdot m') = \Phi(m) \cdot \Phi(m') \\ \forall m \in M_q^D, m' \in M_s^D \quad & \Phi(m \otimes m') = \Phi(m) \otimes \Phi(m'). \end{aligned}$$

0.3. - LES MAGMOIDES LIBRES

On peut également définir la notion de magmoïde libre engendré par un alphabet bigradué donné Σ .

Le magmoïde libre engendré par Σ sera tel que : quel que soit le magmoïde M , quelle que soit l'application Φ de Σ dans M qui respecte la graduation, il existe un et un seul morphisme de magmoïde $\hat{\Phi}$ du magmoïde libre dans M tel que sa restriction à Σ considéré comme partie du magmoïde libre soit Φ . Autrement dit $\hat{\Phi}$ s'étend de façon unique en un morphisme du magmoïde libre dans M .



Remarque :

Le magmoïde exposé en I.1. à titre d'exemple n'est pas un magmoïde libre.

En effet, considérons les relations suivantes :

$$R_1 \subseteq \mathbb{N}^3 \times \mathbb{N}^2 \text{ telle que } (x, y, z) R_1 (x, y)$$

$$R_2 \subseteq \mathbb{N}^2 \times \mathbb{N} \text{ telle que } (x, y) R_2 y$$

$$R_3 \subseteq \mathbb{N}^3 \times \mathbb{N}^2 \text{ telle que } (x, y, z) R_3 (y, z)$$

$$R_4 \subseteq \mathbb{N}^2 \times \mathbb{N} \text{ telle que } (x, y) R_4 x$$

Il est immédiat que $R_1 \cdot R_2 = R_3 \cdot R_4$.

Imaginons l'application g qui laisse invariant R_1, R_2 et R_3 et transforme R_4 en R_2 et étendons-la à la manière d'un morphisme ; on aurait alors :

$$g(R_1 \cdot R_2) = g(R_1) \cdot g(R_2) = R_1 \cdot R_2$$

$$g(R_3 \cdot R_4) = g(R_3) \cdot g(R_4) = R_3 \cdot R_2$$

$$\text{et } R_1 \cdot R_2 \neq R_3 \cdot R_2$$

$\mathcal{R}(E)$ n'est donc pas un magmoïde libre, en d'autres termes l'égalité dans $\mathcal{R}(E)$ ne dépend pas uniquement des propriétés de sa structure de magmoïde, mais d'autres facteurs qui rendent compte d'une interprétation plus large de ses éléments constitutifs de $\mathcal{R}(E)$.

CHAPITRE 1

LE MAGMOÏDE DES DAGS PLANAIRES

PLAN DU CHAPITRE I

I.1. - LE MAGMOÏDE $\tilde{DP}(\Sigma)$

I.2. - REPRESENTATION GRAPHIQUE DES ELEMENTS DE $\tilde{DP}(\Sigma)$

I.3. - EXISTENCE D'UN REPRESENTANT CANONIQUE DES ELEMENTS DE $\tilde{DP}(\Sigma)$

I.4. - FORME NORMALE DUALE D'UN ELEMENT DE $\tilde{DP}(\Sigma)$

I.5. - FEUILLAGE D'UN DAG PLANAIRE

I.6. - EXTENSIONS DES OPERATIONS DU MAGMOÏDE, AUX TAILLIS

I.1. - LE MAGMOÏDE DES DAGS PLANAIRES : $\tilde{DP}(\Sigma)$

On dira que Σ ensemble fini est un alphabet bigradué ssi il existe deux applications (appelées arité et coarité) de Σ dans \mathbb{N} .

On appelle alors Σ_q^p le sous-ensemble de Σ des lettres de coarité p et de arité q .

Nous allons décrire le langage $L(\Sigma)$ des expressions bien formées construites à partir de Σ , de deux opérateurs binaires \cdot et \otimes (appelés produit de composition et produit tensoriel) et de deux nouveaux symboles e et e_0 .

$$L(\Sigma) = \bigcup_{p,q \in \mathbb{N}} L(\Sigma)_q^p$$

où les $L(\Sigma)_q^p$ sont définis de façon récursive par :

$$- \Sigma_q^p \subseteq L(\Sigma)_q^p$$

$$- e_0 \in L(\Sigma)_0^0 ; e \in L(\Sigma)_1^1$$

$$- \forall w \in L(\Sigma)_q^p, \forall w' \in L(\Sigma)_q^{p'} \text{ alors } (w \cdot w') \in L(\Sigma)_q^p$$

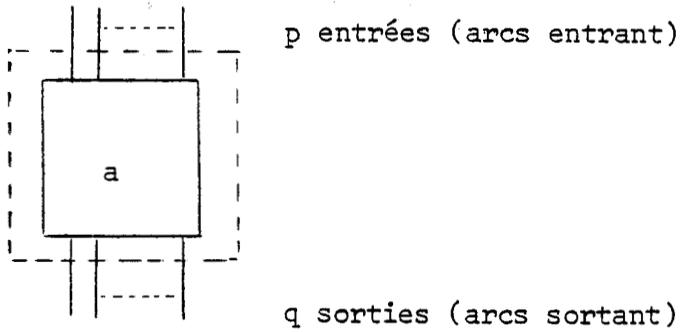
$$- \forall w \in L(\Sigma)_q^p, \forall w' \in L(\Sigma)_q^{p'} \text{ alors } (w \otimes w') \in L(\Sigma)_{q+q'}^{p+p'}$$

On appelle $\tilde{DP}(\Sigma)$ le magmoïde libre engendré par Σ , c'est-à-dire l'ensemble $L(\Sigma)$ quotienté par les relations d'équivalence définies par les axiomes du magmoïde.

I.2. - REPRESENTATION GRAPHIQUE DES ELEMENTS DE $\tilde{DP}(\Sigma)$

Nous allons montrer qu'à tout élément de $L(\Sigma)$, on peut associer un graphe planaire orienté dont les noeuds seront étiquetés par les lettres de Σ ; le nombre d'arcs arrivant sur un noeud, et le nombre d'arcs issus d'un noeud seront respectivement égaux à la coarité et arité de son étiquette.

. A une lettre $a \in \Sigma_q^p$, on associera le graphe

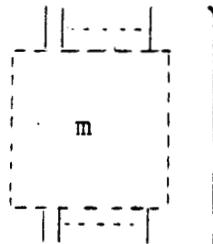


. e sera représenté par un morceau d'arc de longueur quelconque.

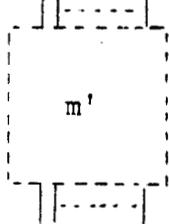
. e_0 n'a pas de représentation visible.

. $\forall m \in L(\Sigma)_q^p \quad \forall m' \in L(\Sigma)_s^r \quad (m \otimes m') \in L(\Sigma)_{q+s}^{p+r}$

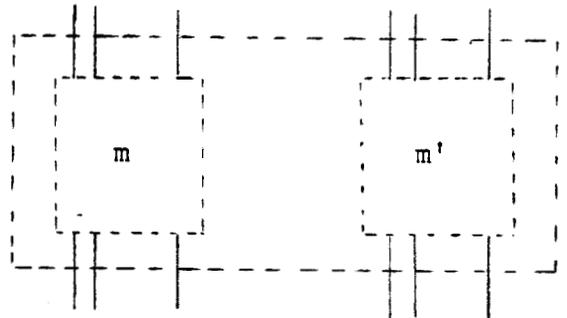
si m se dessine



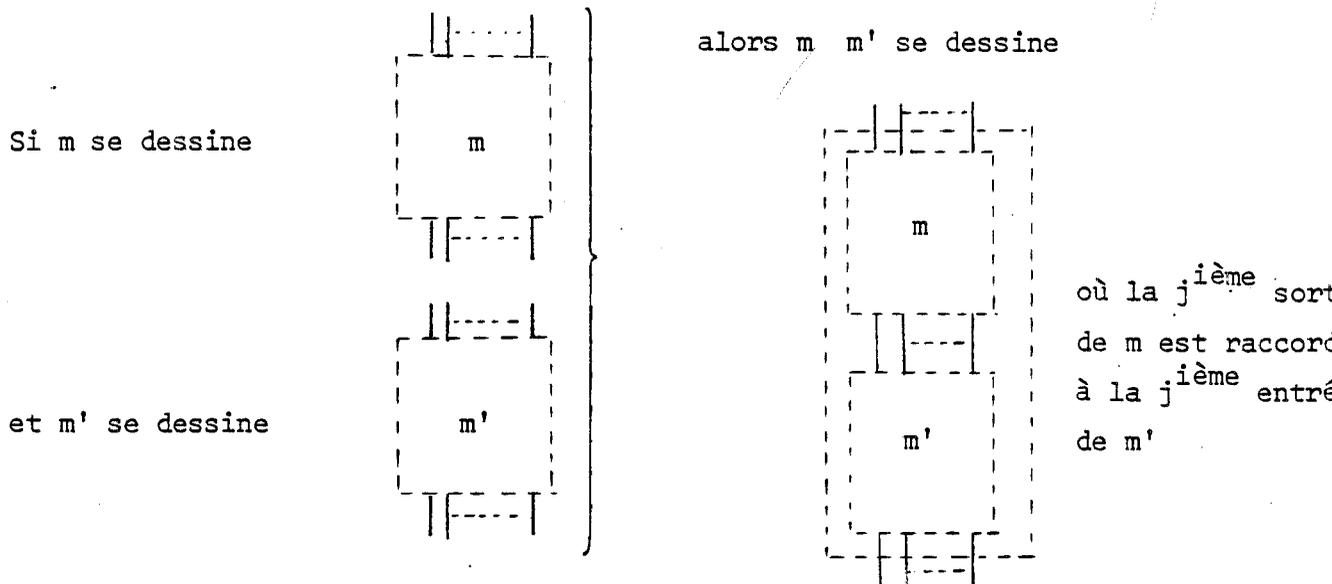
et m' se dessine



alors $(m \otimes m')$ se dessine :



. $\forall m \in L(\Sigma)_q^p \quad \forall m' \in L(\Sigma)_r^q \quad (m \cdot m') \in L(\Sigma)_r^p$



Pour obtenir une représentation en terme de graphe des éléments du magmoïde $\overset{\sim}{DP}(\Sigma)$, nous allons examiner comment s'interprètent les axiomes du magmoïdes sur les graphes obtenus à partir des éléments de $L(\Sigma)$.

La notation linéaire des éléments de $\overset{\sim}{DP}(\Sigma)$ (c'est-à-dire les expressions de $L(\Sigma)$) permet de rendre compte de l'associativité des lois \otimes et \cdot .

En effet, on peut convenir de noter $(m \otimes m' \otimes m'')$ l'élément du magmoïde dont les écritures possibles sont $((m \otimes m') \otimes m'')$ et $(m \otimes (m' \otimes m''))$. La même convention peut-être adoptée pour $(m \cdot m' \cdot m'')$.

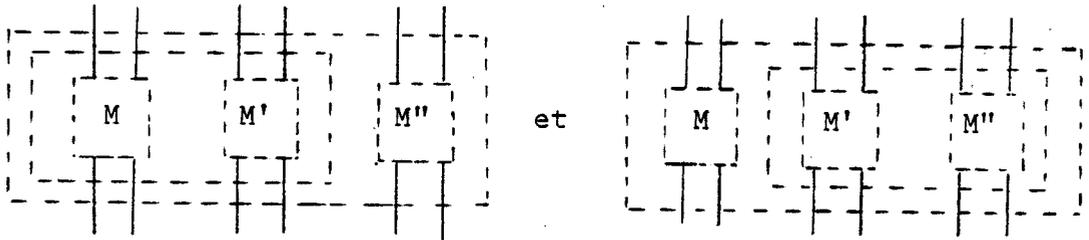
La notation linéaire permet également de tenir compte de la propriété d'élément neutre de e_1 et e_0 .

Malheureusement, on ne peut rendre compte de l'axiome A4 dans ce style d'écriture. Dans $(m_1 \cdot m_2) \otimes (m'_1 \cdot m'_2)$ et $(m_1 \otimes m'_1) \cdot (m_2 \otimes m'_2)$, les places et les nombres d'occurrences des symboles \otimes , et \cdot sont changés.

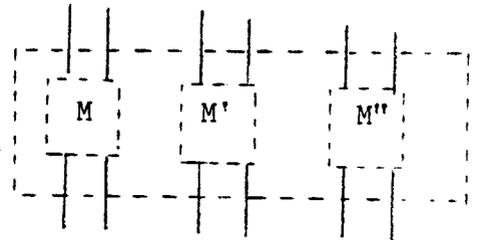
Notre notation en dimension 2, va nous permettre de résoudre le problème.

Transfert de l'axiomatique du magmoïde dans les représentations graphiques

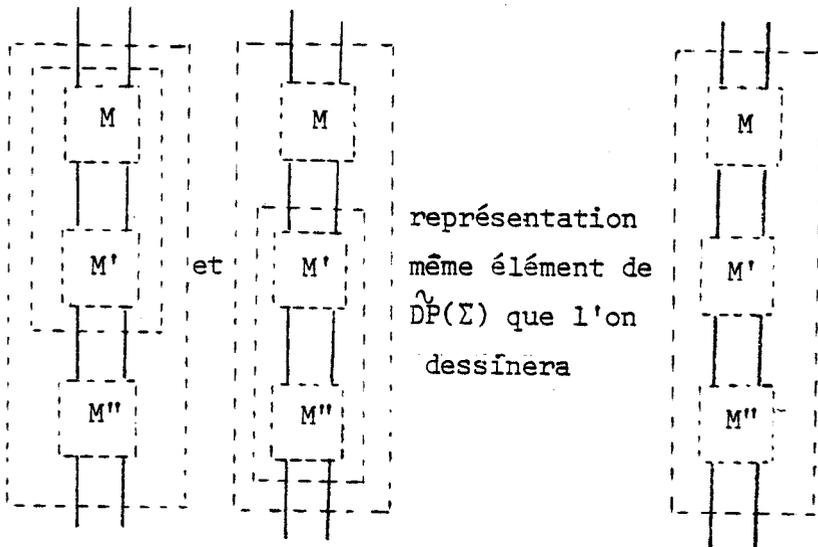
a) L'associativité de \otimes :



représentent un même élément
de $\hat{DP}(\Sigma)$ qui se dessinera :



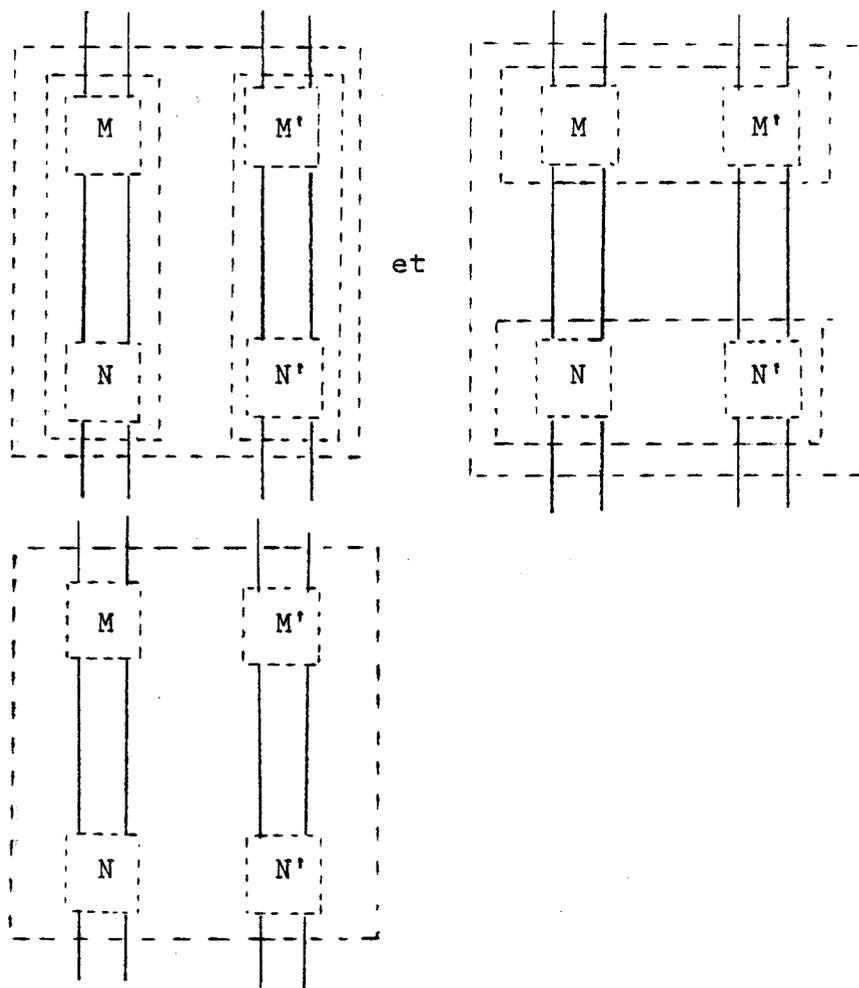
b) L'associativité de \cdot :



c) L'axiome A4 :

$\forall m, m', n, n' \in \hat{DP}(\Sigma)$ si $(m \cdot n) \otimes (m' \cdot n')$ est défini alors

$$(m \cdot n) \otimes (m' \cdot n') = (m \otimes m') \cdot (n \otimes n').$$



représentent un même
élément de $\hat{DP}(\Sigma)$ que
l'on dessinera

d) La neutralité de e_1 et e_0 :

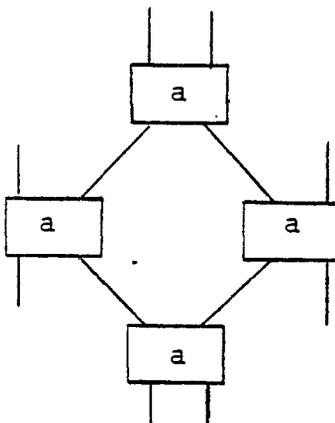
e_0 n'a pas d'existence dans le dessin.

e_1 est matérialisé par un bout d'arc, donc n'intervient pas dans la structure propre du dessin.

On constate, finalement, que la représentation sous forme de dag d'un élément du magmoïde ne comporte plus aucun encadré en pointillé. En effet, ceux-ci disparaissent en appliquant les simplifications de dessin issues des axiomes du magmoïde. Cet encadré était la transposition en dimension 2 des parenthèses de la notation linéaire. On obtient ainsi une écriture unique en dimension 2 (à la longueur des arcs près) d'un élément de $\hat{DP}(\Sigma)$.

Exemple :

La classe d'équivalence de $(e \otimes a \otimes e) \cdot (a \otimes a) \cdot (e \otimes a \otimes e)$ avec $a \in \Sigma_2^2$ pourrait être représentée par le dessin :



Conclusion : Il existe une bijection entre l'ensemble des dags planaires et le magmaïde $\hat{DP}(\Sigma)$. On confrontera ces deux ensembles dans la suite de notre propos.

1.3. - EXISTENCE D'UN REPRESENTANT CANONIQUE DES ELEMENTS DE $\hat{DP}(\Sigma)$

On vient d'exhiber une représentation unique d'un élément de $\hat{DP}(\Sigma)$ en dimension 2, nous allons montrer l'existence d'un représentant canonique calculable pour chaque classe d'équivalence de $L(\Sigma)$: ce représentant sera appelé la forme normale de l'élément du magmaïde associé à la classe d'équivalence de $L(\Sigma)$.

Cette forme normale sera obtenue par application d'un système de réécriture noetherien et confluent, système dont les règles ne sont qu'une retranscription des axiomes sous forme de règles de réécriture orientées.

Remarques préliminaires :

Etant donné que l'associativité des lois \cdot et \otimes peut être exprimée dans la notation linéaire des éléments du magmaïde, nous travaillerons non pas sur des expressions de $L(\Sigma)$, mais sur des expressions dont on aurait déjà réduit au maximum le nombre de parenthèses compte tenue de l'associativité de \cdot et \otimes , et dont on aurait fait disparaître toutes les occurrences de e_0 , appelons ce nouvel ensemble d'expressions $L'(\Sigma)$.

On appellera niveau, tout élément de $L'(\Sigma)$ construit uniquement avec des lettres de Σ u $\{e\}$, et l'opérateur \otimes .

On appellera forme pseudo-normale, tout élément de $L'(\Sigma)$ qui s'écrit sous la forme de produits de composition de niveaux, c'est-à-dire tout $m \in L'(\Sigma)$ est une forme pseudo-normale ssi $m = m_1 \cdot m_2 \cdot \dots \cdot m_n$, avec chacun des m_i niveau.

Calcul de la forme normale :

Nous appliquerons à une expression de $L'(\Sigma)$, un premier système de réécriture (S) dont les règles sont :

(i) - $\forall P, Q, R, S \in L'(\Sigma)$

$(P \cdot Q) \otimes (R \cdot S)$ se réécrit en $(P \otimes R) \cdot (Q \otimes S)$.

(ii) - $\forall R, S \in L'(\Sigma)$

$\forall P \in \Sigma_q^P$ ou $P \in L'(\Sigma)_q^P$ tel que $\exists m_1, m_2 \in L'(\Sigma)$ $P = m_1 \otimes m_2$

$P \otimes (R \cdot S)$ se réécrit en $(P \otimes R) \cdot (e_q \otimes S)$

$(R \cdot S) \otimes P$ se réécrit en $(R \otimes P) \cdot (S \otimes e_q)$

On notera par la suite "se réécrit en" par $\xrightarrow{(S)}$

Ce système de réécriture (S) est noetherien.

Preuve :

Soit v une application de $L'(\Sigma)$ dans \mathbb{N} qui compte le nombre d'occurrences du symbole \otimes dans un élément de $L'(\Sigma)$.

$v : L'(\Sigma) \rightarrow \mathbb{N}$

• $\forall a \in \Sigma \quad v(a) = 0$

• $v(e_0) = v(e) = 0$

• $\forall m_1, m_2 \in L'(\Sigma) \quad v(m_1 \cdot m_2) = v(m_1) + v(m_2)$

$v(m_1 \otimes m_2) = v(m_1) + v(m_2) + 1$

Soit α une application de $L'(\Sigma)$ dans \mathbb{N} , appelée poids qui compte le nombre de \cdot devant passer "à travers" un \otimes .

$$\alpha : L'(\Sigma) \rightarrow \mathbb{N}$$

- $\forall a \in \Sigma \quad \alpha(a) = 0$
- $\alpha(e) = \alpha(e_0) = 0$
- $\alpha(m_1 \cdot m_2) = \alpha(m_1) + \alpha(m_2)$
- $\alpha(m_1 \otimes m_2) = \alpha(m_1) + \alpha(m_2) + v(m_1) + v(m_2)$.

Les applications v et α vérifient de façon triviale, les trois propriétés énoncées ci-après :

Propriété 0 : $\forall m \in L'(\Sigma) \quad v(m) = 0 \Rightarrow \alpha(m) = 0$

Propriété 1 : $\forall m \in L'(\Sigma) \quad \alpha(m) = 0 \Leftrightarrow m$ est une forme pseudo-normale

Propriété 2 : $\forall m, m' \in L'(\Sigma) \quad m \xrightarrow[(S)]{*} m' \Rightarrow \alpha(m) > \alpha(m')$

Pour établir que (S) est noetherien, nous allons constater que toute application d'une règle i) ou ii) fait décroître strictement le poids d'une expression de $L'(\Sigma)$:

- 1 - Si $m \xrightarrow[(S)]{*} m'$ par application d'une règle i) alors une sous-expression m_1 de m est de la forme $(P \cdot Q) \times (R \cdot S)$ et m' est obtenue en substituant dans m , m_1 par m_2 qui est de la forme $(P \times R) \cdot (Q \times S)$.

On a $v(m_1) > v(m_2)$;

- $\alpha(m_1) = \alpha(P) + \alpha(Q) + \alpha(R) + \alpha(S) + v(P) + v(Q) + v(R) + v(S) + 2$
- $\alpha(m_2) = \alpha(P) + \alpha(Q) + \alpha(R) + \alpha(S) + v(P) + v(Q) + v(R) + v(S)$

On a donc $\alpha(m_1) > \alpha(m_2)$.

$$\left. \begin{array}{l} v(m_1) > v(m_2) \\ \text{et} \\ \alpha(m_1) > \alpha(m_2) \end{array} \right\} \Rightarrow \text{quelle que soit la façon dont interviennent } m_1 \text{ et } m_2 \text{ dans leur contexte } \alpha(m) > \alpha(m').$$

2 - Si $m \xrightarrow{(S)} m'$ par application d'une règle ii) ; on prouve par le même procédé que $\alpha(m) > \alpha(m')$.

Cette propriété suffit pour conclure que (S) est noetherien puisque le poids d'une expression est positif ou nul, il ne peut décroître indéfiniment.

On peut établir que la dernière expression obtenue en appliquant (S) à une expression quelconque, est une forme pseudo-normale.

Preuve :

On démontre que tout élément de $L'(\Sigma)$ qui n'est pas une forme pseudo-normale peut être l'objet d'une réécriture dans (S) :

Soit $m \in L'(\Sigma)$ tel que $\alpha(m) \neq 0$; alors deux cas sont envisageables :

cas 1) $m = m_1 \cdot m_2$ et $\alpha(m_1) + \alpha(m_2) \neq 0 \Rightarrow \alpha(m_1) \text{ ou } \alpha(m_2) \neq 0$

Soit m' celui des deux dont le poids est non nul alors

• $m' = m'_1 \cdot m'_2$ et on retombe sur le cas 1)

ou

• $m' = m'_1 \otimes m'_2$ et on retombe sur le cas 2).

cas 2) $m = m_1 \otimes m_2$ et $\alpha(m_1) + \alpha(m_2) + v(m_1) + v(m_2) \neq 0 \Rightarrow v(m_1) \text{ ou } v(m_2) \neq 0$

Soit m' celui des deux tel que $v(m') \neq 0$ alors :

• $m' = m'_1 \cdot m'_2$ auquel cas on peut appliquer à m une réécriture

ou

• $m' = m'_1 \otimes m'_2 \Rightarrow \alpha(m') \neq 0$ et on retombe sur le cas 2).

En analysant de la sorte une expression de $L'(\Sigma)$ de poids non nul, on parvient à isoler une sous-expression qui peut faire l'objet d'une réécriture. \square

On a, en conclusion le résultat suivant :

$$\forall m \in L'(\Sigma) \quad m \stackrel{*}{\underset{(S)}{\sim}} m' \text{ avec } m' \text{ forme pseudo-normale.}$$

Notons également que m et m' sont deux écritures d'un même élément du magmaïde $\overset{\vee}{DP}(\Sigma)$, les règles de (S) étant compatibles avec les axiomes du magmaïde.

Considérons maintenant un second système de réécritures (τ) défini sur l'ensemble des formes pseudo-normales par :

$$- m_1 \cdot m_2 \stackrel{1}{\underset{(\tau)}{\sim}} n_1 \cdot n_2$$

ssi $\exists m_1', m_1'', m_2', m_2''$ niveaux et $a \in \Sigma_p^q$ tels que :

$$m_1 = m_1' \otimes e^q \otimes m_1''$$

$$m_2 = m_2' \otimes a \otimes m_2''$$

$m_1' \cdot m_2'$ et $m_1'' \cdot m_2''$ sont définis

$$n_1 = m_1' \otimes a \otimes m_1''$$

$$n_2 = m_2' \otimes e^p \otimes m_2''$$

dans le cas de lettres de coarité nulle e^q n'apparaît pas dans m_1 , et dans le cas de lettres de arité nulle e^p n'apparaît pas dans n_2 .

$$- m_1 \cdot e_p \stackrel{2}{\underset{(\tau)}{\sim}} m_1 \text{ pour } m_1 \in L(\Sigma)_p^q$$

Remarque :

Par les réécritures de (τ) , on demeure bien dans le sous-ensemble des formes pseudo-normales.

Interprétation graphique des réécritures dans (τ)

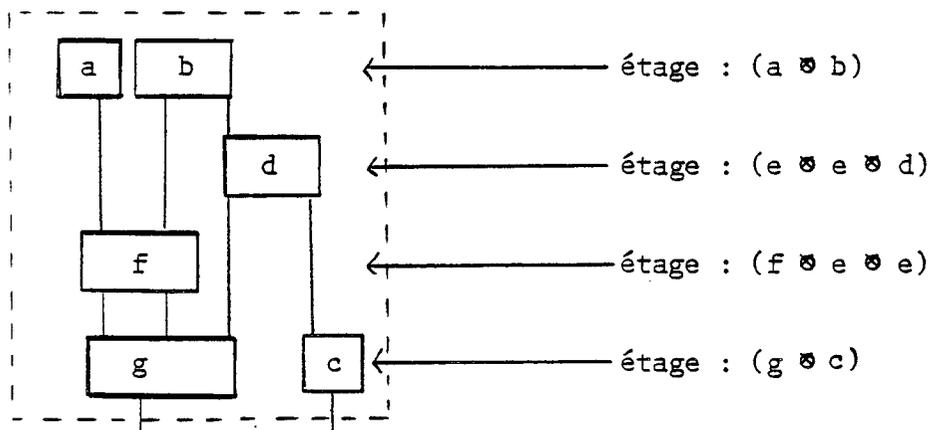
Une forme pseudo-normale se dessine en un empilement d'étages correspondant chacun à un niveau.

Exemple :

Si $a \in \Sigma_1^0$, $b \in \Sigma_2^0$, $c \in \Sigma_1^1$, $d \in \Sigma_2^1$, $f \in \Sigma_2^2$, $g \in \Sigma_1^3$.

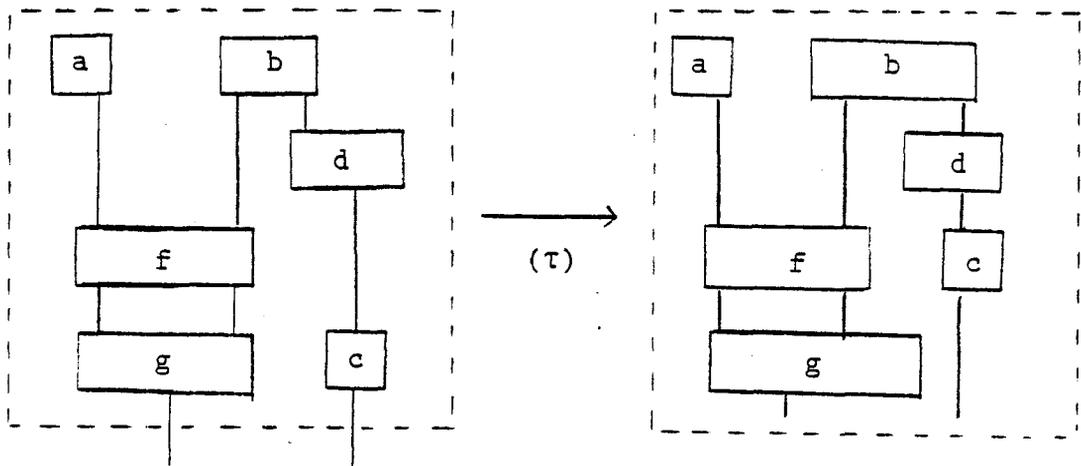
Alors

$(a \otimes b) \cdot (e \otimes e \otimes d) \cdot (f \otimes e \otimes e) \cdot (g \otimes c)$ se dessine :



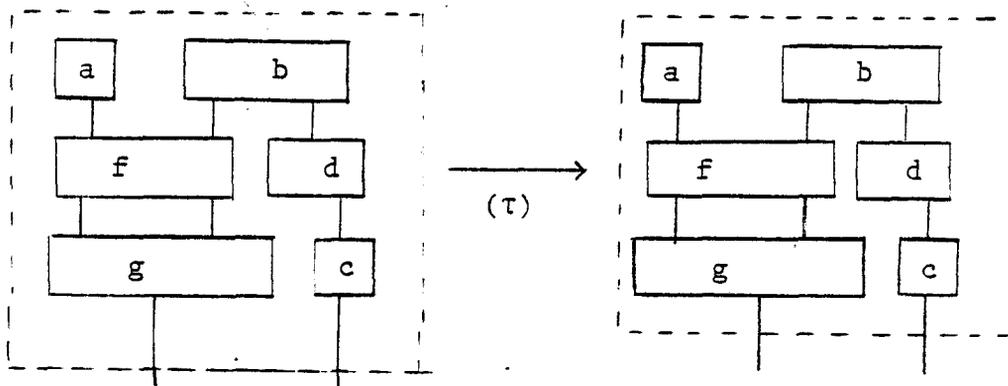
Une réécriture 1 consiste à faire "remonter" des lettres à condition qu'au dessus d'elles "la place soit libre", c'est-à-dire, que la boîte correspondante soit sous un groupe de fils en nombre égal à sa arité.

Exemple :



La réécriture (τ) consiste à supprimer un "étage" constitué uniquement de fils.

Exemple :



On effectue donc par ce système de réécriture une sorte de "tassement" des constituants de l'expression vers le haut.

Nous allons démontrer que (τ) est un système de réécriture noetherien en montrant que toute réécriture dans τ fait décroître strictement une certaine quantité définie sur les formes pseudo-normales.

Cette quantité mesure le nombre de lettres coefficiente par le numéro du niveau dans lequel elle se trouve (un niveau donné aura un poids minimum de 1). D'après l'interprétation graphique donnée précédemment, on conçoit bien que cette grandeur diminue au fur et à mesure des réécritures dans (τ).

Cette grandeur est définie comme une application ρ de l'ensemble des formes pseudo-normales dans \mathbb{N} par :

- $\forall a \in \Sigma \quad \rho(a) = 1$

- $\rho(e) = 0$

- $\forall n$ niveau avec $n = u_1 \otimes u_2 \otimes \dots \otimes u_n$, avec $u_i \in \Sigma \cup \{e\}$ alors

$$\rho(n) = 1 + \sum_{i=1}^n \rho(u_i)$$

- $\forall m$ forme pseudo-normale avec $m = n_1 \cdot n_2 \cdot \dots \cdot n_p$, avec n_j niveau alors

$$\rho(m) = \sum_{j=1}^p j \cdot \rho(n_j)$$

Propriété 3 :

$$\forall m \text{ forme pseudo-normale } \rho(m) > 0 \quad (\text{trivial})$$

Propriété 4 :

$$\forall m, m' \text{ formes pseudo-normales } m \stackrel{(\tau)}{\sim} m' \Rightarrow \rho(m) > \rho(m')$$

Preuve :

cas 1 : m' se déduit de m en appliquant la règle 1

$$\text{d'où } m = m_g \cdot (m_1 \otimes e_q \otimes m_2) \cdot (m_3 \otimes a \otimes m_4) \cdot m_d$$

avec $m_1 \cdot m_3$ et $m_2 \cdot m_4$ définis ; $a \in \Sigma_p^q$

et

$$m' = m_g \cdot (m_1 \otimes a \otimes m_2) \cdot (m_3 \otimes e_p \otimes m_4) \cdot m_d$$

on a donc $\rho(m') = \rho(m) - 1$.

cas 2 : m' se déduit de m en appliquant la règle 2

d'où $m = m_1 \cdot m_2 \cdot \dots \cdot m_k$ avec $m_i = e^p$

et

$$m' = m_1 \cdot m_2 \cdot \dots \cdot m_{i-1} \cdot m_{i+1} \cdot \dots \cdot m_k$$

$$\left. \begin{array}{l} \text{on a donc } \rho(m) = \sum_{j=1}^{i-1} j \cdot \rho(m_j) + i + \sum_{j=i+1}^k j \cdot \rho(m_j) \\ \text{et} \\ \rho(m') = \sum_{j=1}^{i-1} j \cdot \rho(m_j) + \sum_{j=i+1}^k (j-1) \rho(m_j) \end{array} \right\} \Rightarrow \rho(m) > \rho(m')$$

On déduit des propriétés 3 et 4 que le système de réécriture (τ) est noetherien : $\forall m$ forme pseudo-normale $m \xrightarrow{(\tau)^*} m'$; m et m' sont deux notations d'un même élément de $\hat{D}\hat{P}(\Sigma)$; les règles de (τ) ne sont en effet qu'une version orientée des axiomes A_4 et A_5 du magmaïde.

A ce stade de la démonstration de l'existence et de la calculabilité d'une forme normale pour tout élément de $\hat{D}\hat{P}(\Sigma)$, nous sommes assuré que pour toute écriture m d'un élément de $\hat{D}\hat{P}(\Sigma)$ on a :

$$\left\| \begin{array}{l} m \xrightarrow{(S)^*} m' \xrightarrow{(\tau)^*} m'' ; \text{ avec } m' \text{ et } m'' \text{ formes pseudo-normales} \\ \text{et } m, m', m'' \text{ notations du même élément du magmaïde } \hat{D}\hat{P}(\Sigma). \end{array} \right.$$

Notre objectif est de montrer que pour deux notations m_1 et m_2 d'un même élément du magmaïde, on a :

$$\left. \begin{array}{l} m_1 \xrightarrow{(S)^*} m'_1 \xrightarrow{(\tau)^*} m''_1 \\ \text{et} \\ m_2 \xrightarrow{(S)^*} m'_2 \xrightarrow{(\tau)^*} m''_2 \end{array} \right\} \Rightarrow m''_1 = m''_2 \text{ (égalité dans } L'(\Sigma))$$

Preuve :

m_1'' et m_2'' sont deux formes pseudo-normales représentant le même élément du magmaïde.

Soit $m_1'' = n_{11} \cdot n_{12} \cdot \dots \cdot n_{1k}$ avec n_{1i} niveau

Soit $m_2'' = n_{21} \cdot n_{22} \cdot \dots \cdot n_{2\ell}$ avec n_{2i} niveau

Supposons n_{11} différent de n_{21} , alors une lettre apparaissant sur l'étage 1 de la représentation graphique de m_1'' , n'apparaît pas sur l'étage 1 de la représentation graphique de m_2'' (ou vice-versa) ; elle se trouve donc en dessous. Etant donné que ces deux représentations graphiques illustrent un même élément du magmaïde, cette boîte par réduction de la longueur des fils pourrait remonter vers les étages supérieurs. L'une des deux écritures est donc encore susceptible d'une réécriture dans (τ) , conclusion contradictoire avec notre hypothèse de départ, d'où $n_{11} = n_{21}$.

Par induction, on en déduit que si $k \leq \ell$ alors $\forall i \in [1 \dots k] n_{1i} = n_{2i}$.

Les étages supplémentaires dans l'une ou l'autre des deux écritures ne pourraient être constitués que d'éléments neutres, ceci est en contradiction avec le fait que m_1'' et m_2'' sont des réécritures achevées dans (τ) ; on a donc nécessairement $k = \ell$.

$\Rightarrow m_1'' = m_2''$ (égalité dans $L'(\Sigma)$). \square

On appellera $FN(m)$ l'écriture ainsi obtenue pour $m \in \overset{\vee}{DP}(\Sigma)$.

I.4. - FORME NORMALE DUALE D'UN ELEMENT DU MAGMOÏDE $\tilde{DP}(\Sigma)$

En considérant, les axiomes orientés du magmoïde comme les règles d'un système de réécritures, on obtient comme nous l'avons vu précédemment, par application de ce système de réécritures, la même réécriture terminale pour des éléments d'une même classe d'équivalence de $L(\Sigma)$: la forme normale.

Observons, cependant, qu'à des choix différents du sens d'orientation des axiomes, correspondent des systèmes de réécritures différents dont on pourrait montrer le caractère noetherien et confluent sur les classes d'équivalence de $L(\Sigma)$. On calcule donc par ces différents systèmes de réécritures à chaque fois des représentants particuliers mais distincts des classes d'équivalence de $L(\Sigma)$ (i.e. les éléments du magmoïde). On a donc le choix entre différents type de "forme normale".

Nous nous intéressons dans notre étude à deux types de formes normales ; celle mise en évidence au chapitre I.6. qui sera appelée forme normale, et une seconde dont nous allons expliciter la procédure de calcul qui sera appelée forme normale duale.

Calcul de la forme normale duale

La forme normale duale sera obtenue en appliquant à un élément de $L'(\Sigma)$ successivement (S) tel qu'il a été défini au chapitre I.6. et (τ') qui reprend les règles de (τ) du chapitre I.6, en inversant le sens de la règle 1.

Calculer la forme normale duale d'une expression de $L'(\Sigma)$ consiste donc à :

- trouver une forme pseudo-normale par application de (S),
- descendre les lettres le plus bas possible, tout en réduisant au maximum le nombre de niveaux. (Interprétation symétrique de la règle 1 de (τ)).

Notation :

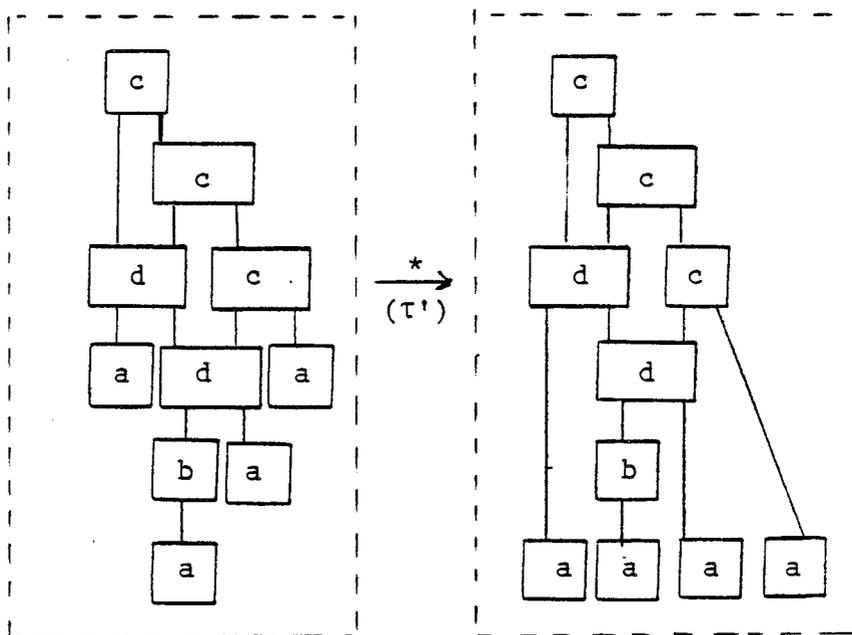
On désignera par $FND(m)$, la forme normale duale associée à un élément m de $\hat{DP}(\Sigma)$.

Exemple de forme normale duale

Soit Σ alphabet bigradué avec $a \in \Sigma_0^1$, $b \in \Sigma_1^1$, $c \in \Sigma_2^1$, $d \in \Sigma_2^2$ et $m = c \cdot (e \otimes c) \cdot (d \otimes c) \cdot (a \otimes d \otimes a) \cdot (b \otimes a) \cdot a \in L'(\Sigma)$.

Etant donné que m est une expression pseudo-normale, on applique directement (τ') pour déterminer la forme normale duale.

Regardons comment opère (τ') sur les représentations graphiques des éléments de $L'(\Sigma)$.



La forme normale duale représentant le même élément du magmaïde que m s'écrit donc :

$$c \cdot (e \otimes c) \cdot (d \otimes c) \cdot (e \otimes d \otimes e) \cdot (e \otimes b \otimes e \otimes e) \cdot (a \otimes a \otimes a \otimes a).$$

I.5. - FEUILLAGE D'UN DAG PLANAIRE

Notation :

Σ^{\otimes} désignera le monoïde libre construit à partir de l'alphabet $\Sigma \cup \{e\}$, muni de l'opération \otimes et de l'élément neutre e_0 .

Définition :

On définit alors l'application de feuillage ϕ de $\hat{DP}(\Sigma)$ dans Σ^{\otimes} comme celle qui à un élément de magmaïde $\hat{DP}(\Sigma)$ associe le dernier niveau de la forme normale duale qui lui est associée.

Soit $m \in \hat{DP}(\Sigma)$ et $FND(m) = m_1 \cdot m_2 \cdot \dots \cdot m_n$ avec m_i niveaux alors $\phi(m) = m_n$.

Propriété :

- Si $t \in \Sigma \cup \{e\}$ $\phi(t) = t$.
- $\forall m_1, m_2 \in \hat{DP}(\Sigma)$ $\phi(m_1 \otimes m_2) = \phi(m_1) \otimes \phi(m_2)$.
- D'une manière générale, on n'a pas $\phi(m_1 \cdot m_2) = \phi(m_2)$.

Exemple : $\Sigma = \{a, b, c, d\}$ (cf. exemple précédent)

$$m_1 = c \cdot (a \times b) \text{ et } m_2 = a$$

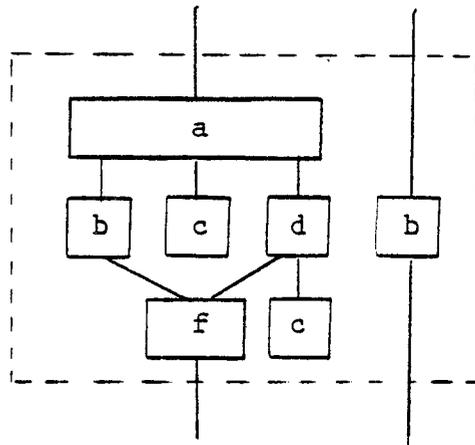
$$\phi(m_1 \cdot m_2) = a \times a \text{ et } \phi(m_2) = a.$$

Remarque :

Pour tout élément de $\hat{DP}(\Sigma)$, qui peut être considéré comme un arbre (i.e. tous les noeuds de son graphe n'ont qu'un seul arc rentrant) la définition de feuillage que nous venons de donner, coïncide exactement avec la définition classique du feuillage d'un arbre.

Exemple :

Le dag planaire :



a comme forme normale :
 $(a \otimes b) \cdot (b \otimes c \otimes d \otimes e)$.
 $(f \otimes c \otimes e)$

a comme forme normale duale :
 $(a \otimes e) \cdot (b \otimes c \otimes d \otimes e)$.
 $(f \otimes c \otimes b)$

a comme feuillage :
 $f \otimes c \otimes b$.

I.6. - EXTENSIONS DES OPERATIONS DU MAGMOÏDE, AUX TAILLIS

I.6.1. - Définition d'un TAILLIS

Soit M , un magmoïde, on appellera $\mathcal{P}(M)$, l'ensemble des parties de ce magmoïde. Un élément de $\mathcal{P}(M)$ est appelé un taillis.

Un TAILLIS est donc aux dags planaires ce qu'un langage est aux mots ou une forêt aux arbres.

I.6.2. - Produit tensoriel de deux taillis, noté \otimes

$\forall A, \forall B \in \mathcal{P}(M)$

$$A \otimes B = \{a \otimes b \in M / a \in A \text{ et } b \in B\}.$$

I.6.3. - Produit de composition de deux taillis, noté \cdot

$\forall A, \forall B \in \mathcal{P}(M)$

$$A \cdot B = \{a \cdot b \in M / a \in A \text{ et } b \in B\}.$$

I.6.4. - Etoile du produit tensoriel de deux taillis, notée \otimes^*

Pour tout $n \geq 0$, on définit $A^{\otimes n}$ par induction, où $A \in \mathcal{P}(M)$

$$A^{\otimes 0} = \{e_0\}; A^1 = A; A^2 = A \otimes A; \dots; A^{i+1} = A^i \otimes A$$

et

$$A^{\otimes^*} = \bigcup_{n \geq 0} A^{\otimes n}$$

I.6.5. - Etoile du produit de composition de deux taillis,
notée *

Pour tout $n \geq 0$ on définit A^{*n} par induction, où $A \in \mathcal{P}(M)$

$$A^{*0} = \{e\} ; A^{*1} = A ; \dots ; A^{*i+1} = A^i \cdot A$$

et

$$A^* = \bigcup_{n \geq 0} A^{*n}$$

Remarque :

Ces opérations ensemblistes ne sont pas à rapprocher de la définition classique du magmaïde des parties.

L'ensemble des taillis muni des opérations \circ et \cdot définies ci-dessus ne forme pas un magmaïde.

CHAPITRE II

LES AUTOMATES DE DAGS PLANAIRES

PLAN DU CHAPITRE II

II.1. - DEFINITIONS ET CLASSIFICATIONS

- II.1.1. - Automates descendants
- II.1.2. - Exemple et critiques des définitions
- II.1.3. - Dualité entre automates ascendants et automates descendants
- II.1.4. - Théorème de "remontée (resp. descente) du contrôle bas (resp. haut)"
- II.1.5. - Automates d'arbres et automates de dags planaires
- II.1.6. - Automates déterministes

II.2. - PROPRIETES DE CLOTURE

- II.2.1. - Clôture par union
- II.2.2. - Clôture par intersection
- II.2.3. - Clôture par produit tensoriel
- II.2.4. - Clôture par étoile du produit tensoriel
- II.2.5. - Clôture par produit de composition
- II.2.6. - Non-clôture par étoile du produit de composition
- II.2.7. - Non-clôture par homomorphisme de magmaïde
- II.2.8. - Clôture par homomorphisme connexe
- II.2.9. - Clôture par homomorphisme inverse
- II.2.10. - Problème ouvert : clôture de REC par complémentaire

II. LES AUTOMATES D'ETATS FINIS DE DAGS PLANAIRES

Notation :

Q^{\otimes} désignera le monoïde libre construit à partir de l'alphabet Q , muni de l'opération \otimes et de l'élément neutre e_0 .

II.1. - DEFINITIONS ET CLASSIFICATION

Comme dans le cas des arbres, on définit les automates ascendants et descendants.

II.1.1. - Automates descendants

II.1.1.1. - Automates descendants d'états finis de dags planaires

C'est la donnée d'un quintuplet $M = \langle \Sigma, Q, I, F, R \rangle$ où

- a) Σ est un ensemble fini de lettres bigraduées de co-arité et de arité > 0 .
- b) Q est un ensemble fini d'états bigradués de co-arité et de arité égaux à 1. $\Sigma \cap Q = \emptyset$, on note Q^{\otimes} le monoïde libre (Q, \otimes) , d'élément neutre e_0 .
- c) $I \subseteq Q^{\otimes}$, langage rationnel, appelé indifféremment contrôle haut ou contrôle initial.
- d) $F \subseteq Q^{\otimes}$, langage rationnel, appelé indifféremment contrôle bas ou contrôle final.
- e) R est ensemble de règles, inclus dans $Q^{\otimes} \times \Sigma \times Q^{\otimes}$, écrites sous la forme :

$$(q_1 \otimes q_2 \otimes \dots \otimes q_n) \cdot c \rightarrow c \cdot (q'_1 \otimes \dots \otimes q'_n),$$

où $c \in \Sigma_n^n$,

II.1.1.2. - Facteur d'un dag planaire

$m \in \overset{\vee}{DP}(\Sigma)$ est un facteur de $\delta \in \overset{\vee}{DP}(\Sigma)$ ssi

soit $\delta = \delta_1 \cdot m \cdot \delta_2$

soit $\delta = \delta_1 \otimes m \otimes \delta_2$

soit m est facteur d'un facteur de δ .

II.1.1.3. - Mouvement élémentaire de l'automate

C'est une relation sur $\overset{\vee}{DP}(\Sigma \cup Q)$: $\delta \mid_M^* \delta'$ si et seulement si δ' est obtenu en remplaçant un facteur m de δ , par m' tel que $m \rightarrow m' \in R$:

On note \mid_M^* , la clôture réflexive et transitive.

II.1.1.4. - Taillis reconnu par l'automate

Soit $M = \langle \Sigma, Q, I, F, R \rangle$, un automate descendant d'états finis de dags planaires. On note $T(M)$, le taillis reconnu par M et définit par :

$$T(M) = \{ \delta \in \overset{\vee}{DP}(\Sigma) / \exists q_i \in I, \exists q_f \in F \text{ avec } q_i \cdot \delta \mid_M^* \delta \cdot q_f \}.$$

II.1.1.5. - Taillis reconnaissable

$T \subseteq \overset{\vee}{DP}(\Sigma)$ est un taillis reconnaissable si et seulement si $\exists M$, automate descendant d'états finis de dags planaires tel que $T(M) = T$.

L'ensemble des taillis reconnaissable est noté REC.

Remarque :

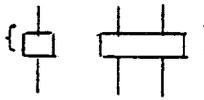
Pour des raisons de commodité nous ne préciserons pas toujours la nature, «d'états finis de dags planaires», des automates quand il n'y aura pas d'ambiguïté sur les automates dont il est question.

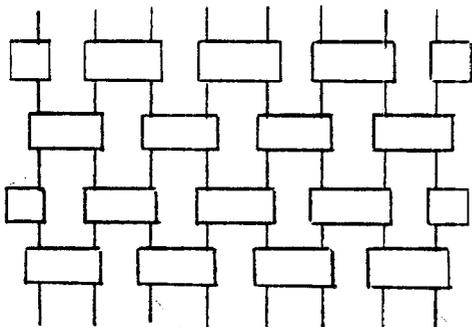
II.1.2. - Exemple et critiques des définitions

II.1.2.1. - Exemple de taillis reconnaissable

Exemple :

Un ensemble de «murs» .

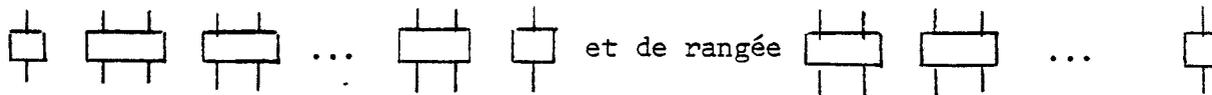
Soit $\Sigma = \{a, b\}$ encore écrit pour l'exemple qui suit : 



représente un dag planaire sur Σ ,
qui s'écrit sous forme normale avec
les notations du magmaïde :

$$(a \otimes b \otimes b \otimes b \otimes a) \cdot (b \otimes b \otimes b \otimes b) \cdot \\ (a \otimes b \otimes b \otimes b \otimes a) \cdot (b \otimes b \otimes b \otimes b)$$

La forme générale de tel dag planaire étant : une alternance de rangée



L'automate reconnaissant le taillis ci-dessus : $M = \langle \Sigma, Q, I, F, R \rangle$

où

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

$$I = F = q_0 \otimes (q_1 \otimes q_2)^* \otimes q_0 + (q_1 \otimes q_2)^+ \text{ et}$$

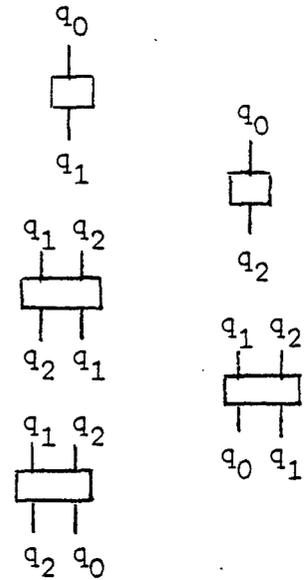
R : $q_0 \cdot a \rightarrow a \cdot q_1$ ou encore, écrit graphiquement

$q_0 \cdot a \rightarrow a \cdot q_2$ " "

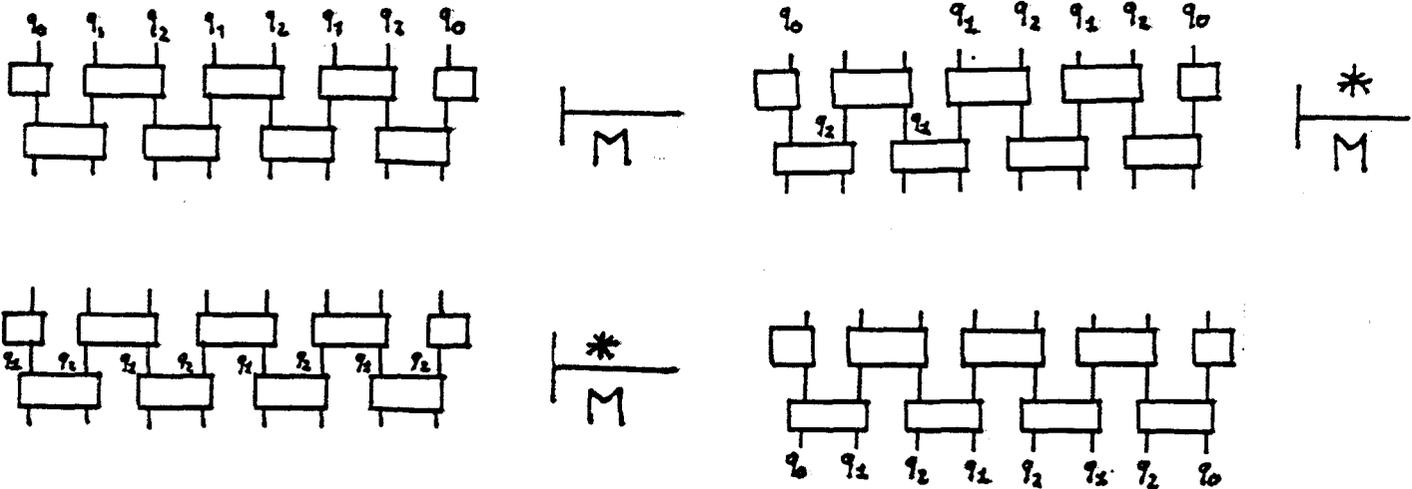
$(q_1 \otimes q_2) \cdot b \rightarrow b \cdot (q_2 \otimes q_1)$ " "

$(q_1 \otimes q_2) \cdot b \rightarrow b \cdot (q_0 \otimes q_1)$ " "

$(q_1 \otimes q_2) \cdot b \rightarrow b \cdot (q_2 \otimes q_0)$ " "



ainsi



ce qui s'écrit encore, sous forme normale :

$$(q_0 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_0) \cdot (a \otimes b \otimes b \otimes b \otimes a) \cdot (b \otimes b \otimes b \otimes b) \Big|_M^*$$

$$(q_0 \otimes b \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_0) \cdot (a \otimes q_2 \otimes q_1 \otimes b \otimes b \otimes a) \cdot (b \otimes b \otimes b \otimes b) \Big|_M^*$$

$$(a \otimes b \otimes b \otimes b \otimes a) \cdot (q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2) \cdot (b \otimes b \otimes b \otimes b) \Big|_M^*$$

$$(a \otimes b \otimes b \otimes b \otimes a) \cdot (b \otimes b \otimes b \otimes b) \cdot (q_0 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_0).$$

Et notre «mur» est reconnu par M, puisque $(q_0 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_1 \otimes q_2 \otimes q_0)$ est élément de I et de F.

II.1.2.2. - Critiques des définitions

Les définitions proposées pour les automates descendants de dags planaires sont classiques, cf. automates de mots en [29] et automates d'arbres en [33]. Cependant certains choix méritent d'être détaillés.

II.1.2.2.1. - Choix du contrôle haut et du contrôle bas

Il est impératif que tout sous-ensemble fini de $\hat{DP}(\Sigma)$, comme par exemple $\{a \otimes a\}$, soit reconnaissable. D'autre part, on peut établir immédiatement un isomorphisme entre le taillis $\{a \otimes b, a \otimes a \otimes b, \dots\}$ et le langage rationnel de mots a^+b . Ces deux remarques justifient que le contrôle haut (ou bas) soit un langage rationnel de mots d'états et non un ensemble d'états.

On montrera ultérieurement [II.1.4.] que l'un des deux contrôles peut toujours se restreindre à un état, i.e. $I = \{q_0\}^{\otimes}$ ou $F = \{q_f\}^{\otimes}$, sans rien perdre de la reconnaissabilité.

II.1.2.2.2. - Absence de lettres de co-arité et arité nulles

II.1.2.2.2.1. - Si on s'autorise des lettres de co-arité ou de arité nulle, des lettres telles que $\overset{\uparrow}{a} \in \Sigma_1^0$ et $\overset{\downarrow}{a'} \in \Sigma_0^1$ seraient permises et il serait raisonnable de considérer le taillis fini

$$\left\{ \begin{array}{c} a \\ | \\ a' \end{array} \right\} \subseteq \hat{DP}(\Sigma)_0^0$$

comme un taillis reconnaissable.

Une telle démarche nécessiterait une adaptation de la définition des automates de dags planaires : il s'agirait, comme pour les automates d'arbres classiques [] de faire naître un état à la lecture d'une lettre a et de faire disparaître un état à la lecture d'une lettre a' . Donc un tel dag $\begin{array}{c} a \\ | \\ a' \end{array}$ serait reconnu en partant d'aucun état et en arrivant à aucun état.

En d'autres termes sa reconnaissance ne serait subordonnée à aucun lien avec l'extérieur et de tel dag pourrait proliférer, sans aucun contrôle, dans n'importe quel dag reconnu par ailleurs. Ainsi un automate reconnaissant $\begin{array}{c} a \\ | \\ a' \end{array}$, reconnaîtrait obligatoirement $a \otimes a$, $a \otimes a \otimes a$, ...

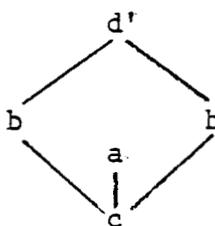
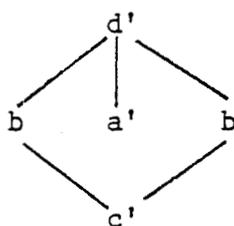
$$\begin{array}{c} a \\ | \\ a' \end{array} \otimes \begin{array}{c} a \\ | \\ a' \end{array}, \begin{array}{c} a \\ | \\ a' \end{array} \otimes \begin{array}{c} a \\ | \\ a' \end{array} \otimes \begin{array}{c} a \\ | \\ a' \end{array}, \dots$$

et donc le taillis $\left\{ \begin{array}{c} a \\ | \\ a' \end{array} \right\}$ ne serait pas reconnaissable, ce qui explique la restriction prise.

II.1.2.2.2.2. - On aurait pu rompre la symétrie en ne s'interdisant que la co-arité nulle ou que l'arité nulle. Nous avons exclu ce cas par souci de simplicité.

Conséquence :

Les dags de la forme



sont exclus de la reconnaissabilité.

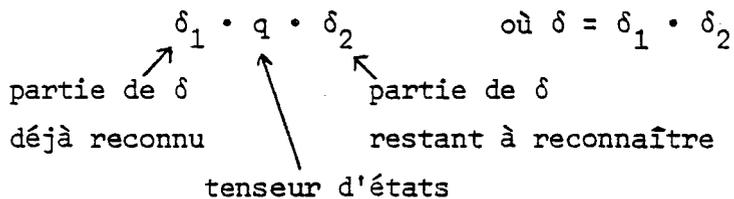
Remarque :

Notre travail pourrait être étendu de façon à considérer ces derniers cas sans changer le fond de la démarche, mais en compliquant formalismes et preuves par l'introduction de cas supplémentaires.

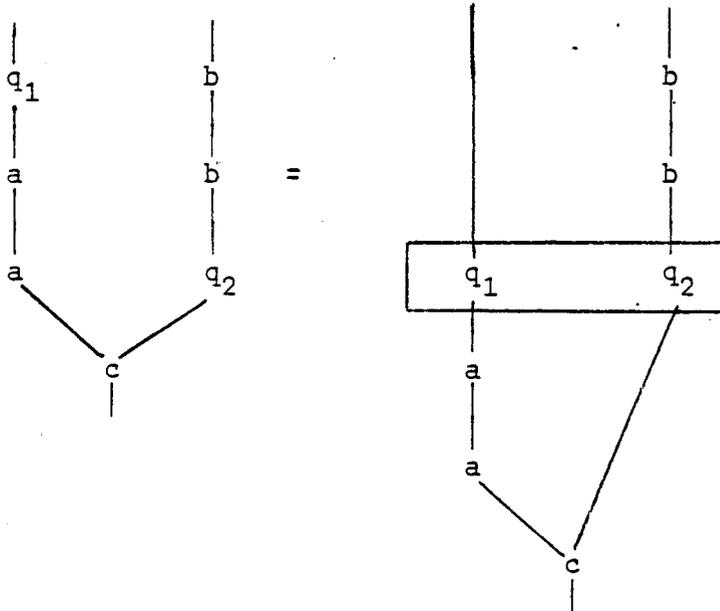
II.1.2.2.3. - Choix du cadre formel des reconnaissances : réécritures
dans le magmaïde

Les trois *remarques* suivantes ont pour but d'illustrer que l'association d'un mouvement élémentaire à une réécriture dans le magmaïde permet de manipuler facilement les reconnaissances des automates. Cette facilité rendra plus crédible les démonstrations futures.

II.1.2.2.3.1. - Dans une reconnaissance d'un dag planaire δ , chaque étape de réécriture de cette reconnaissance peut s'écrire, dans le magmaïde, sous la forme :



Exemple :



en se rappelant [] que dans la représentation graphique d'un élément d'un magmaïde l'élément neutre, e , du produit de composition est représenté par un « fil ». La longueur de ce fil pouvant être variable puisque $e \cdot e \cdot \dots \cdot e = e$.

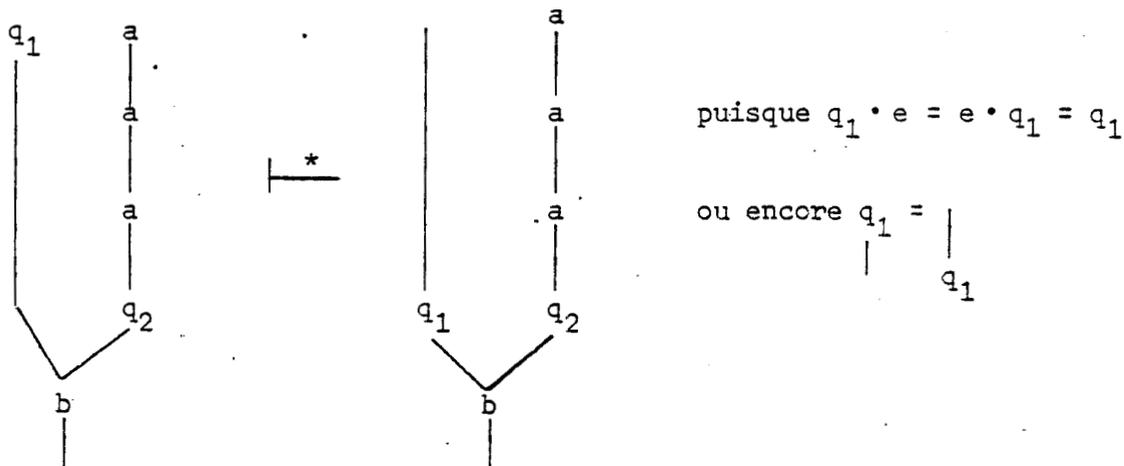
L'exemple proposé pouvant encore s'écrire :

$$(q_1 \otimes b) \cdot (a \otimes b) \cdot (a \otimes q_2) \cdot c = \underbrace{(e \otimes b) \cdot (e \otimes b)}_{\delta_1} \cdot \boxed{(q_1 \otimes q_2)} \cdot \underbrace{(a \otimes e) \cdot (a \otimes e)}_{\delta_2} \cdot c$$

II.1.2.2.3.2. - Dans l'exemple ci-dessus, du fait de l'égalité dans le magmaïde des deux expressions et de la réflexivité de \vdash^* , on peut écrire :

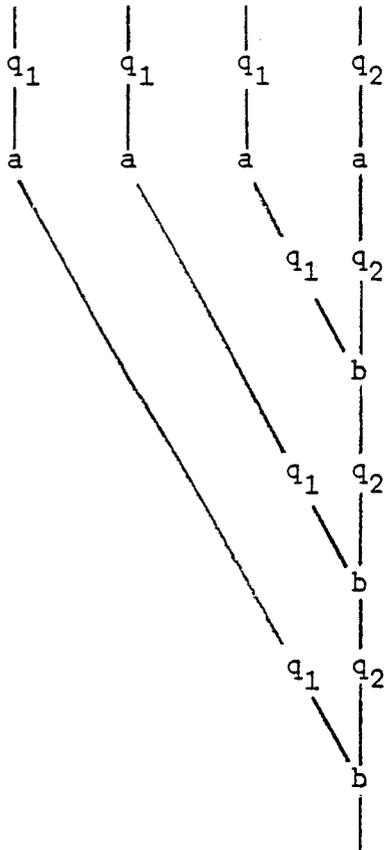
$$(q_1 \otimes b) \cdot (a \otimes b) \cdot (a \otimes q_2) \cdot c \vdash^* (e \otimes b) \cdot (e \otimes b) \cdot (q_1 \otimes q_2) \cdot (a \otimes e) \cdot (a \otimes e) \cdot c$$

Ce phénomène peut s'appeler «faire glisser un état le long d'un fil» et est mieux illustrer par l'exemple suivant



II.1.2.2.3.3. - Dans une reconnaissance de dags planaires, une lettre n'étant reconnu qu'une fois et une seule, on peut décrire toutes les réécritures d'une reconnaissance en un seul dag : en insérant avant et après chaque lettre les tenseurs d'états qui ont permis de reconnaître cette lettre. Plus formellement, on obtient un élément de magmaïde typé [VI.1.], cet élément est obtenu par le système de reconnaissance τ associé à l'automate [VII.2.1.].

Exemple :



II.1.3. - Dualité de l'automate descendant et de l'automate ascendant

II.1.3.1. - Automate ascendant

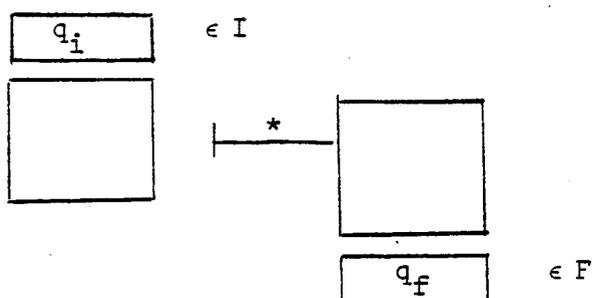
La définition de l'automate ascendant s'obtient en inversant le sens des règles dans la définition de l'automate descendant.

II.1.3.2. - Taillis reconnu par un automate ascendant

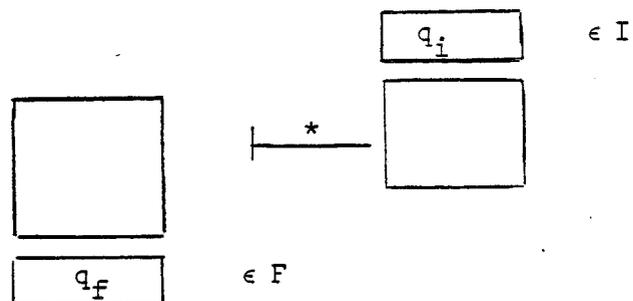
Soit $M = \langle \Sigma, Q, I, F, R \rangle$, un automate ascendant. On note $T(M)$, le taillis reconnu par M et définit par :

$$T(M) = \{ \delta \in \overset{\vee}{DP}(\Sigma) / \exists q_f \in F, \exists q_i \in I \text{ tels que } \delta \cdot q_f \stackrel{*}{\underset{M}{\mid}} q_i \cdot \delta \}.$$

Reconnaissance *descendante*



Reconnaissance *ascendante*



II.1.3.3. - Dualité

Il est clair que toutes propriétés dans le cas descendant se retrouveront symétriquement dans le cas ascendant et réciproquement. Plus précisément, soit Δ , l'opérateur qui inverse le sens des flèches des règles de l'automate.

$$\begin{array}{ccc}
 M = \langle \Sigma, Q, I, F, R \rangle & \xrightarrow{\Delta} & M_{\Delta} = \langle \Sigma, Q, I, F, R_{\Delta} \rangle \\
 \text{(descendant)} & & \text{(ascendant)} \\
 & \xleftarrow{\Delta} &
 \end{array}$$

Il est évident que :

- $(M_{\Delta})_{\Delta} = M$

si M est ascendant (resp. descendant) alors M_{Δ} est descendant (resp. ascendant)

- $q \cdot \delta \stackrel{*}{\mid}_M \delta \cdot q' \iff \delta \cdot q' \stackrel{*}{\mid}_{M_{\Delta}} q \cdot \delta$

- $T(M) = T(M_{\Delta})$

Cette dualité permet de parler de propriétés de taillis reconnaissables sans préciser si elles sont obtenues à partir d'un automate ascendant ou un automate descendant. Dans la suite, les démonstrations seront faites dans le cas des automates descendant, sans transposer les démonstrations au cas ascendant.

II.1.4. - Théorème de "remontée (resp. descente) du contrôle bas (resp. haut)"

Le théorème suivant montre que dans un automate le contrôle bas peut toujours se réduire à un ensemble d'états, dit états finaux, c'est-à-dire $F = Q_f^{\otimes}$ avec $Q_f \subseteq Q$. Ensuite un corollaire établit que cet ensemble d'états finaux peut toujours se réduire à un état final, c'est-à-dire $F = \{q_f\}^{\otimes}$ avec $q_f \in Q$. Ce résultat peut se transposer au contrôle haut, ce qui nous assure que l'on peut toujours réduire à un état, l'un des deux contrôles, c'est-à-dire $I = \{q_i\}^{\otimes}$, avec $q_i \in Q$ ou $F = \{q_f\}^{\otimes}$ avec $q_f \in Q$.

L'intérêt de ce théorème vient du fait que l'un des deux contrôles devient "local" : présence ou absence de l'état final (resp. initial) en fin (resp. début) de reconnaissance.

II.1.4.1. - Enoncé

$\forall M = \langle \Sigma, Q, I, F, R \rangle$, automate de dags planaires, il existe effectivement $M' = \langle \Sigma, Q', I', F', R' \rangle$ tel que

- $F' = Q_F^{\otimes}$ où $Q_F \subseteq Q'$
- $T(M) = T(M')$

II.1.4.2. - Notations

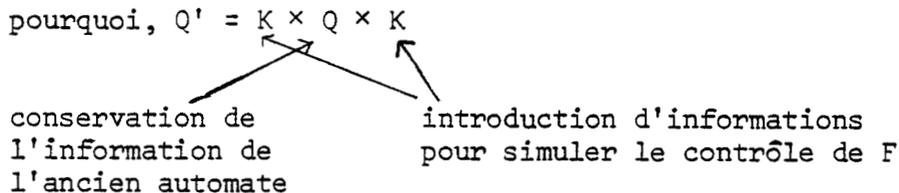
- $A_F = \langle Q, K, \{k_0\}, \{k_f\}, \lambda \rangle$, l'automate de mots sur Q , qui reconnaît F .
- $\Delta_{k_i, k_j} = \{m \in (K \times Q \times K)^{\otimes} / (k_i, q_1, k_1)(k_1, q_2, k_2) \dots (k_n, q_{n+1}, k_j)\}$
pour k_i et $k_j \in K$.

II.1.4.3. - Idée de construction

L'information contenu dans le contrôle bas F , sera, dans le nouvel automate M' , réparti dans :

- le contrôle haut,
- les règles de transitions,
- le choix des états de Q_f .

Le contrôle haut et les règles de transitions devront contenir les informations qu'ils avaient auparavant pour avoir $T(M') \subseteq T(M)$ et posséder des informations relatives au contrôle bas pour avoir $T(M) = T(M')$. C'est pourquoi, $Q' = K \times Q \times K$



II.1.4.4. - Construction

$$Q' = K \times Q \times K$$

$$I' = \{m \in Q'^{\otimes} / \Pi_Q(m) \in I \text{ et } m \in \Delta_{k_0, k_f}\}$$

$$F' = \lambda^{\otimes}$$

$$R' = \{(m, a, m') \in Q'^{\otimes} \times \Sigma \times Q'^{\otimes} \text{ tel que}$$

$$\bullet a \in \Sigma_{\substack{|m| \\ |m'|}}$$

$$\bullet (\Pi_Q(m), a, \Pi_Q(m')) \in R$$

$$\bullet m \text{ et } m' \in \Delta_{k_i, k_j}, \text{ pour } k_i, k_j \in K\}$$

- Le rôle des première et troisième composantes de Q' est d'effectuer le contrôle qui était effectué par le contrôle bas.

- Le rôle de la deuxième composante de Q' est d'effectuer le même contrôle de l'ancien automate M , ce qui explique que

$$(\Pi_Q(m), a, \Pi_Q(m')) \in R \text{ et}$$

$$\Pi_Q(I') = I$$

- Les reconnaissances admises par M' peuvent se schématiser :

$$(k_0, q_1, k_1) (k_1, q_2, k_2) \dots (k_{n-1}, q_n, k_f)$$

$$\omega$$

$$\frac{*}{M'}$$

$$\omega_1$$

$$(k_0, q'_1, k'_1) (k'_1, q'_2, k'_2) \dots (k'_{n'-1}, q'_{n'}, k_k)$$

$$\omega_2$$

$$\frac{*}{M'}$$

$$\omega$$

$$(k_0, q''_1, k''_1) (k''_1, q''_2, k''_2) \dots (k''_{n''-1}, q''_{n''}, k_f)$$

$$\rightarrow \in \Delta_{k_0, k_f}$$

II.1.4.5. - Démonstration

Après avoir montré par récurrence sur le nombre d'occurrences de lettres de δ .

II.1.4.5.1. - Lemme

$\forall \delta \in \tilde{DP}(\Sigma) : \exists q_i \in Q^{\otimes}, \exists q_f \in Q^{\otimes}$ tels que $q_i \cdot \delta \stackrel{*}{M} \delta \cdot q_f$
 $\Leftrightarrow \exists m_1 \in \Delta_{k_0, k_f}$ avec $\Pi_Q(m_1) = q_i, \exists m_2 \in \Delta_{k_0, k_f}$ avec $\Pi_Q(m_2) = q_f$
 tels que $m_1 \cdot \delta \stackrel{*}{M'} \delta \cdot m_2$.

il vient :

(a) $\delta \in T(M) \Leftrightarrow \exists q_i \in I, \exists q_f \in F$ tels que $q_i \cdot \delta \stackrel{*}{M} \delta \cdot q_f$ en utilisant le lemme

(b) $\Leftrightarrow \exists m_1 \in \Delta_{k_0, k_f}$ avec $\Pi_Q(m_1) = q_i \in I, \exists m_2 \in \Delta_{k_0, k_f}$ avec $\Pi_Q(m_2) = q_f \in F$
 tels que $m_1 \cdot \delta \stackrel{*}{M'} \delta \cdot m_2$.

En remarquant que : $m_1 \in \Delta_{k_0, k_f}$ et $m_1 \cdot \delta \stackrel{*}{M'} \delta \cdot m_2$
 $\Rightarrow m_2 \in \Delta_{k_0, k_f}$ et que d'autre part $m_2 \in \lambda^{\otimes} = F' \Leftarrow m_2 \in \Delta_{k_0, k_f}$

(c) $\Leftrightarrow \exists m_1 \in I', \exists m_2 \in F'$ tels que $m_1 \cdot \delta \stackrel{*}{M'} \delta \cdot m_2$

(d) $\Leftrightarrow \delta \in T(M')$.

□

II.1.4.5.2. - Preuve du lemme

Par récurrence sur le nombre, l , d'occurrences de lettres de δ

• $l = 0 \Rightarrow \omega = e_p$ ($e_p = \underbrace{e \otimes \dots \otimes e}_{p \text{ fois}}$), $q_i = q_f$ pour $m_1 = m_2$ avec $\Pi_Q(m_1) = \Pi_Q(m_2)$
 $= q_i = q_f$. Le lemme se vérifie.

Hypothèse de récurrence :

$\forall r \in \mathbb{N}^*$, le lemme est vérifié pour $\delta \in \overset{\vee}{\text{DP}}(\Sigma)$ tel que son nombre, ℓ , d'occurrences de lettres soit strictement inférieur à r .

- $\ell = r \Rightarrow \delta = \delta_1 \cdot (e_i \otimes a \otimes e_j)$, où le nombre d'occurrences de lettres de δ_1 est $\ell-1$, et on a :

$$(a) \Leftrightarrow \exists q_i \in Q^{\otimes}, \exists q_f \in Q^{\otimes}, \exists q \in Q^{\otimes} \text{ tels que } q_i \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta_1 \cdot q \cdot (e_i \otimes a \otimes e_j) \stackrel{*}{\underset{M}{\mid}} \delta \cdot q_f.$$

En utilisant l'hypothèse de récurrence et en remarquant $q \cdot (e_i \otimes a \otimes e_j) \stackrel{*}{\underset{M}{\mid}} (e_i \otimes a \otimes e_j) \cdot q_f$

$$(b) \Leftrightarrow \exists m_i \in \Delta_{k_0, k_f} \text{ avec } \Pi_Q(m_i) = q_i, \exists m \in \Delta_{k_0, k_f} \text{ avec } \Pi_Q(m) = q \text{ tels que } m_i \cdot \delta_1 \stackrel{*}{\underset{M}{\mid}} \delta_1 \cdot m, \text{ et}$$

$\exists q_g \cdot a \rightarrow a \cdot q_d \in R$ telle que $q = q_1 \otimes q_g \otimes q_2$ et $q_f = q_1 \otimes q_d \otimes q_2$ en considérant la définition de R' .

$$(c) \Leftrightarrow \exists m_i \in \Delta_{k_0, k_f} \text{ avec } \Pi_Q(m_i) = q_i, \exists m_f \in \Delta_{k_0, k_f} \text{ avec } \Pi_Q(m_f) = q_f \text{ tels que } m_i \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta \cdot m_f.$$

□

II.1.4.6. - Corollaire

$\forall M = \langle \Sigma, Q, I, F, R \rangle$, automate de dags planaires, il existe effectivement $M'' = \langle \Sigma, Q'', I'', F'', R'' \rangle$ tel que

$$\bullet F'' = \{q_f\}^{\otimes} \quad \text{où } q_f \in Q''$$

$$\bullet T(M) = T(M'')$$

II.1.4.6.1. - Idée de construction

On ajoute un nouvel état, q_f , à la construction précédente.

• On reprend les mêmes éléments de M' pour avoir *les mêmes réécritures* et *en plus* on donne la *possibilité* à chaque état final de F' , apparaissant soit à droite dans une règle de transition, soit dans I' d'être remplacé par q_f (ainsi q_f est un état final que l'on ne peut réécrire).

• Enfin, on définit $F'' = \{q_f\}^{\otimes}$.

Remarque :

Cette construction va introduire un indéterminisme important puisque chaque fois qu'un état final de F' pourra apparaître, il faudra décider, *a priori* :

- s'il doit être réécrit par la suite, et choisir la règle qui le laisse tel quel.
- s'il ne doit pas être réécrit, et choisir la règle qui fait apparaître q_f à sa place.

II.1.4.6.2. - Construction

Soit M' , l'automate construit au théorème précédent alors $M'' = \langle \Sigma, Q' \cup \{q_f\}, I'', \{q_f\}^{\otimes}, R'' \rangle$ où

- $Q' \cap \{q_f\} = \emptyset$ q_f est un nouvel état
- $I'' = s(I')$
- $R'' = \{q_g \cdot a \rightarrow a \cdot s(q_d) \text{ telle que } q_g \cdot a \rightarrow a \cdot q_d \in R'\}$

où $s : Q'^{\otimes} \rightarrow (Q' \cup q_f)^{\otimes}$, substitution finie définie par :

- $\forall q \in Q'$ si $q \in \lambda$ alors $s(q) = \{q, q_f\}$
- si $q \notin \lambda$ alors $s(q) = \{q\}$.

II.1.4.6.3. - Démonstration

Immédiate, après avoir montré :

II.1.4.6.3.1. - Lemme

$\forall \delta \in \hat{D}\hat{P}(\Sigma), \forall m_1, \forall m_2 \in Q'^{\otimes}$, on a :

$$m_1 \cdot \delta \Big|_M \delta \cdot m_2 \iff s(m_1) \cdot \delta \Big|_{M'}^* \delta \cdot s(m_2).$$

Il vient :

(a) $\delta \in T(M') \iff \exists m_1 \in I', \exists m_2 \in F'$ tels que $m_1 \cdot \delta \Big|_{M'} \delta \cdot m_2$ en utilisant le lemme.

(b) $\iff \exists m_1 \in I', \exists m_2 \in F'$ tels que $s(m_1) \cdot \delta \Big|_{M''}^* \delta \cdot s(m_2)$ en considérant la définition de s et le fait que $F' = \lambda^{\otimes}$.

(c) $\iff \delta \in T(M'')$.

□

La preuve du lemme est laissé au soin du lecteur. Par exemple, une récurrence sur le nombre de lettres de δ .

II.1.4.7. - Corollaire

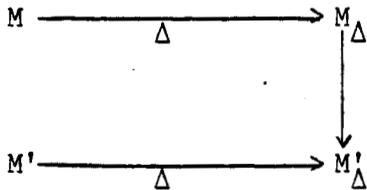
$\forall M = \langle \Sigma, Q, I, F, R \rangle$, automate de dags planaires, il existe effectivement $M' = \langle \Sigma, Q', I', F', R' \rangle$ tel que

- $I' = \{q_i\}^{\otimes}, q_i \in Q'$

- $T(M) = T(M')$

II.1.4.7.1. - Preuve du corollaire

En notant Δ , l'opérateur qui transforme un automate descendant (resp. ascendant) en un automate ascendant (resp. descendant) dual [cf. II.1.3.], on peut écrire :



en utilisant la construction du lemme précédent, mais en "réduisant" le rationnel I dans M_{Δ} .

et le contrôle haut de M' est égal à $\{q_i\}^{\otimes}$. □

II.1.4.8. - Conclusion

En introduisant un indéterminisme important, on peut toujours réduire un des deux contrôles, haut ou bas, à un état, c'est-à-dire $F = \{q_f\}^{\otimes}$ ou $I = \{q_i\}^{\otimes}$.

II.1.5. - Automates d'arbres et automates de dags planaires

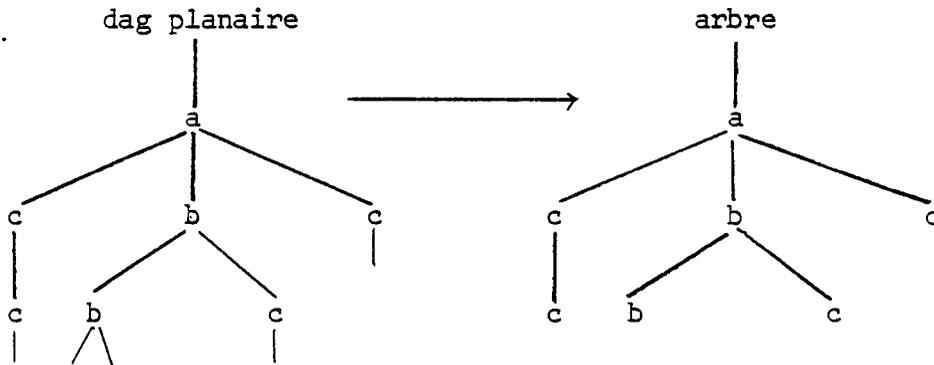
Nous allons montrer qu'aux notations près, la définition des automates de dags planaires prolonge celle des automates d'arbre. Ainsi, «une forêt est reconnaissable si et seulement si elle l'est en tant que taillis sur un alphabet d'arbres ($\Sigma = \Sigma^1$)». Pour cela, on montre qu'au formalisme près :

- Un dag planaire sur un alphabet d'arbres s'identifie à un arbre, et réciproquement.
- Un automate de dags planaires sur un alphabet d'arbres s'identifie à un automate d'arbres, et réciproquement.

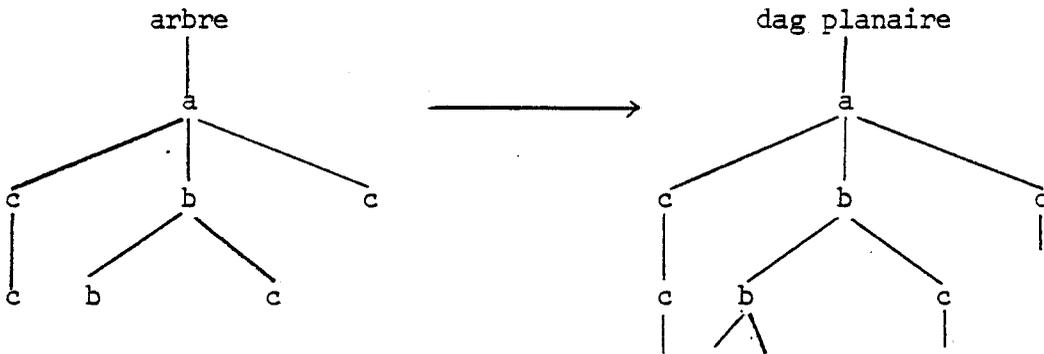
II.1.5.1. - Arbres et dags planaires

La différence de formalisme concerne le degré (ou arité) des lettres du feuillage.

a) Pour identifier un dag planaire sur un alphabet d'arbres à un arbre, on attribuera l'arité 0 aux lettres du feuillage de ce dag.



b) Et d'une manière similaire, pour identifier un arbre à un dag planaire construit sur un alphabet d'arbres, on attribuera l'arité 1 aux lettres du feuillage de cet arbre.



Conclusion :

Ainsi, l'identification est faite à l'arité près des noeuds des feuillages

II.1.5.2. - Automates

Il est évident que le fonctionnement des automates de dags planaires se calque sur celui des automates d'arbres. La seule notion introduite pour les dags est celle de "contrôle". Nous allons voir que dans le cas des "arbres", le contrôle haut et le contrôle bas peuvent toujours être supprimés.

a) Comme nous considérons des taillis sur un alphabet d'arbres, donc avec des lettres de *co-arité égale à 1*, les "mots d'états" du contrôle haut seront de longueur 1, donc des états. Ainsi le contrôle haut se réduit à un ensemble d'états, comme dans les arbres.

b) Nous avons vu [cf théorème II.1.4.] que *le contrôle bas* d'un automate de dags planaires peut toujours être réduit à *un état final*, ceci en modifiant le contrôle haut et les règles de l'automate, ce qui nous renvoie aux considérations ci-dessus. Et donc le contrôle bas d'un automate de dags peut toujours se réduire à la présence ou à l'absence d'un état, comme dans les arbres.

Conclusion :

Ainsi, moyennant la suppression, par remontée, du contrôle bas, le lecteur vérifiera que les automates de dags planaires sur un alphabet d'arbres s'identifie à un automate d'arbres, et réciproquement.

II.1.6. - Automates déterministes

Comme toujours, la sous-classe déterministe d'une classe de machines considérée, en l'occurrence celle des automates de dags planaires, est obtenue en imposant des restrictions syntaxiques excluant le choix dans les calculs. Précédemment, [II.1.1.] et [II.1.4.], nous avons donné deux définitions équivalentes pour la classe des automates. En imposant le déterministe, suivant le principe de restriction rappelée ci-dessus, à l'une ou l'autre définition, on obtient deux définitions différentes des automates déterministes :

- avec un contrôle bas quelconque (langage rationnel),
- sans contrôle bas, au plus exactement réduit à un état final par remontée de ce contrôle [théorème II.1.4.].

Mais en remontant le contrôle bas afin de le réduire à un état final, il est clair [cf. construction en II.1.4.3. et II.1.4.4.] que nous introduisons un non-déterminisme dans les règles, d'où la restriction de ces deux définitions au cas déterministe vont a priori donner deux classes différentes de taillis. On définit le déterminisme fort et le déterminisme faible.

II.1.6.1. - Définitions

II.1.6.1.1. - Déterminisme faible

Un automate descendant de dags planaire $M = \langle \Sigma, Q, I, F, R \rangle$ est faiblement déterministe si et seulement si :

- $I \subseteq \{q_0\}^*$, $q_0 \in Q$ (un seul état peut participer à la définition de I)
- R est une application partielle de $Q^* \cdot \Sigma \rightarrow \Sigma \cdot Q^*$.

II.1.6.2.2. - Déterminisme fort

Un automate descendant de dags planaires $M = \langle \Sigma, Q, I, F, R \rangle$ est fortement déterministe si et seulement si :

- il est faiblement déterministe

- $F = Q_f^{\otimes}$, $Q_f \subseteq Q$

Remarque :

Le cas ascendant est défini en échangeant les restrictions entre I et F.

II.1.6.2. - Critiques des définitions sur le déterminisme

Nous avons rappelé en début de [II.1.6.] que la sous-classe des automates déterministe est obtenue en imposant des restrictions syntaxiques excluant le choix dans les calculs, c'est-à-dire tout au long du processus de reconnaissance. Ainsi :

II.1.6.2.1. - Restrictions sur le contrôle haut

La nature de l'objet à reconnaître doit induire l'unique tenseur d'états avec lequel il faut commencer la reconnaissance, ce qui explique que $I = \{q_0\}^{\otimes}$. Un contrôle haut faisant intervenir plus d'un état introduirait un indéterminisme.

II.1.6.2.2. - Restrictions sur les règles de reconnaissance

Le choix de la règle à utiliser pour reconnaître une lettre doit être unique ou ne pas exister, c'est pourquoi R est une application partielle.

II.1.6.2.3. - Restrictions sur le contrôle bas

La décision d'accepter ou de ne pas accepter le tenseur d'états doit être aussi déterministe, on a le *déterminisme fort* où cette décision se restreint à l'appartenance, de chacun des états du tenseur d'états obtenu, à un ensemble d'états finaux ; le *déterminisme faible* où cette décision est soumise à l'appartenance, du tenseur d'états obtenu à un langage rationnel, cette condition se justifiant en considérant le magmaïde comme une extension de la structure de monoïde et par la possibilité de pouvoir toujours réduire le non-déterminisme dans les automates d'états finis de mots.

II.1.6.3. - Théorème de non-réductibilité du déterminisme fort

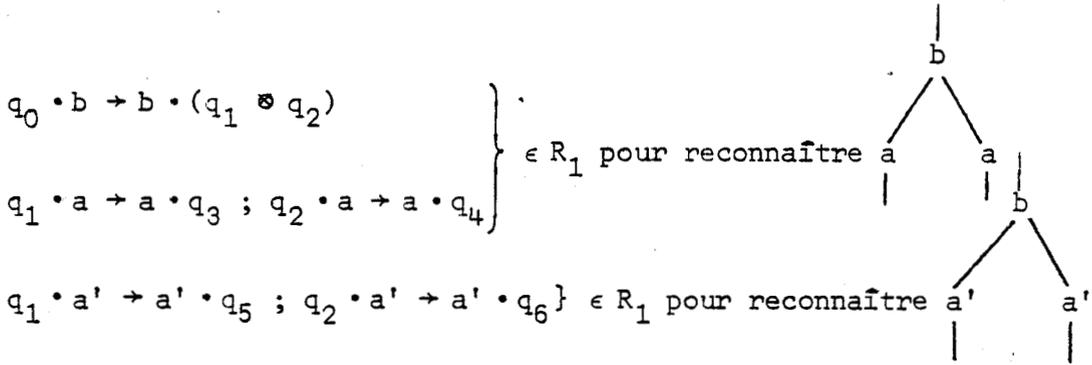
Le théorème suivant montre qu'il existe des taillis reconnaissables (et même finis) qui ne peuvent être reconnus par un automate fortement déterministe, qu'il soit ascendant ou descendant. L'idée est de partir du phénomène bien connu dans les arbres de non-réductibilité du non-déterminisme (qui correspond ici au déterminisme fort) de l'automate descendant et par dualité de l'étendre au cas ascendant.

II.1.6.3.1. - Énoncé

Il existe des taillis reconnaissables qui ne peuvent être reconnus par un automate fortement déterministe.

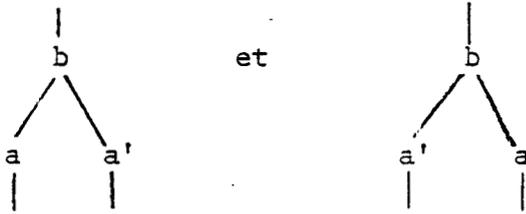
II.1.6.3.2. - Preuve

a) Le taillis reconnaissable $T_1 = \left\{ \begin{array}{c} | \\ b \\ / \quad \backslash \\ a \quad a \\ | \quad | \\ a \quad a \end{array} \right\}, \left\{ \begin{array}{c} | \\ b \\ / \quad \backslash \\ a' \quad q' \\ | \quad | \\ a' \quad q' \end{array} \right\}$ ne peut être reconnu par un automate, $M_1 = \langle \Sigma_1, Q_1, T_1, F_1, R_1 \rangle$ fortement déterministe et descendant, puisque :



avec $I_1 = \{q_0\}$ et $F_1 = \{q_3, q_4, q_5, q_6\}^{\otimes}$, puisque M_1 fortement déterministe.

Mais avec ces conditions nécessaires, M_1 reconnaît en plus



b) de la même manière, on montre que le taillis reconnaissable $T_2 = \left\{ \begin{array}{c} a \quad a, \quad a' \quad a' \\ \backslash \quad / \quad \backslash \quad / \\ \quad b \quad \quad b \end{array} \right\}$ ne peut être reconnu par un automate ascendant fortement déterministe.



c) et le taillis reconnaissable $T_1 \cup T_2$ ne peut être reconnu par un automate fortement déterministe qu'il soit ascendant ou descendant.

□

II.1.6.4. - Théorème de non-réductibilité du déterminisme faible

La classe des taillis reconnus par les automates faiblement déterministes est beaucoup vaste que celle reconnue par les automates fortement déterministes, notamment elle comprend la sous-classe des taillis finis. Cependant, cette classe de taillis n'est pas aussi riche que celle des automates non-déterministes. Le théorème suivant montre qu'il existe des taillis reconnaissables qui ne peuvent être reconnu par un automate faiblement déterministe.

II.1.6.4.1. - Énoncé

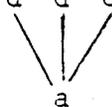
Il existe des taillis reconnaissables qui ne peuvent être reconnus par un automate faiblement déterministe.

II.1.6.4.2. - Preuve

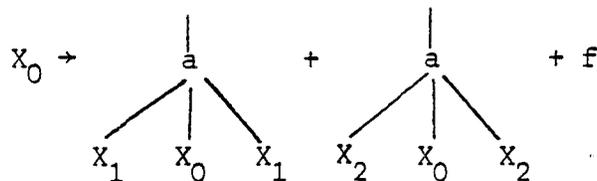
Nous allons montrer qu'il existe des forêts reconnaissables qu'on ne peut reconnaître par un automate descendant déterministe. Ainsi si T est cette forêt alors $T \cup T'$ n'est pas un taillis reconnaissable si T' est le taillis composé des arbres de T mais pris dans l'autre sens, i.e. si $a \in T \Leftrightarrow$



$d \ d \ d \in T'$. Prenons $T = L(G, X_0)$ où G est la grammaire régulière d'arbres



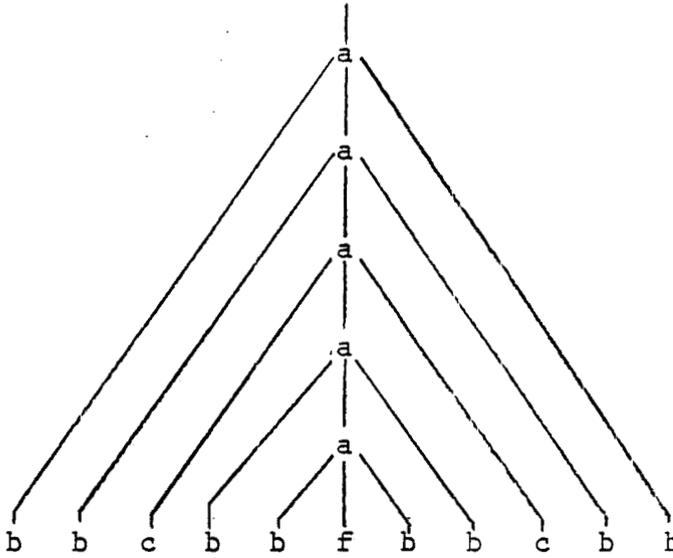
définie par :



$X_1 \rightarrow b$

$X_2 \rightarrow c$

Les arbres engendrés par cette grammaire sont de la forme :

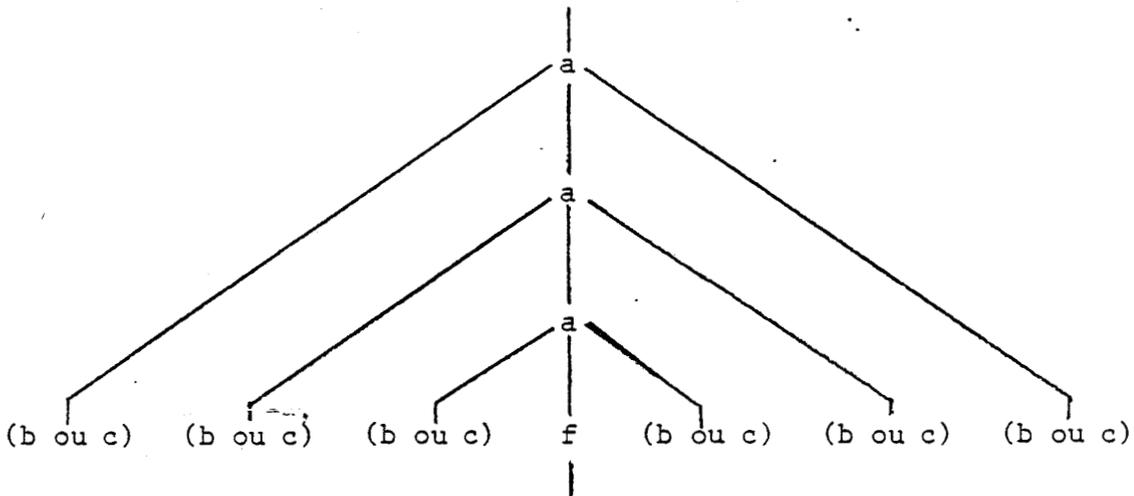


et $\Phi(T) = \{\omega \in \{b, c, f\}^* / \omega = \omega_1 f \tilde{\omega}_1\}$, $\tilde{\omega}_1$ désignant le mot miroir de ω_1 et ϕ , l'application de feuillage.

II.1.6.4.3. - Idée de démonstration

En montrant que la construction de $M = \langle \Sigma, Q, I, F, R \rangle$, automate descendant faiblement déterministe, tel que $T(M) = \bar{T}$ est impossible. Plus précisément, on procède en trois temps.

a) Dans un premier temps, on montre que si M est fortement déterministe alors $T \subseteq T(M)$ implique que M reconnaît les arbres de la forme :



b) puis on montre que étendre M en un automate faiblement déterministe tel que M reconnaisse exactement la forêt T, revient à trouver un langage de mots, F, sur $\{b, c, f\}^*$ tel que :

- F soit rationnel

- $F = \{\omega \in \{b, c, f\}^* / \omega = \omega_1 f \tilde{\omega}_1\}$

c) enfin, on montre qu'un tel langage n'existe pas. Et donc qu'on ne peut construire M, faiblement déterministe tel que M reconnaisse T.

II.1.6.4.4. - Démonstration

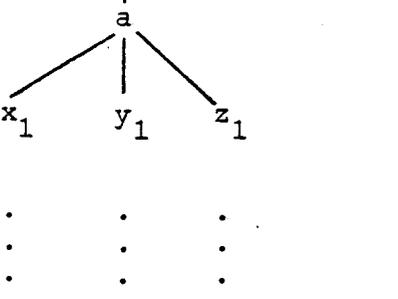
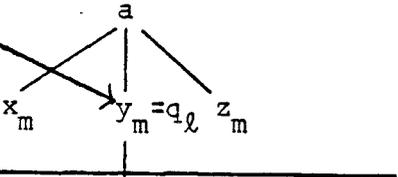
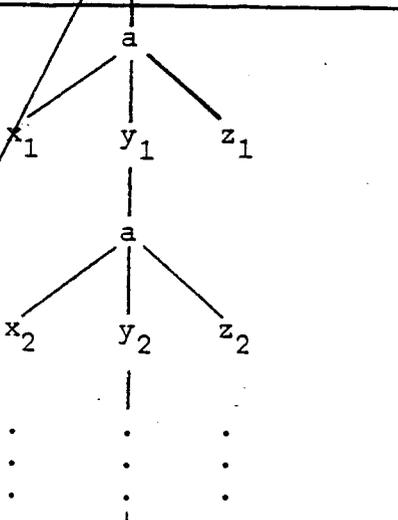
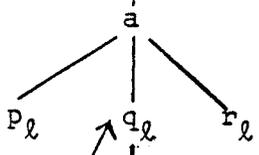
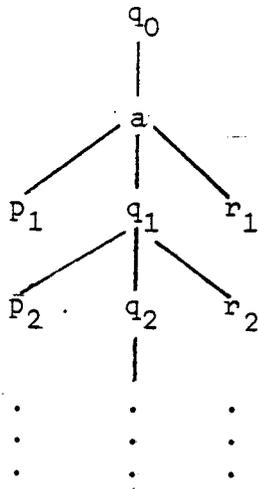
Essayons de construire $M = \langle \Sigma, Q, I, F, R \rangle$, automate descendant faiblement déterministe tel que $T(M) = T$.

a) si M existe alors $M' = \langle \Sigma, Q, I, Q^*, R \rangle$ est un automate fortement déterministe que l'on peut construire. Les restrictions imposées par le déterminisme faible et la structure de l'ensemble de dags à reconnaître, notamment qu'il soient tous d'une co-arité égale à 1, impose :

- $I = \{q_0\}$

- Les seules reconnaissances admises par M' sont de la forme :

et



= y_m

$$p_i \cdot b \rightarrow b \cdot p_i'$$

$$p_i \cdot c \rightarrow c \cdot p_i'' \text{ pour } i = 1, \dots, l.$$

$$r_i \cdot b \rightarrow b \cdot r_i'$$

$$r_i \cdot c \rightarrow c \cdot r_i'' \text{ pour } i = 1, \dots, l.$$

et aussi

$$x_i \cdot b \rightarrow b \cdot x_i'$$

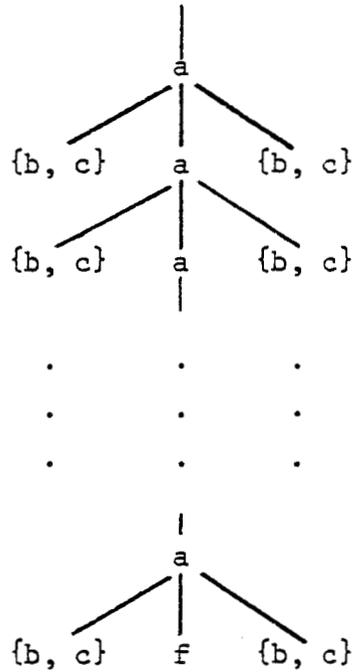
$$x_i \cdot c \rightarrow c \cdot x_i'' \text{ pour } i = 1, \dots, m.$$

$$z_i \cdot b \rightarrow b \cdot z_i'$$

$$z_i \cdot c \rightarrow c \cdot z_i'' \text{ pour } i = 1, \dots, m.$$

$$y_m \cdot f \rightarrow f \cdot \bar{y}$$

Les dags reconnus par M' sont les arbres de la forme :



Les feuillage en fin de reconnaissance sont de la forme :

$$(p_1' + p_1'') (p_2' + p_2'') \dots (p_\ell' + p_\ell'') [(x_1' + x_1'') \dots \dots (x_m' + x_m'')]^n \bar{y}$$

$$[(z_m' + z_m'') \dots \dots (z_1' + z_1'')]^n (r_\ell' + r_\ell'') \dots (r_1' + r_1'')$$

pour $n \in \mathbb{N}$.

Remarque :

Il est facile de vérifier que les éléments de construction proposée jusqu'à présent sont nécessaires puisque l'automate doit être fortement déterministe.

b) si l'on note Q' (resp. Q'') l'ensemble des états engendrés après la reconnaissance d'une lettre b (resp. d'une lettre c). Il est clair que l'existence de l'automate M' faiblement déterministe, est liée aux conditions :

- $Q' \cap Q'' = \emptyset$
- il existe un langage F sur Q^{\otimes} tel que
 - F soit rationnel
 - deux états à distances égales du milieu \bar{y}_m sont : soient tous deux éléments de Q' , soient tous deux éléments de Q'' .

Ainsi en d'autres termes l'existence de M faiblement déterministe est liée à l'existence d'un langage F' sur $\{b, c, f\}^{\otimes}$ tel que :

- F' soit rationnel
- $F' = \{\omega \in \{b, c, f\}^{\otimes} / \omega = \omega_1 f \tilde{\omega}_1\}$ où $\tilde{\omega}_1$ est le mot miroir de ω_1 .

c) F' ne peut être rationnel, est un résultat classique de la théorie des langages, et ainsi il n'existe pas M descendant, faiblement déterministe tel que $T = T(M)$.

□

II.1.6.4.5. - Remarque

Le fait d'introduire un contrôle bas rationnel dans les reconnaissances d'arbres (définition du déterminisme faible), permet d'étendre la sous-famille des forêts reconnaissables reconnue par les automates déterministes. L'exemple qui le met en évidence [II.1.6.3.] est une forêt finie. Nous allons voir, par l'exemple ci-dessous, que des forêts beaucoup plus "compliquées" font partie de cette extension, i.e. qu'elles peuvent être reconnues par un automate faiblement déterministe mais non par un automate fortement déterministe.

Exemple :

Soit l'automate d'arbre non déterministe suivant :

$$M = \langle \Sigma, Q, Q_0, Q_f, R \rangle \text{ où}$$

$$\Sigma = \Sigma_2^1 = \{b, b'\}$$

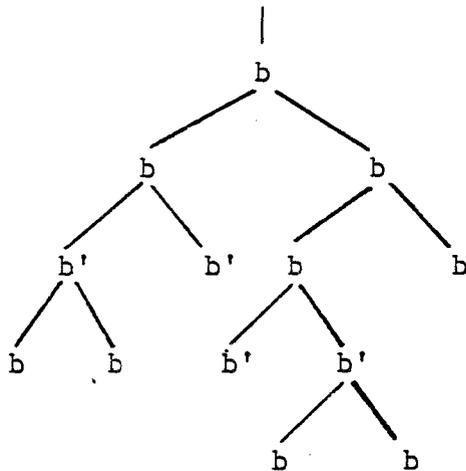
$$Q = \{q_0, q_1\}$$

$$Q_0 = \{q_0, q_1\} ; Q_f = \{q_0, q_1\}$$

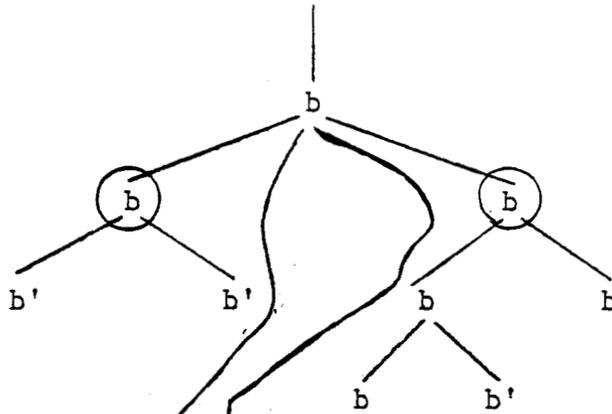
$$R = \{q_0 \cdot b \vdash b \cdot (q_0 \otimes q_0) ; q_0 \cdot b \vdash b \cdot (q_1 \otimes q_1) ;$$

$$q_1 \cdot b' \vdash b' \cdot (q_0 \otimes q_0) ; q_1 \cdot b' \vdash b' \cdot (q_1 \otimes q_1) \} .$$

Les arbres reconnus sont ceux tels que les successeurs de toute lettre soient deux lettres identiques (b ou b') :



Nous allons construire un automate d'arbre faiblement déterministe (avec contrôle bas) qui reconnaît cette forêt. La vérification de l'égalité des successeurs directs d'un noeud aura lieu sur le mot d'état obtenu en bout de reconnaissance. On transportera à ces fins des marques caractéristiques des lettres qui ont été rencontrées selon le principe suivant :



transport le long de cette arête d'une marque indiquant que l'on a rencontré un b comme successeur droit

transport le long de cette arête d'une marque indiquant que l'on a rencontré un b comme successeur gauche

vérification dans le mot final que ces deux marques se suivent bien.

Dans l'automate que nous allons construire, les marques auront les significations suivantes :

-) : b comme successeur droit
- (: b comme successeur gauche
-] : b' comme successeur droit
- [: b' comme successeur gauche

Les arêtes droites seront repérées par un état q^d porteur d'une marque et les arêtes gauches seront repérées par un état q^g porteur d'une marque.

L'automate M' descendant et faiblement déterministe qui reconnaît la forêt considérée sera donc :

$$M' = \langle \Sigma, Q, I, F, R \rangle \text{ où}$$

$$\Sigma = \Sigma_2^1 = \{b, b'\}$$

$$Q = \{q, q^g, q^d, q_{\leftarrow}^g, q_{\leftarrow}^d, q_{\rightarrow}^g, q_{\rightarrow}^d\}$$

$$I = \{q\}$$

$$F = q^g \otimes (q_{\leftarrow}^d q_{\rightarrow}^g + q_{\rightarrow}^d q_{\leftarrow}^g) \otimes q^d$$

$$R = \{q \cdot b \vdash b \cdot (q^g \otimes q^d) ; q \cdot b' \vdash b' \cdot (q^g \otimes q^d) ;$$

$$q^g \cdot b \vdash b \cdot (q^g \otimes q_{\leftarrow}^d) ; q^g \cdot b' \vdash b' \cdot (q^g \otimes q_{\leftarrow}^d) ;$$

$$q^d \cdot b \vdash b \cdot (q_{\rightarrow}^g \otimes q^d) ; q^d \cdot b' \vdash b' \cdot (q_{\rightarrow}^g \otimes q^d) ;$$

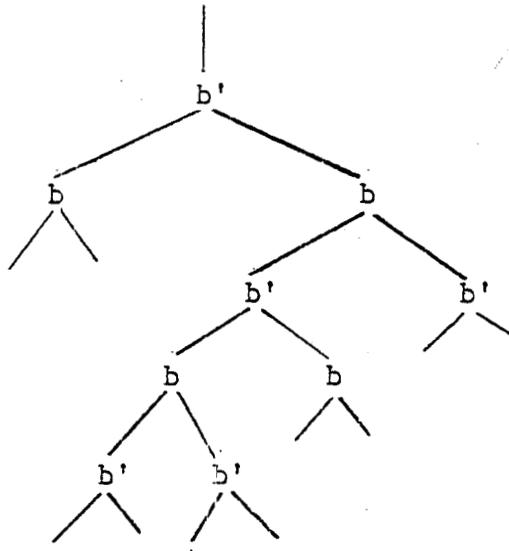
$$q_{\leftarrow}^g \cdot b \vdash b \cdot (q_{\leftarrow}^g \otimes q_{\leftarrow}^d) ; q_{\leftarrow}^g \cdot b' \vdash b' \cdot (q_{\leftarrow}^g \otimes q_{\leftarrow}^d) ;$$

$$q_{\leftarrow}^d \cdot b \vdash b \cdot (q_{\leftarrow}^g \otimes q_{\leftarrow}^d) ; q_{\leftarrow}^d \cdot b' \vdash b' \cdot (q_{\leftarrow}^g \otimes q_{\leftarrow}^d) ;$$

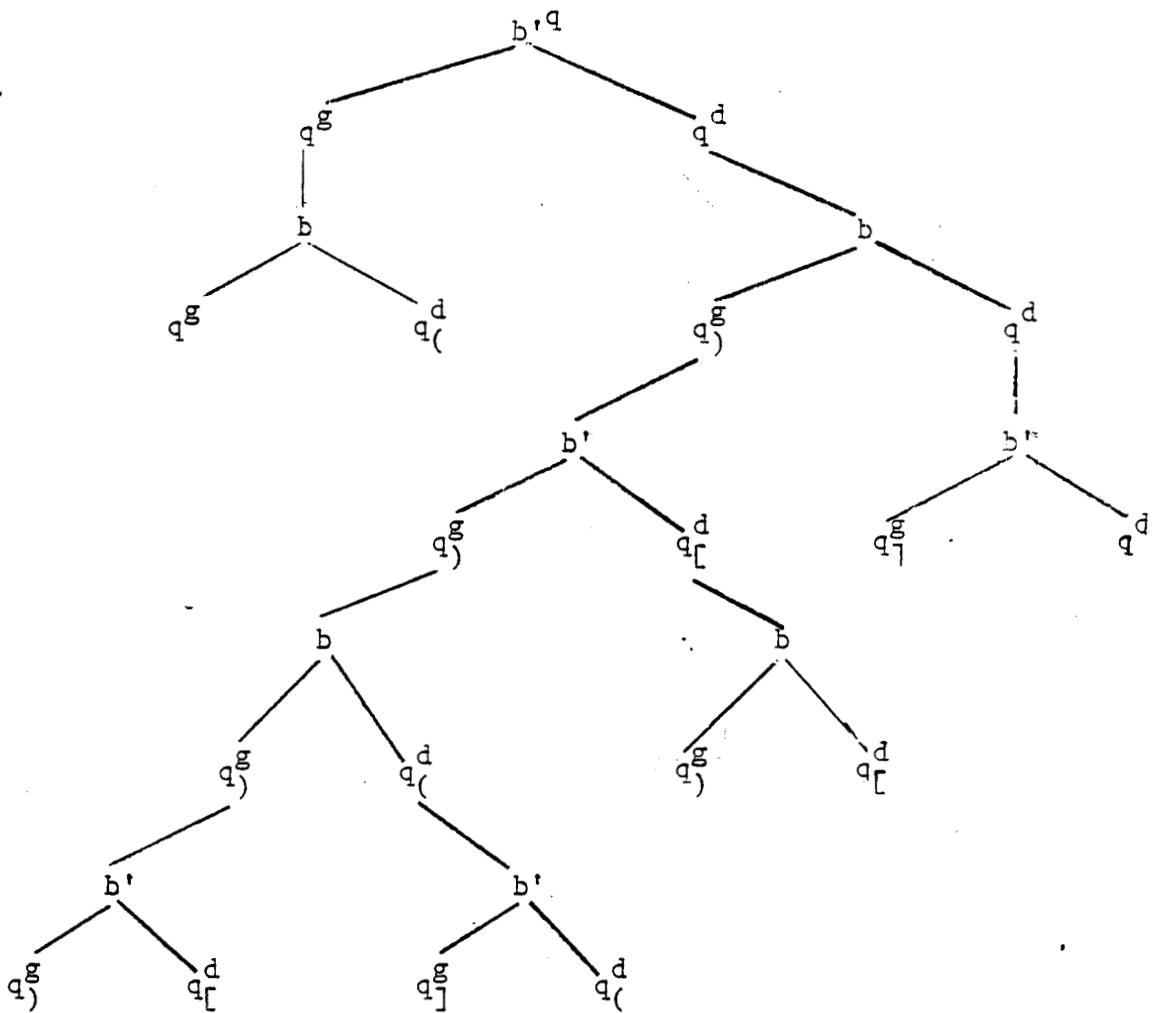
$$q_{\rightarrow}^g \cdot b \vdash b \cdot (q_{\rightarrow}^g \otimes q^d) ; q_{\rightarrow}^g \cdot b' \vdash b' \cdot (q_{\rightarrow}^g \otimes q^d) ;$$

$$q_{\rightarrow}^d \cdot b \vdash b \cdot (q_{\rightarrow}^g \otimes q_{\leftarrow}^d) ; q_{\rightarrow}^d \cdot b' \vdash b' \cdot (q_{\rightarrow}^g \otimes q_{\leftarrow}^d) \}$$

Exemple de reconnaissance d'un élément de la forêt :

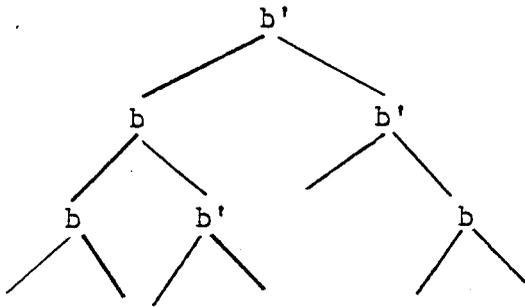


donne la reconnaissance suivante :

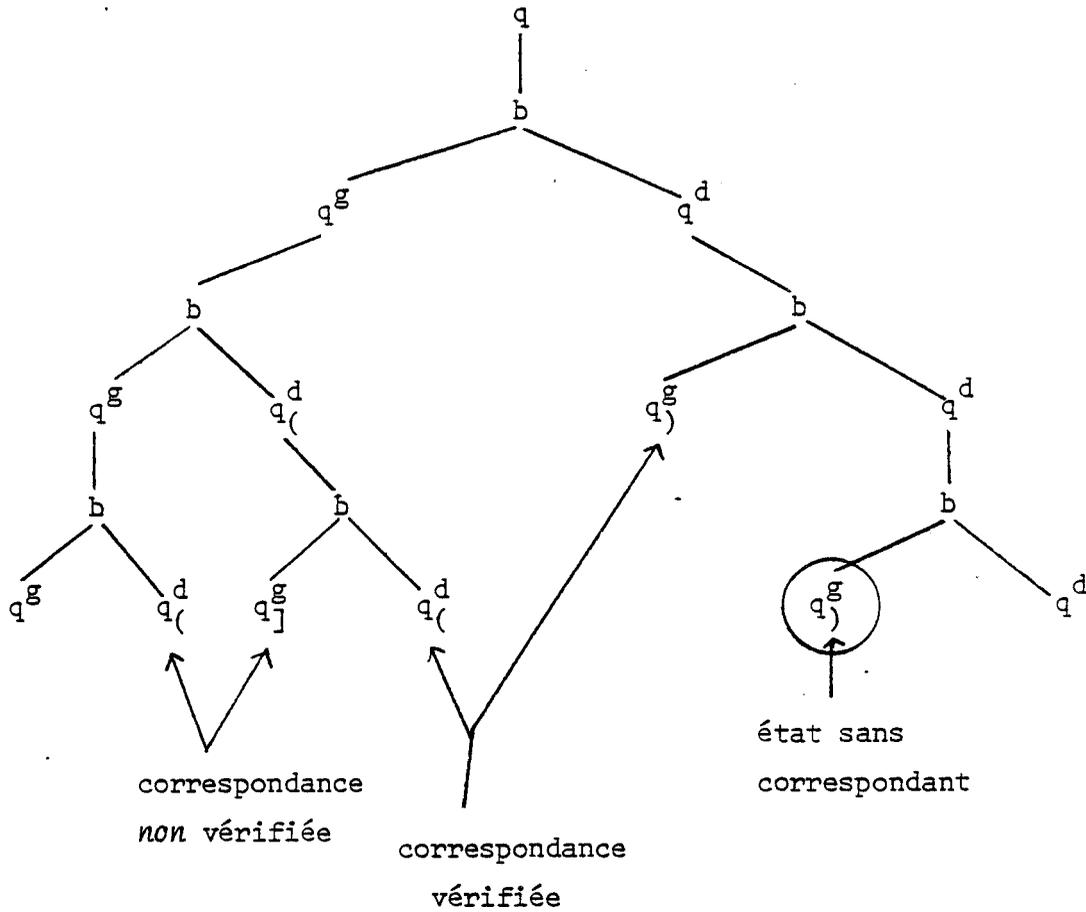


et le mot d'état obtenu appartient bien à F.

Exemple de non-reconnaissance d'un élément hors de la forêt



donne la reconnaissance suivante :



II.2. - PROPRIETES DE CLOTURE SUR LES ENSEMBLES RECONNAISSABLES DE DAGS

Le but de cette partie est de mettre en évidence (et de démontrer) les propriétés de clôture ou de non-clôture des ensembles reconnaissables de dags planaires par les opérations usuelles comme l'union, l'intersection... ou plus spécifiques du magmaïde comme le produit tensoriel, le produit de composition... Les résultats obtenus, sans respect de l'ordre de présentation dans les pages suivantes, sont :

a) Résultats de clôture par

- Union
- Intersection
- Produit tensoriel
- Etoile du produit tensoriel .
- Produit de composition
- Homomorphisme inverse
- Homomorphisme convexe.

b) Résultats de "non-clôture"

- Etoile du produit de composition
- Homomorphisme.

c) Problème ouvert

- Le complémentaire.

Les différentes constructions et démonstrations mettrons en évidence un aspect fondamental des dags planaires : la CONNEXITE.

II.2.1. - Clôture par unionII.2.1.1. - Propriété

REC est clos par union.

II.2.1.2. - Preuve

$\forall D_1, \forall D_2 \in \text{REC}$, il existe effectivement un automate M_3 tel que $T(M_3) = D_1 \cup D_2$.

II.2.1.3. - Construction

$D_1 \in \text{REC} \Rightarrow \exists M_1 = \langle \Sigma_1, Q_1, I_1, F_1, R_1 \rangle$ avec $T(M_1) = D_1$,
 $D_2 \in \text{REC} \Rightarrow \exists M_2 = \langle \Sigma_2, Q_2, I_2, F_2, R_2 \rangle$ avec $T(M_2) = D_2$. On peut supposer $Q_1 \cap Q_2 = \emptyset$ et ainsi $M_3 = \langle \Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2, I_1 \cup I_2, F_1 \cup F_2, R_1 \cup R_2 \rangle$.

II.2.1.4. - Justification informelle de la construction

- L'alphabet (respectivement les états, les règles) de M_3 est défini comme l'union (respectivement des états, des règles) de M_1 et $M_2 \Rightarrow$ on a les mêmes réécritures dans M_3 que dans M_1 et M_2 réunis.

- De plus, le contrôle haut (resp. bas) de M_3 est défini comme l'union des contrôles hauts (resp. bas) de M_1 et $M_2 \Rightarrow T(M_3) \subseteq T(M_1) \cup T(M_2)$.

- Supposer $Q_1 \cap Q_2 = \emptyset$ permet d'éviter que les réécritures ne se mélangent ou n'interfèrent entre-elles et assure $T(M_3) = T(M_1) \cup T(M_2)$, i.e. $T(M_3) = D_1 \cup D_2$.

La justification formelle est évidente.

□

II.2.2. - Clôture par intersection .

Préliminaire :

1. L'opérateur Sh

Soient X^* et Y^* les monoïdes libres engendrés respectivement par l'alphabet X et Y , on définit l'application partielle $Sh : X^* \times Y^* \rightarrow (X \times Y)^*$ par :

$\forall u \in X^*$ où $u = u_1 u_2 \dots u_n$ avec $u_i \in X$ pour $i = 1, \dots, n$

$\forall v \in Y^*$ où $v = v_1 v_2 \dots v_{n'}$, avec $v_i \in Y$ pour $i = 1, \dots, n'$.

Si $n \neq n'$ alors

$Sh(u, v)$ n'est pas défini

sinon

$$Sh(u, v) = (u_1, v_1) (u_2, v_2) \dots (u_n, v_n).$$

On étend $Sh : P(X^*) \times P(Y^*) \rightarrow P((X \times Y)^*)$ par $\forall U \subseteq X^*, \forall V \subseteq Y^*$ alors $Sh(U, V) = \{Sh(u, v) / u \in V \text{ et } v \in U\}$.

L'extension de Sh pour un ensemble fini $X_1^*, X_2^*, \dots, X_m^*$ de monoïde libre est laissé aux soins du lecteur.

2. La projection Π

Soit $Q_1 \times Q_2 \times \dots \times Q_n$ un ensemble produit d'ensemble, on notera Π_{Q_i} la projection de $Q_1 \times Q_2 \times \dots \times Q_n$ sur Q_i .

$$\Pi_{Q_i} : Q_1 \times Q_2 \times \dots \times Q_n \rightsquigarrow Q_i$$

$$(q_1, q_2, \dots, q_n) \rightarrow q_i$$

et par extension, on notera $\Pi_{Q_{i_1} \times Q_{i_2} \times \dots \times Q_{i_n}}$ l'application de Q_1

$$Q_1 \times Q_2 \times \dots \times Q_n \rightarrow Q_{i_1} \times Q_{i_2} \times \dots \times Q_{i_n}$$

$$\omega \rightsquigarrow (\Pi_{Q_{i_1}}(\omega), \Pi_{Q_{i_2}}(\omega), \dots, \Pi_{Q_{i_n}}(\omega))$$

II.2.2.1. -- Propriété

REC est clos par intersection.

II.2.2.2. -- Preuve

$\forall D_1, \forall D_2 \in \text{REC}$, il existe effectivement un automate M_3 tel que $T(M_3) = D_1 \cap D_2$.

II.2.2.3. -- Construction

$D_1 \in \text{REC} \Rightarrow \exists M_1 = \langle \Sigma_1, Q_1, I_1, F_1, R_1 \rangle$ avec $T(M_1) = D_1$,
 $D_2 \in \text{REC} \Rightarrow \exists M_2 = \langle \Sigma_2, Q_2, I_2, F_2, R_2 \rangle$ avec $T(M_2) = D_2$, et ainsi $M_3 = \langle \Sigma_1 \cap \Sigma_2, Q_1 \times Q_2, \text{Sh}(I_1, I_2), \text{Sh}(F_1, F_2), R_3 \rangle$, où Sh est l'opération définie ci-dessus.

R_3 définit par :

$$c \in \Sigma_{1,n}^n \cap \Sigma_{2,n}^n$$

$$\left. \begin{array}{l} (q_1 \otimes \dots \otimes q_n) \cdot c + c \cdot (q'_1 \otimes \dots \otimes q'_n) \in R_1 \\ (\bar{q}_1 \otimes \dots \otimes \bar{q}_n) \cdot c + c \cdot (\bar{q}'_1 \otimes \dots \otimes \bar{q}'_n) \in R_2 \end{array} \right| \Leftrightarrow$$

$$(q_1, \bar{q}_1) \otimes \dots \otimes (q_n, \bar{q}_n) \cdot c + c \cdot (q'_1, \bar{q}'_1) \otimes \dots \otimes (q'_n, \bar{q}'_n) \in R_3.$$

II.2.2.4. -- Justification informelle de la construction

- La définition des états et des règles de $M_3 \Rightarrow$ la reconnaissance d'une lettre peut se faire par M_3 si et seulement si elle peut se faire à la fois par M_1 et par M_2 .

- De plus, la définition du contrôle haut (resp. bas) de M_3 comme résultat par Sh des contrôles haut (resp. bas) de M_1 et M_2 permet à ce contrôle haut (resp. bas) de prendre en compte en même temps les contrôles haut (resp. bas) de M_1 et de M_2 et $\Rightarrow T(M_3) = T(M_1) \cap T(M_2)$.

II.2.2.5. - DémonstrationLemme :

$\forall m_1, \forall m'_1 \in Q_1^{\otimes} ; \forall m_2, \forall m'_2 \in Q_2^{\otimes} ; \forall \omega \in \hat{D}P(\Sigma_1 \cap \Sigma_2)$ on a : $m_1 \cdot \omega \mid_{M_1}^* \omega \cdot m'_1$
 et $m_2 \cdot \omega \mid_{M_2}^* \omega \cdot m'_2 \iff \text{Sh}(m_1, m_2) \cdot \omega \mid_{M_3}^* \omega \cdot \text{Sh}(m'_1, m'_2)$.

Ainsi il vient (comme démonstration de la propriété) :

$$\omega \in D_1 \cap D_2 \iff \omega \in D_1 \text{ et } \omega \in D_2$$

$\iff \exists m_1 \in I_1, \exists m_2 \in I_2, \exists m'_1 \in F_1, \exists m'_2 \in F_2$ tels que

$$m_1 \cdot \omega \mid_{M_1}^* \omega \cdot m'_1 \text{ et } m_2 \cdot \omega \mid_{M_2}^* \omega \cdot m'_2$$

en utilisant le lemme,

$\iff \exists m_1 \in I_1, \exists m_2 \in I_2, \exists m'_1 \in F_1, \exists m'_2 \in F_2$ tels que

$$\text{Sh}(m_1, m_2) \cdot \omega \mid_{M_3}^* \omega \cdot \text{Sh}(m'_1, m'_2).$$

en remarquant que $\text{Sh}(I_1, I_2)$ et $\text{Sh}(F_1, F_2)$ sont respectivement le contrôle haut et le contrôle bas de M_3 .

$\iff \omega \in T(M_3)$.

□

Preuve du lemme :

Récurrence sur le nombre, l , d'occurrences de lettres dans ω .

- $l = 0 \implies \omega = e_p, m_1 = m'_1$ et $m_2 = m'_2$. Le lemme se vérifie aisément.

Hypothèse de récurrence :

$\forall r \in \mathbb{N}^*$, le lemme est vérifié pour $l \in \mathbb{N}$ et $l < r$.

$l = r \Rightarrow \omega = \omega_1 \cdot (e_i \otimes a \otimes e_j)$, où le nombre d'occurrences de lettres dans ω_1 est $l-1$,

$$\Rightarrow m_1 \cdot \omega \Big|_{M_1}^* \omega \cdot m_1' \text{ et } m_2 \cdot \omega \Big|_{M_2}^* \omega \cdot m_2'$$

$$\Leftrightarrow m_1 \cdot \omega \Big|_{M_1}^* \omega_1 \cdot m_1'' \cdot (e_i \otimes a \otimes e_j) \Big|_{M_1}^* \omega \cdot m_1' \text{ et}$$

$$m_2 \cdot \omega \Big|_{M_2}^* \omega_1 \cdot m_2'' \cdot (e_i \otimes a \otimes e_j) \Big|_{M_2}^* \omega \cdot m_2'$$

en utilisant l'hypothèse de récurrence

$$\text{Sh}(m_1, m_2) \cdot \omega_1 \Big|_{M_3}^* \omega_1 \cdot \text{Sh}(m_1'', m_2'') \text{ et il existe}$$

$$(q_1 \otimes \dots \otimes q_n) \cdot a \rightarrow a \cdot (q_1' \otimes \dots \otimes q_n') \in R_1 / m_1'' = u \otimes q_1 \otimes \dots \otimes q_n \otimes v \text{ et}$$

$$m_1' = u \otimes q_1' \otimes \dots \otimes q_n', \otimes v$$

$$(\bar{q}_1 \otimes \dots \otimes \bar{q}_n) \cdot a \rightarrow a \cdot (\bar{q}_1' \otimes \dots \otimes \bar{q}_n') \in R_2 / m_2'' = \bar{u} \otimes \bar{q}_1 \otimes \dots \otimes \bar{q}_n \otimes \bar{v} \text{ et}$$

$$m_2' = \bar{u} \otimes \bar{q}_1' \otimes \dots \otimes \bar{q}_n', \otimes \bar{v} \text{ en considérant la construction de } R_3,$$

$$\Leftrightarrow \text{Sh}(m_1, m_2) \cdot \omega_1 \Big|_{M_3}^* \omega_1 \cdot \text{Sh}(m_1'', m_2'') \text{ et } \text{Sh}(m_1'', m_2'') \cdot (e_i \otimes a \otimes e_j) \Big|_{M_3}^*$$

$$(e_i \otimes a \otimes e_j) \cdot \text{Sh}(m_1', m_2').$$

$$\Leftrightarrow \text{Sh}(m_1, m_2) \cdot \omega \Big|_{M_3}^* \omega \cdot \text{Sh}(m_1', m_2').$$

□

II.2.3. - Clôture par produit tensoriel

II.2.3.1. - Propriété

REC est clos par produit tensoriel.

II.2.3.2. - Preuve

$\forall D_1, \forall D_2 \in \text{REC}$, il existe effectivement un automate M_3 tel que $T(M_3) = D_1 \otimes D_2$.

II.2.3.3. - Construction

$D_1 \in \text{REC} \Rightarrow \exists M_1 = \langle \Sigma_1, Q_1, I_1, F_1, R_1 \rangle$ avec $T(M_1) = D_1$.

$D_2 \in \text{REC} \Rightarrow \exists M_2 = \langle \Sigma_2, Q_2, I_2, F_2, R_2 \rangle$ avec $T(M_2) = D_2$. On peut supposer

$Q_1 \cap Q_2 = \emptyset$ et ainsi $M_3 = \langle \Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2, I_1 \otimes I_2, F_1 \otimes F_2, R_1 \cup R_2 \rangle$.

II.2.3.4. - Justification informelle de la construction

- L'alphabet (resp. les états, les règles) de M_3 est défini comme l'union des alphabets (resp. des états, des règles) de M_1 et $M_2 \Rightarrow$ on a les mêmes réécritures dans M_3 que dans M_1 et M_2 réunis.

- de plus, le contrôle haut (resp. bas) de M_3 est le produit tensoriel des contrôles hauts (resp. bas) de M_1 et M_2 et donc $\Rightarrow D_1 \otimes D_2 \subseteq T(M_3)$.

- supposer $Q_1 \cap Q_2 = \emptyset$ permet d'éviter que les réécritures ne se mélangent ou n'interfèrent entre-elles et assure $T(M_3) = T(M_1) \otimes T(M_2)$, i.e. $T(M_3) = D_1 \otimes D_2$.

La justification formelle est évidente.

□

II.2.4. - Clôture par étoile tensorielle

II.2.4.1. - Propriété

REC est clos par étoile tensorielle.

II.2.4.2. - Preuve

$\forall D \in \text{REC}$, il existe effectivement un automate M' tel que $T(M') = D^{\otimes}$.

II.2.4.3. - Notations

Notons $M = \langle \Sigma, Q, I, F, R \rangle$ l'automate qui reconnaît D et $M' = \langle \Sigma, Q', I', F', R' \rangle$ l'automate que nous cherchons à construire tel que $T(M') = D^{\otimes}$.

II.2.4.4. - Idée de construction

Il faut que $T(M') = D^{\otimes} = \bigcup_{i \in \mathbb{N}} D^i$ ou encore

$D^{\otimes} = \{ \delta \in \text{DP}(\Sigma) \mid \delta = \delta_1 \otimes \dots \otimes \delta_n \text{ où } \delta_i \in D, \text{ pour } i = 1, \dots, n \}$.

C'est-à-dire que toute reconnaissance dans M' implique et réciproquement plusieurs reconnaissances indépendantes de dags $\delta_i \in D$. La construction que nous proposons peut s'expliquer en deux principes.

Premier principe de la construction de l'automate M' :

- On garde les mêmes règles que l'automate M
- le contrôle haut devient I^{\otimes}
- le contrôle bas devient F^{\otimes}

Ceci dans l'espoir d'avoir $\delta \in T(M')$ si et seulement si

- $\delta = \delta_1 \otimes \dots \otimes \delta_n$ où $\delta_i \in D$ pour $i = 1, \dots, n$
- et δ reconnu par une reconnaissance schématisée :

$m_1 \in I$	$m_2 \in I$...	$m_n \in I$
δ_1	δ_2	...	δ_n
$m'_1 \in F$	$m'_2 \in F$...	$m'_n \in F$

mais ce seul principe de construction, n'assure pas que les reconnaissances des δ_i , au lieu de rester indépendantes, n'interfèrent entre-elles et aboutissent à reconnaître des dags qui ne sont pas éléments de D^{\otimes} .

Exemple :

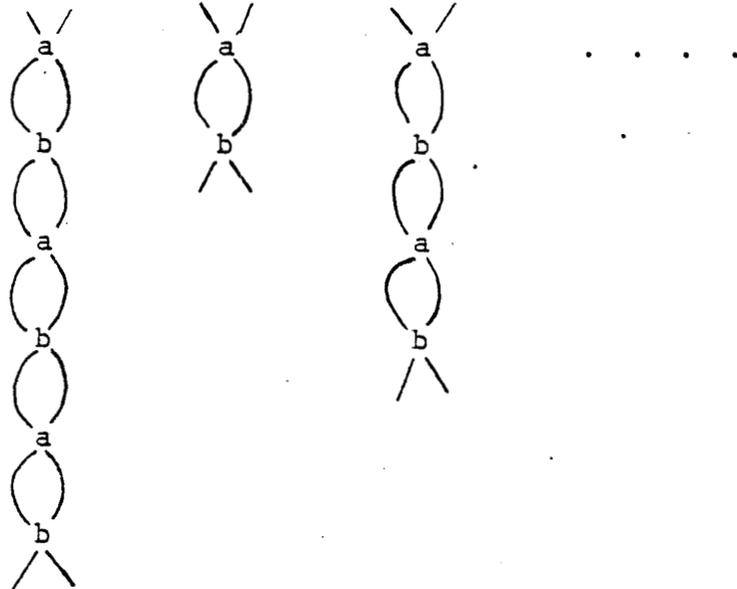
Soit l'automate M définit par

$$\begin{aligned}
 I &= \{1 \otimes 2\} \\
 F &= \{1 \otimes 2\} \\
 \left. \begin{aligned}
 (1 \otimes 2) \cdot a &\rightarrow a \cdot (2 \otimes 1) \\
 (2 \otimes 1) \cdot b &\rightarrow b \cdot (1 \otimes 2)
 \end{aligned} \right\} &= R
 \end{aligned}$$

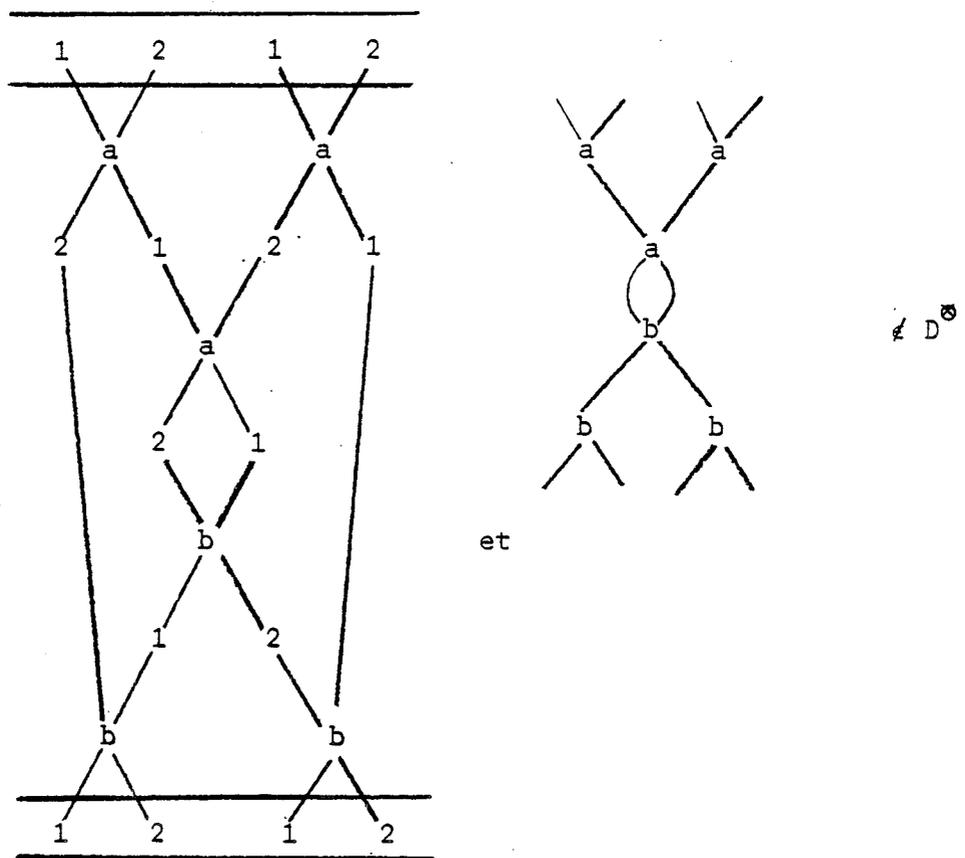
$T(M) = D = (a \cdot b)^*$, c'est-à-dire les dags de la forme :



et $D^{\otimes} = ((a \cdot b)^*)^{\otimes}$, c'est-à-dire les dags de la forme



alors la construction de M' ne prenant en compte que ce seul premier principe admettrait la reconnaissance



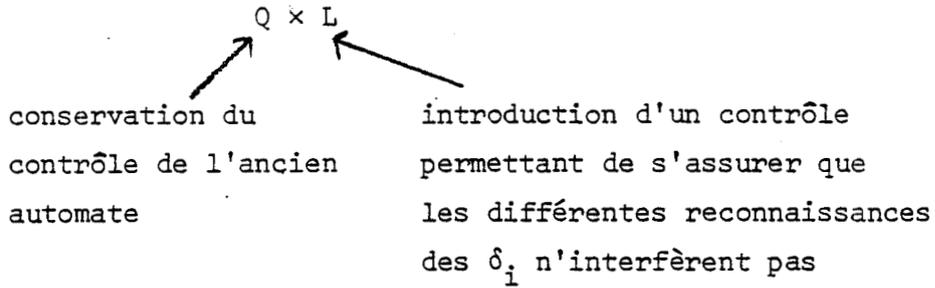
avec $(1 \otimes 2) \otimes (1 \otimes 2) \in I^{\otimes}$ et F^{\otimes} .

Dans cet exemple, une des lettres ~~a~~ est reconnue en étant contrôlée par des états issus de deux débuts de reconnaissances qui auraient dû rester indépendantes. Et "malheureusement", la reconnaissance concernée aboutit à une configuration d'état élément de F^{\otimes} , donc à une reconnaissance acceptée par l'automate M' , mais le dag reconnu n'est pas élément de D^{\otimes} .

Il est clair que les interférences possibles entre reconnaissances indépendantes se situent sur les "bords" de ces reconnaissances. C'est pourquoi nous allons mémoriser dans les états une information supplémentaire.

Deuxième principe de la construction de l'automate M' :

Puisque le problème se situe sur les "bords" des reconnaissances des δ_i , nous allons définir l'automate M' sur l'ensemble d'états.



où $L = \{bg, m, bd, 0\}$, ce qui permet de "marquer" chaque état et de savoir son rôle dans la reconnaissance en interprétant

bg \rightarrow état bord gauche (de la reconnaissance d'un $\omega_i \in D$)

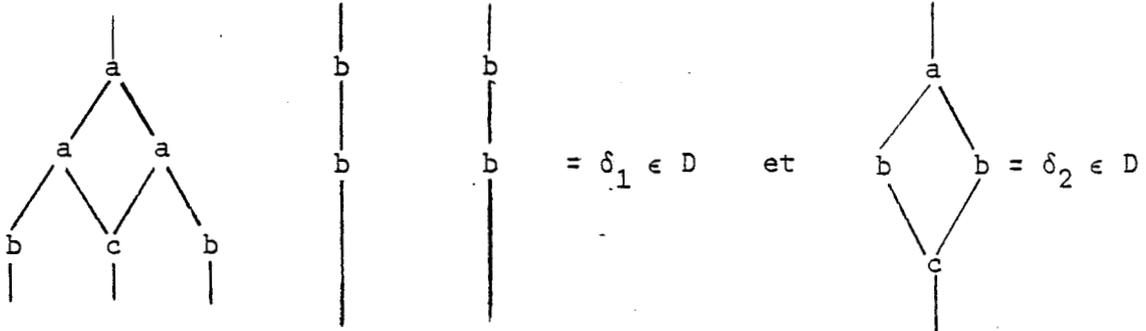
m \rightarrow état milieu

bd \rightarrow état bord droit

0 \rightarrow état à la fois bord gauche et bord droit (cas de coarité ou arité = 1)

Exemple :

Considérons les deux dags planaires

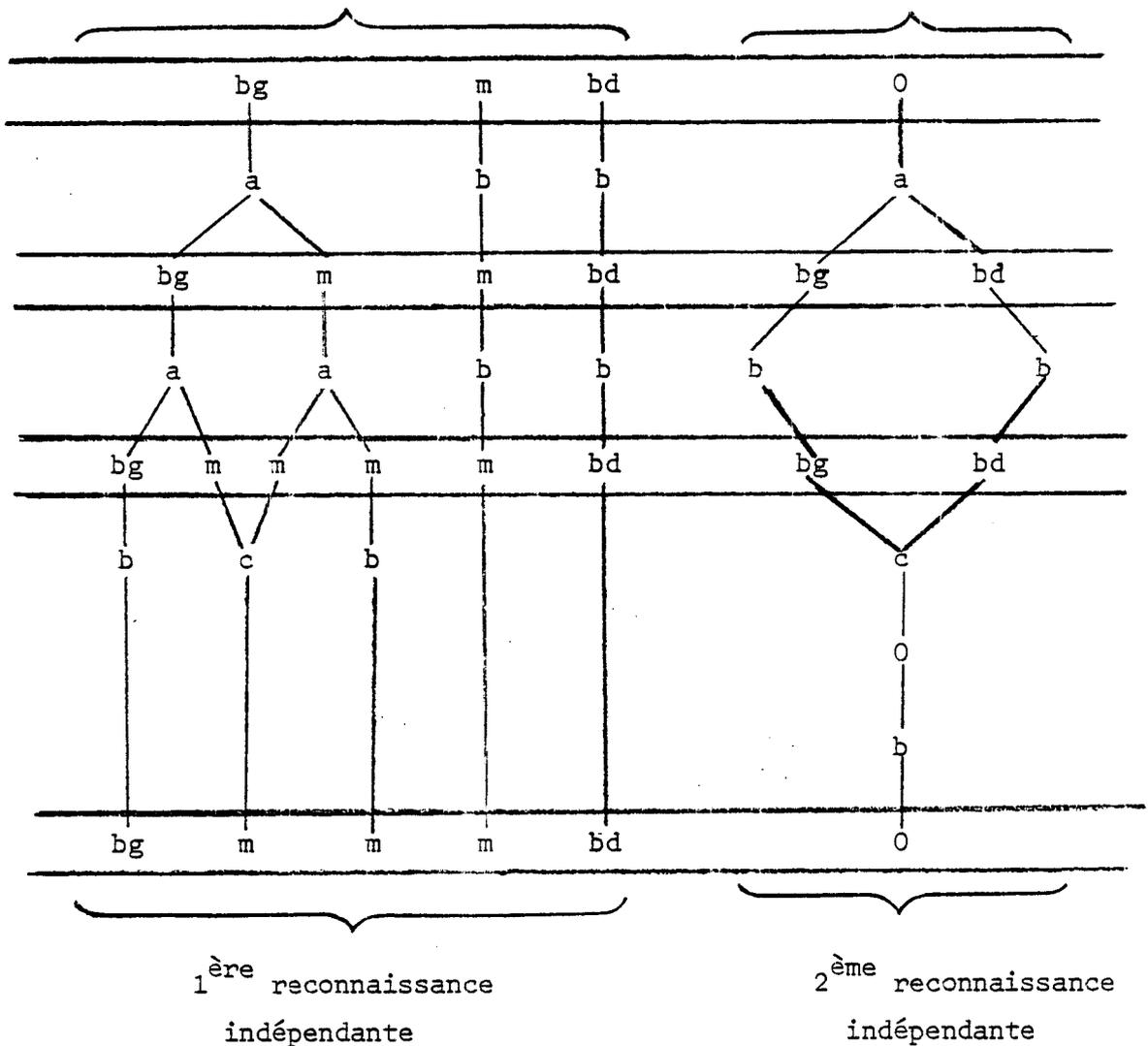


alors, en ne nous *intéressant qu'à la composante L des états*, la reconnaissance de $\delta_1 \otimes \delta_2$ par M' peut se schématiser :

reconnaissance de

 δ_1

⊗

 δ_2 

C'est-à-dire que :

- Dès le début de la reconnaissance un choix de découpage des reconnaissances indépendantes est fait à PRIORI, grâce au choix de la composante L du contrôle I'. Dans l'exemple ci-dessus le bon choix était $bg \otimes m \otimes bd \otimes 0$.

- Les règles vont "propager" le découpage choisi par I' et donc empêcher que les reconnaissances indépendantes s'interfèrent puisque une même lettre ne pourra réécrire une séquence d'états contenant : $bd \otimes bg$, $bd \otimes 0$, $0 \otimes bg$ ou $0 \otimes 0$.

- Enfin, le contrôle bas F' va vérifier A POSTERIORI (si on réussit une reconnaissance complète) que l'on aboutit pour chaque découpage, à un mot de F . Ceci se fait en imposant à F' la même contrainte que celle imposée à I' .

Application des deux principes :

Les états Q' de l'automate M' à construire étant $Q \times L$, les deux principes expliqués se traduisent par :

- premier principe (traduit plus formellement en II.2.4.5.)

$$\begin{aligned} - \Pi_Q(I') &= I^{\otimes} \\ - (m, q, m') \in R' &\implies (\Pi_Q(m), q, \Pi_Q(m')) \in R \\ - \Pi_Q(F') &= F^{\otimes} \end{aligned}$$

- deuxième principe

$$\begin{aligned} - \forall m \in (Q \times L)^{\otimes} \text{ tel que } m \text{ et un facteur de } I' \\ \Pi_L(m) \in L \implies \Pi_Q(m) \in I \text{ où } L = (bg \otimes m^{\otimes} \otimes bd + 0) \\ - \forall (m, a, m') \in R' \implies \\ - \text{soit } \Pi_L(m) \text{ et } \Pi_L(m') \in m^{\otimes} \otimes bd \\ - \text{soit } \Pi_L(m) \text{ et } \Pi_L(m') \in bg \otimes m \\ - \text{soit } \Pi_L(m) \text{ et } \Pi_L(m') \in m^{\otimes} \\ - \text{soit } \Pi_L(m) \text{ et } \Pi_L(m') \in L \end{aligned}$$

II.2.4.5. - Construction

$$M' = \langle \Sigma, Q' = Q \times L, I' = \text{Sh}(I, L)^{\otimes}, F' = \text{Sh}(F, L)^{\otimes}, R' \rangle \text{ où}$$

Sh est l'opérateur définit en II.2.2.6.

$$R' = \{(m, a, m') \text{ tel que } (\Pi_Q(m), a, \Pi_Q(m')) \in R \text{ et}$$

soit $\Pi_L(m)$ et $\Pi_L(m') \in m^{\otimes} \otimes bd$

soit $\Pi_L(m)$ et $\Pi_L(m') \in bg \otimes m$

soit $\Pi_L(m)$ et $\Pi_L(m') \in m^{\otimes}$

soit $\Pi_L(m)$ et $\Pi_L(m') \in L$

où $L = (bg \otimes m^{\otimes} \otimes bd + 0)$.

II.2.4.6. - Démonstration

Après avoir rappelé que l'ensemble des langages rationnels de mots est clos par l'opérateur Sh [déf. en II.2.2.6.], et avoir montré

II.2.4.6.1. - Lemme

$\forall \delta \in \hat{DP}(\Sigma), \forall m \in Q^{\otimes}, \forall m' \in Q^{\otimes}$, on a

$$m \cdot \delta \Big|_M^* \delta \cdot m' \iff \exists u \in (Q \times L)^{\otimes} \text{ avec } \Pi_Q(u) = m \text{ et } \Pi_L(u) \in L$$

$$\exists u' \in (Q \times L)^{\otimes} \text{ avec } \Pi_Q(u') = m' \text{ et } \Pi_L(u') \in L$$

$$\text{et } u \cdot \delta \Big|_{M'}^* \delta \cdot u'$$

il vient :

$$\delta \in D^{\otimes} \iff \delta = \delta_1 \otimes \dots \otimes \delta_n \text{ avec } \delta_i \in D \text{ pour } i = 1, \dots, n$$

$$\iff \exists m = m_1 \otimes \dots \otimes m_n \text{ avec } m_i \in I \text{ pour } i = 1, \dots, n$$

$$\exists m' = m'_1 \otimes \dots \otimes m'_n \text{ avec } m'_i \in F \text{ pour } i = 1, \dots, n$$

$$\text{et } m_i \cdot \delta_i \Big|_M^* \delta_i \cdot m'_i$$

en utilisant n fois le lemme

$$\iff \exists u = u_1 \otimes \dots \otimes u_n \text{ avec } \Pi_Q(u_i) = m_i \in I \text{ et } \Pi_L(u_i) \in L \text{ pour } i = 1, \dots, n$$

$$\exists u' = u'_1 \otimes \dots \otimes u'_n \text{ avec } \Pi_Q(u'_i) = m'_i \in F \text{ et } \Pi_L(u'_i) \in L \text{ pour } i = 1, \dots, n$$

$$\text{et } u_i \cdot \delta_i \Big|_{M'}^* \delta_i \cdot u'_i \text{ pour } i = 1, \dots, n$$

d'après la définition

$$\iff \exists u \in \text{Sh}(I, L)^{\otimes}, \exists u' \in \text{Sh}(F, L)^{\otimes} \text{ tels que } u \cdot \delta \Big|_{M'}^* \delta \cdot u'$$

étant donné que $I' = \text{Sh}(I, L)^{\otimes}$ et $F' = \text{Sh}(F, L)$

$$\iff \delta \in T(M').$$

□

II.2.4.7. - Preuve du lemme

Par récurrence sur le nombre, l , d'occurrences de lettres dans δ .

- $l = 0$ $\Rightarrow m = m'$, $\delta = e_p$ et $u = u' = \text{Sh}(\{m\}, \{v\})$ où v est l'unique mot de L de longueur $|m|$, le lemme se vérifie.

Hypothèse de récurrence : $\forall p \in \mathbb{N}^*$, le lemme se vérifie pour $\delta \in \hat{D}\mathcal{P}(\Sigma)$ tel que son nombre d'occurrences de lettres soit strictement inférieur à p .

- $l = p$ $\Rightarrow \delta = \delta_1 \cdot (e_i \otimes a \otimes e_j)$, où le nombre d'occurrences de lettres dans δ_1 est $p-1$.

$$\Rightarrow m \cdot \delta \stackrel{*}{M} \delta_1 \cdot m'' \cdot (e_i \otimes a \otimes e_j) \stackrel{*}{M} \omega \cdot m'$$

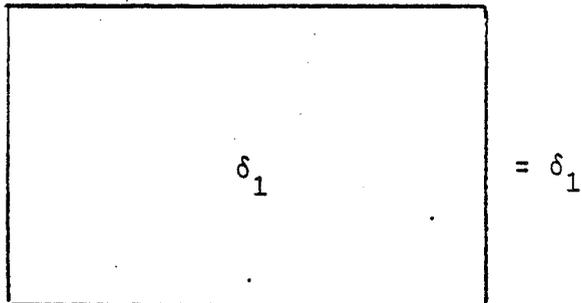
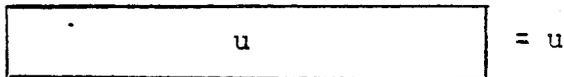
$$\Leftrightarrow m \cdot \delta_1 \stackrel{*}{M} \delta_1 \cdot m'' \text{ et}$$

Il existe une règle de R : $m'' \cdot a \rightarrow a \cdot m'_2$ telle que

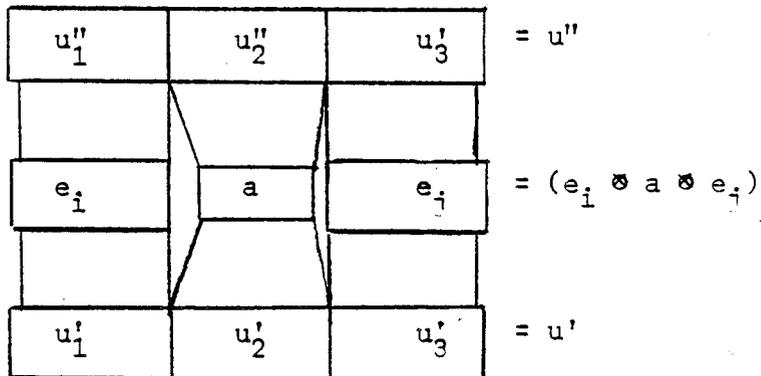
$$m'' = m'_1 \otimes m''_2 \otimes m'_3 \text{ et } m' = m'_1 \otimes m'_2 \otimes m'_3$$

\Leftrightarrow (i) en utilisant l'hypothèse de récurrence pour δ_1 , on a :

$\Pi_Q(u) = m$ et $\Pi_L(u) \in L$



$\Pi_Q(u'') = m''$ et $\Pi_L(u'') \in L$



avec $\Pi_Q(u'_1) = m'_1, \Pi_Q(u'_2) = m'_2, \Pi_Q(u'_3) = m'_3$

et $m'_2 \cdot a \rightarrow a \cdot m'_2 \in R$.

\Leftrightarrow d'après la définition des règles de M' et en remarquant que $\Pi_L(u'') \in L \Rightarrow \Pi_L(u'_2) \in m \otimes bd + bg \otimes m + m + L$, il est clair que

$u \cdot \delta \xrightarrow{M'} \delta \cdot u'$ où $\delta = \delta_1 \cdot (e_i \otimes a \otimes e_j)$

tel que

$\Pi_Q(u) = m$ et $\Pi_L(u) \in L$

$\Pi_Q(u') = m'$ et $\Pi_L(u') \in L$

□

II.2.4.8. - Remarque

Une autre construction pour M' est possible. Elle consiste à construire M' à partir de deux automates M_1 et M_2 , chacun reconnaissant D , mais ayant des ensembles d'états disjoints. Ainsi soient $M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ et $M_2 = \langle \Sigma, I_2, F_2, R_2 \rangle$ tels que $T(M_1) = T(M_2) = D$ avec $Q_1 \cap Q_2 = \emptyset$, en traitant à part le cas où $e_0 \in I_1 \cup I_2$:

$$M' = \langle \Sigma, Q_1 \cup Q_2, (I_1 \otimes (I_2 \otimes I_1) + (I_1 \times I_2)) \otimes, (F_1 \otimes (F_2 \otimes F_1) + (F_1 \otimes F_2)) \otimes, R_1 \cup R_2 \rangle.$$

II.2.5. - Clôture par produit de compositionII.2.5.1. - Propriété

REC est clos par produit de composition.

II.2.5.2. - Preuve

$\forall D_1 \in \text{REC}, \forall D_2 \in \text{REC}$, il existe effectivement un automate M_3 tel que $T(M_3) = D_1 \cdot D_2$.

II.2.5.3. - Construction

$D_1 \in \text{REC} \Rightarrow$ il existe un automate $M_1 = \langle \Sigma_1, Q_1, I_1, F_1, R_1 \rangle$ tel que $T(M_1) = D_1$, $D_2 \in \text{REC} \Rightarrow$ il existe un automate $M_2 = \langle \Sigma_2, Q_2, I_2, F_2, R_2 \rangle$ tel que $T(M_2) = D_2$.

En utilisant les corollaires [II.1.4.6.] et [II.1.4.7.], on peut construire l'automate $M'_1 = \langle \Sigma_1, Q'_1, I'_1, F'_1, R'_1 \rangle$ tel que $T(M_1) = T(M'_1)$ et $F'_1 = \{q_f\} \otimes$ et l'automate $M'_2 = \langle \Sigma_2, Q'_2, I'_2, F'_2, R'_2 \rangle$ tel que $T(M_2) = T(M'_2)$ et $I'_2 = \{q_f\} \otimes$.

$$M_1 \xrightarrow{\text{remontée du contrôle bas}} M'_1 / F'_1 = \{q_f\} \otimes$$

$$M_2 \xrightarrow{\text{descente du contrôle haut}} M'_2 / I'_2 = \{q_f\} \otimes$$

de plus, on peut supposer que $Q'_1 \cap Q'_2 = \{q_f\}$ et ainsi

$$M_3 = \langle \Sigma_1 \cup \Sigma_2, Q_1' \cup Q_2', I_1', F_2', R_1' \cup R_2' \rangle$$

II.2.5.4. - Démonstration : directement

$$\begin{aligned} \omega \in D_1 \cdot D_2 &\Leftrightarrow \omega = \omega_1 \cdot \omega_2 \text{ avec } \omega_1 \in D_1 \text{ et } \omega_2 \in D_2 \\ \Leftrightarrow \exists m_1 \in I_1', \exists m_1' \in F_1' / m_1 \cdot \omega_1 \stackrel{*}{M_1} \omega_1 \cdot m_1' \text{ et } \exists m_1'' \in I_2', \exists m_2' \in F_2' / \\ m_1' \cdot \omega_2 \stackrel{*}{M_2} \omega \cdot m_2'. \end{aligned}$$

$$\text{puisque } Q_1' \cap Q_2' = \{q_f\}, F_1' = \{q_f\}^{\otimes} = I_2'$$

$$\Leftrightarrow \exists m_1 \in I_1', \exists m_2' \in F_2' / m_1 \cdot \omega_1 \cdot \omega_2 \stackrel{*}{M_3} \omega_1 \cdot m_1' \cdot \omega_2 \stackrel{*}{M_3} \omega_1 \cdot \omega_2 \cdot m_2'$$

$$\Leftrightarrow \omega_1 \cdot \omega_2 = \omega \in T(M_3).$$

□

II.2.6. - Non-clôture par étoile du produit de composition

II.2.6.1. - Propriété

REC n'est pas clos par étoile du produit de composition.

II.2.6.2. - Preuve

Soit le taillis reconnaissable T , $T = \left\{ \begin{array}{c} | \\ c \otimes d \\ | \end{array} \right\}$, alors T^* n'est pas un taillis reconnaissable, en effet REC est clos par produit de composition car

$$\begin{array}{c} \begin{array}{c} | \\ \{a\} \\ \wedge \end{array} \cdot T^* = \left\{ \begin{array}{c} | \\ a \\ \wedge \\ \begin{array}{cc} c & d \\ | & | \end{array} \end{array}, \begin{array}{c} | \\ a \\ \wedge \\ \begin{array}{cc} c & d \\ | & | \\ c & d \\ | & | \end{array} \end{array}, \begin{array}{c} | \\ a \\ \wedge \\ \begin{array}{cc} c & d \\ | & | \\ c & d \\ | & | \\ c & d \\ | & | \end{array} \end{array}, \dots \right\} \text{ n'est pas un taillis reconnais-} \end{array}$$

sable puisque ce n'est pas une forêt reconnaissable.

□

II.2.7. - Non-clôture par homomorphisme de magmaïde

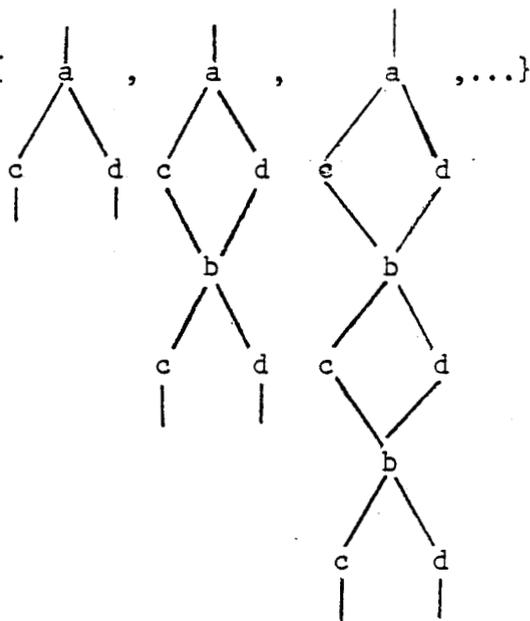
II.2.7.1. - Propriété

REC n'est pas clos par homomorphisme de magmaïde.

II.2.7.2. - Preuve

Soit $h : \hat{DP}(\Sigma) \rightarrow \hat{DP}(\Sigma')$, homomorphisme de magmaïde alors
 $D \in \text{REC sur } \Sigma \not\Rightarrow h(D) \in \text{REC sur } \Sigma'$. En effet,

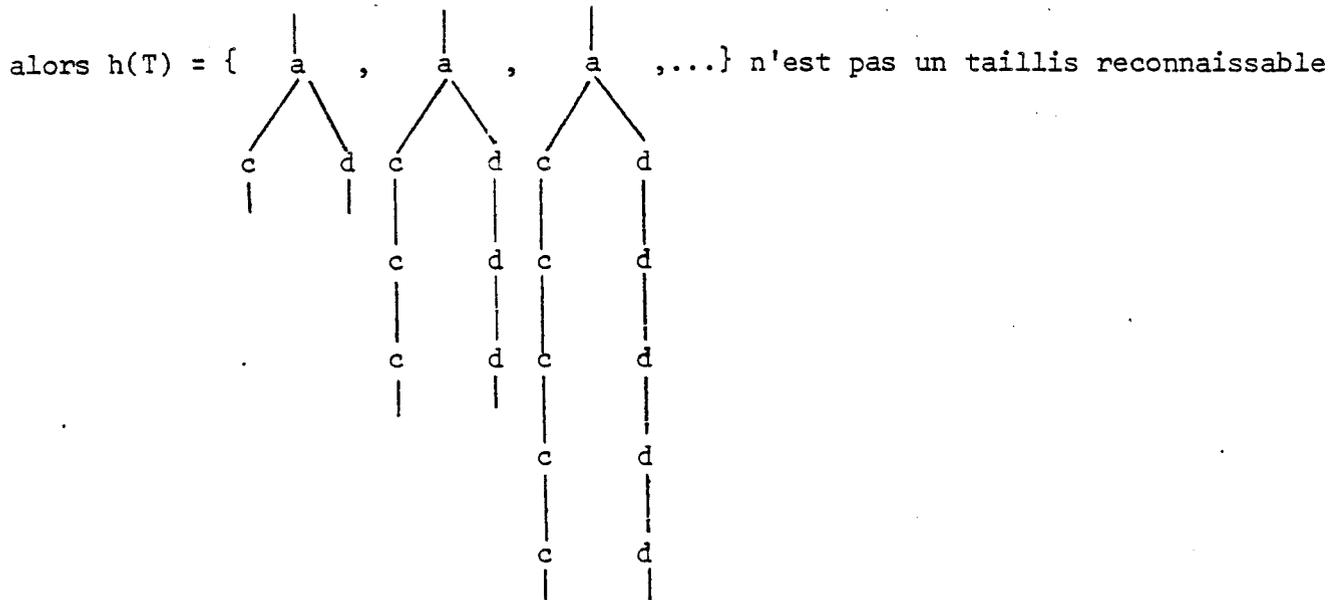
Soit le taillis reconnaissable $T = \{$



reconnu par : $q_0 \cdot a \vdash a \cdot (q_1 \otimes q_2)$; $q_1 \cdot c \vdash c \cdot q_f$; $q_2 \cdot d \vdash d \cdot q_f$
 $(q_f \otimes q_f) \cdot b \vdash b \cdot (q_1 \otimes q_2)$ avec $I = \{q_0\}$ et $F = \{q_f \otimes q_f\}$ où

$\Sigma = \{a, b, c, d\}$ et $h : \hat{DP}(\Sigma) \rightarrow \hat{DP}(\Sigma)$, morphisme de magmaïde défini par :

$$\begin{array}{l}
 \begin{array}{c} | \\ \hline a \\ | \\ \hline c \\ | \\ \hline \end{array} \\
 h(a) = a \\
 \begin{array}{c} | \\ \hline c \\ | \\ \hline \end{array} \\
 h(c) = c
 \end{array}
 \quad
 \begin{array}{l}
 \begin{array}{c} \diagup \diagdown \\ | \\ \hline b \\ | \\ \hline c \quad d \\ | \quad | \\ \hline \end{array} \\
 h(b) = c \otimes d \\
 \begin{array}{c} | \\ \hline d \\ | \\ \hline \end{array} \\
 h(d) = d
 \end{array}
 \end{array}$$



puisque ce n'est pas une forêt reconnaissable. □

II.2.7.3. - Corollaire

REC n'est pas clos par homomorphisme alphabétique effaçant.

Preuve du corollaire :

Même exemple mais $h(b) = e \otimes e$. □

II.2.8. - Clôture par homomorphisme connexe

II.2.8.1. - Propriété

REC est clos par homomorphisme connexe.

II.2.8.2. - Preuve

$\forall D \in \text{REC}$ et $\forall h : \hat{D}P(\Sigma) \rightarrow \hat{D}P(\Sigma')$, homomorphisme connexe \Rightarrow il existe effectivement un automate M' tel que $T(M') = h(D)$.

II.2.8.3. - Notations

Notons $M = \langle \Sigma, Q, I, F, R \rangle$ l'automate qui reconnaît D et $M' = \langle \Sigma', Q', I', F', R' \rangle$ l'automate que nous cherchons à construire tel que $T(M') = h(D)$.

II.2.8.4. - Idée de construction

L'idée de la construction peut s'expliquer en quatre points.

Premier point

Il serait naturel de construire M' en s'assurant $Q \subseteq Q'$ et la propriété

$$(q_1 \otimes \dots \otimes q_n) \cdot a \rightarrow a \cdot (q'_1 \otimes \dots \otimes q'_n) \in R \iff (q_1 \otimes \dots \otimes q_n) \cdot \delta \left| \frac{*}{M'} \right. \delta \cdot (q'_1 \otimes \dots \otimes q'_n)$$

où $\delta \in (h(a))^*$ si $q_1 \otimes \dots \otimes q_n = q'_1 \otimes \dots \otimes q'_n$,
 $\delta = h(a)$ sinon

Puis de poser $I = I'$ et $F = F'$. Mais nous ne pouvons appliquer cette propriété à toutes les règles de R . Ainsi la règle $q \cdot b \rightarrow b \cdot q'$ telle que $q \neq q'$ et $h(b) = \underset{|}{e}$ ne pourra respecter la propriété énoncée, car il faudrait que $q \cdot \underset{|}{e} \left| \frac{*}{M'} \right. \underset{|}{e} \cdot q'$ avec $q \neq q'$, or ceci est impossible puisque $q \cdot \underset{|}{e} \left| \frac{*}{M'} \right. \underset{|}{e} \cdot q'$ si et seulement si $q = q'$.

Remarque :

$$h(b) = \underset{|}{e} \implies b \in \Sigma_1^1 \text{ puisque l'homomorphisme } h \text{ est connexe.}$$

Deuxième point :

Il faut donc compléter l'idée de construction proposée au premier point en distinguant les règles de R où la lettre reconnue ne s'efface pas (i.e. $h(a) \neq \underset{|}{e}$) et celles où la lettre reconnue s'efface. Ainsi

a) Pour les règles $r \in R$, où la lettre reconnue ne s'efface pas,
 i.e. $r = (q_1 \otimes \dots \otimes q_n) \cdot a \rightarrow a \cdot (q'_1 \otimes \dots \otimes q'_n) \in R$ avec $h(a) \neq \epsilon$ on construit M' en respectant la propriété énoncée au premier point. C'est-à-dire plus concrètement, à chacune de ces règles r , on associe un ensemble de règles R_r et d'états Q_r tels que :

$$(q_1 \otimes \dots \otimes q_n) \cdot \delta \stackrel{*}{\underset{R_r}{\mid}} \delta \cdot (q'_1 \otimes \dots \otimes q'_n) \iff \begin{cases} \delta \in (h(a))^* & \text{si } q_1 \otimes \dots \otimes q_n = q'_1 \otimes \dots \otimes q'_n \\ \delta = h(a) & \text{sinon} \end{cases}$$

b) Pour les règles de R où la lettre s'efface par h , il faut prévoir à priori les effacements possibles de lettres, à la suite de chaque reconnaissance d'un $h(a) \neq \epsilon$. Plus concrètement, cela révient à considérer que :

- s'il apparaît un état $q \in Q / q \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta \cdot q'$ avec $h(\delta) = \epsilon$, en partie droite d'une des règles d'un des ensembles R_r , alors il peut apparaître aussi l'état q' .

Troisième point :

Mais le principe ci-dessus ne permet pas de tenir compte des lettres qui s'efface dès le début, il faut donc définir le contrôle haut de M' comme étant le contrôle haut de M mais où chaque fois qu'apparaît un état $q \in Q / q \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta \cdot q'$ avec $h(\delta) = \epsilon$, il peut apparaître l'état q' .

Quatrième point :

En s'assurant (pour des raisons techniques : éviter des "interférences" entre les reconnaissances des différents ensembles de règles R_r), que les Q_r soient disjoints deux à deux et disjoints de Q , la construction de M' est immédiate (la construction formelle se trouve en II.2.8.6. et II.2.8.7.) :

$$Q' = \bigcup_{r \in R} Q_r \cup Q$$

$$R' = \bigcup_{r \in R} R_r$$

I' est issu de I en tenant compte du troisième point

$$F' = F$$

II.2.8.5. -- Exemple

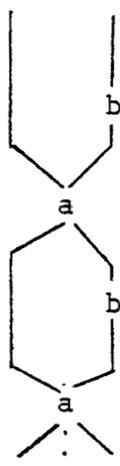
Soit l'automate $M = \langle \{a, b\}, \{q_1, q_2, q_3\}, I = \{q_1 \otimes q_3\}, F = \{q_1 \otimes q_3\}, R \rangle$
 où $R =$

règle 1 = $(q_1 \otimes q_2) \cdot a \rightarrow a \cdot (q_1 \otimes q_3)$

règle 2 = $q_3 \cdot b \rightarrow b \cdot q_2$

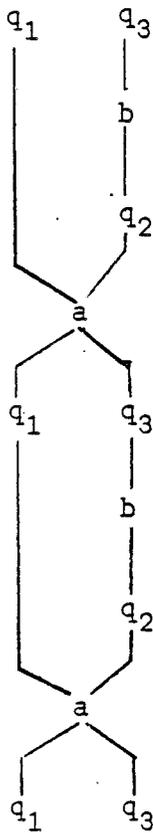
M reconnaît donc les dags de la forme : $(e \otimes b) \cdot a \cdot (e \otimes b) \cdot a \dots$

soit schématiquement :



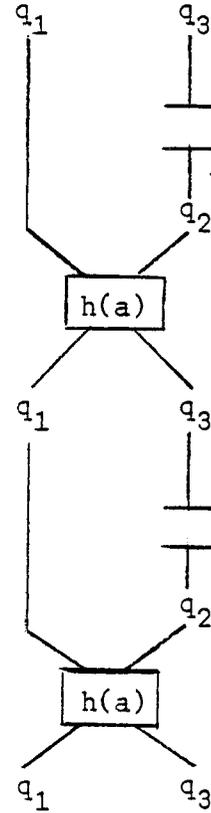
Soit h , l'homomorphisme connexe, sur $\{a, b\}$ tel que $h(b) = \begin{matrix} | \\ e \\ | \end{matrix}$.

On peut illustrer le défaut de la propriété énoncée au premier point.



En se limitant
aux règles définies
au premier point

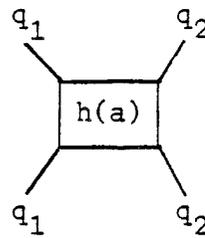
développé dans
l'idée de construction
on a :



dans les règles
définies au
premier point
le problème de
l'effacement
sible n'a pas
été pris en co
te

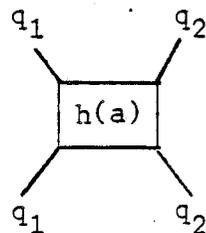
en ajoutant les modifications des points suivants :

$$I' = \{q_1 \otimes q_3, q_1 \otimes q_2\}$$

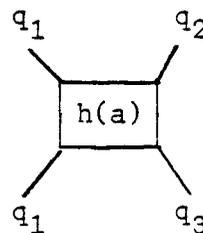


ainsi $M' = \langle \Sigma', \bigcup_{r \in R} Q_r \cup Q, \{q_1 \otimes q_3, q_1 \otimes q_2\}, F, \bigcup_{r \in R} R_r \rangle$ où Q_r et R_r sont

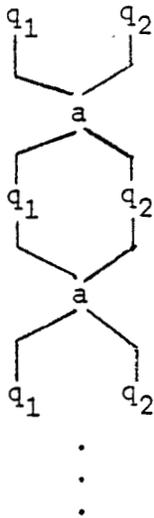
respectivement les états et les règles intermédiaires pour avoir



et



et les reconnaissances de M' peuvent se schématiser :

Remarque 1 :

Nous ne décrivons pas les règles qui permettront d'avoir $(q_1 \otimes q_2) \cdot h(a) \stackrel{*}{\vdash} h(a) \cdot (q_1 \otimes q_2)$ puisque $h(a)$ est un dag fini connexe, cela ne pose pas de problème. Nous prendrons seulement la précaution nécessaire d'utiliser de nouveaux états pour les états, Q_r , intermédiaires, afin que les règles définies pour reconnaître $h(a)$ n'interfèrent pas avec les autres.

Remarque 2 :

L'exemple choisi est assez simple, et certaines règles issues de la construction peuvent apparaître superflues, comme le fait d'avoir $(q_1 \otimes q_2) \cdot h(a) \stackrel{*}{\vdash}_{M'} h(a) \cdot (q_1 \otimes q_3)$, qui ne servira pas dans les reconnaissances de M' . Mais il suffit d'envisager que l'automate initial M possède la règle $(q_1 \otimes q_3) \cdot c \rightarrow c \cdot (q_1 \otimes q_2)$ pour s'apercevoir de son utilité.

II.2.8.6. - Construction

Notons $R_1 = \{q \cdot b \rightarrow b \cdot q' \in R \mid h(b) = e\}$

et $R_2 = R \setminus R_1 = \{m \cdot a \rightarrow a \cdot m' \mid h(a) \neq e\}$ alors

a) à chaque règle $r \in R_2 : m \cdot a \rightarrow a \cdot m$, on associe

- un ensemble de nouveaux états Q_r

- un ensemble de règles sur $(Q_r \cup Q)^\otimes \times \Sigma \times (Q_r \cup Q)^\otimes$ tels que

(i) les ensembles Q_r soient disjoints deux à deux et disjoints de Q

(ii) $\forall m_1 \in Q^\otimes, \forall m_2 \in Q^\otimes, \forall \delta \in \tilde{DP}(\Sigma), \forall \delta' \in \tilde{DP}(\Sigma')$

$$m_1 \cdot \delta \stackrel{*}{\underset{R}{\mid}} \delta \cdot m_2 \text{ et } \delta' = h(\delta) \iff m_1 \cdot \delta' \stackrel{*}{\underset{R}{\mid}} \delta' \cdot m_2$$

b) Ainsi $M' = \langle \Sigma', \bigcup_{r \in R} Q_r \cup Q, s(I), F, R' \rangle$ où

(i) s est la substitution finie, définit par

$$\forall q \in Q, s(q) = \{q\} \cup \{q' \in Q / q \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta \cdot q' \text{ et } h(\delta) = e\}$$

(ii) $R' = \{m \cdot a \rightarrow a \cdot s(m') / m \cdot a \rightarrow a \cdot m' \in \bigcup_{r \in R_2} R_r\}$.

II.2.8.7. - Démonstration

En trois temps nous illustrons d'abord par un exemple, afin de mieux exposer le problème, que si l'homomorphisme h n'est pas connexe alors la construction proposée peut s'avérer impossible. Deuxièmement, nous montrerons que si l'homomorphisme h est connexe alors la construction est effective ; et enfin nous montrerons que la construction proposée est telle que $T(M') = h(D)$.

II.2.8.7.1. - On peut facilement voir sur un exemple que si l'homomorphisme h n'est pas connexe, la construction proposée peut s'avérer impossible puisque du fait de la non-connexité de h , les conditions imposées à la construction peuvent, dans certains cas, se contredire. En effet :

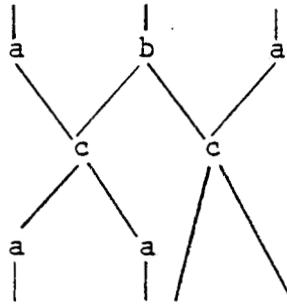
Considérons la règle $i : (q_1 \otimes q_2) \cdot a \rightarrow a \cdot (q_1 \otimes q_3)$ telle que

$h(a) = b \otimes c$ alors il faut trouver R_i telle que

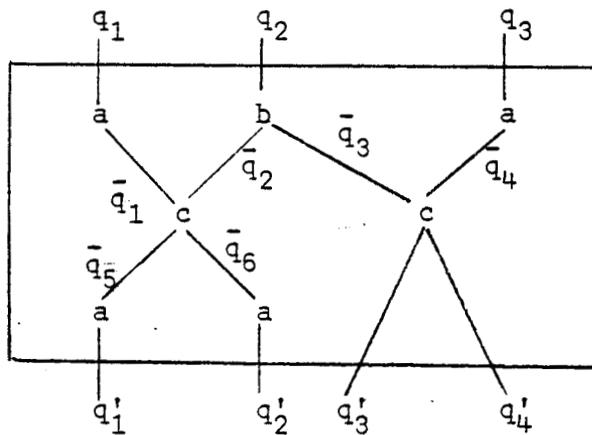
$(q_1 \otimes q_2) \cdot (b \otimes c) \stackrel{*}{\underset{R_i}{\mid}} (b \otimes c) \cdot (q_1 \otimes q_3)$, condition nécessaire, \Rightarrow

$(q_1 \otimes q_2) \cdot (e \otimes c) \stackrel{*}{\underset{R_i}{\mid}} (e \otimes c) \cdot (q_1 \otimes q_3)$, interdit puisque $e \otimes c \neq h(a)$.

II.2.8.7.2. - La construction de M' sera effective, si pour chaque règle $i \in R$, la construction de R_i est effective. La description formelle du processus de construction et sa validation n'apporte rien, limitons-nous à un exemple : soit la règle $i : (q_1 \otimes q_2 \otimes q_3) \cdot d \rightarrow d \cdot (q'_1 \otimes q'_2 \otimes q'_3 \otimes q'_4)$ avec



on peut schématiser R_i par
où $Q_i = \{\bar{q}_1, \bar{q}_2, \bar{q}_3, \bar{q}_4, \bar{q}_5, \bar{q}_6\}$



Le principe de la construction étant de créer dans R_i , une règle par occurrence de lettres dans $h(d)$. Un exemple de construction de même nature se trouve en III.2.1.4.

II.2.8.7.3. - On démontre facilement que la construction proposée est bonne, i.e. $T(M') = h(D)$. En effet, après avoir montré

II.2.8.7.3.1. - Lemme

$\forall q_i \in Q^{\emptyset}, \forall q_f \in Q^{\emptyset}, \forall \delta \in \hat{D}\hat{P}(\Sigma), \forall \delta' \in \hat{D}\hat{P}(\Sigma') :$

$$q_i \cdot \delta \Big|_M^* \delta \cdot q_f \text{ et } \delta' = h(\delta) \iff \exists q'_i \in s(q_i) / q'_i \cdot \delta' \Big|_{M'}^* \delta' \cdot q_f$$

il vient :

$\delta \in T(M) \iff \exists q_i \in I, \exists q_f \in F / q_i \cdot \delta \Big|_M^* \delta \cdot q_f$ et $\delta' = h(\delta)$ en utilisant le lemme

$$\iff \exists q'_i \in I', \exists q_f \in F' / q'_i \cdot \delta' \Big|_{M'}^* \delta' \cdot q_f$$

$$\iff \delta' \in T(M').$$

□

II.2.8.7.4. - Preuve du lemme

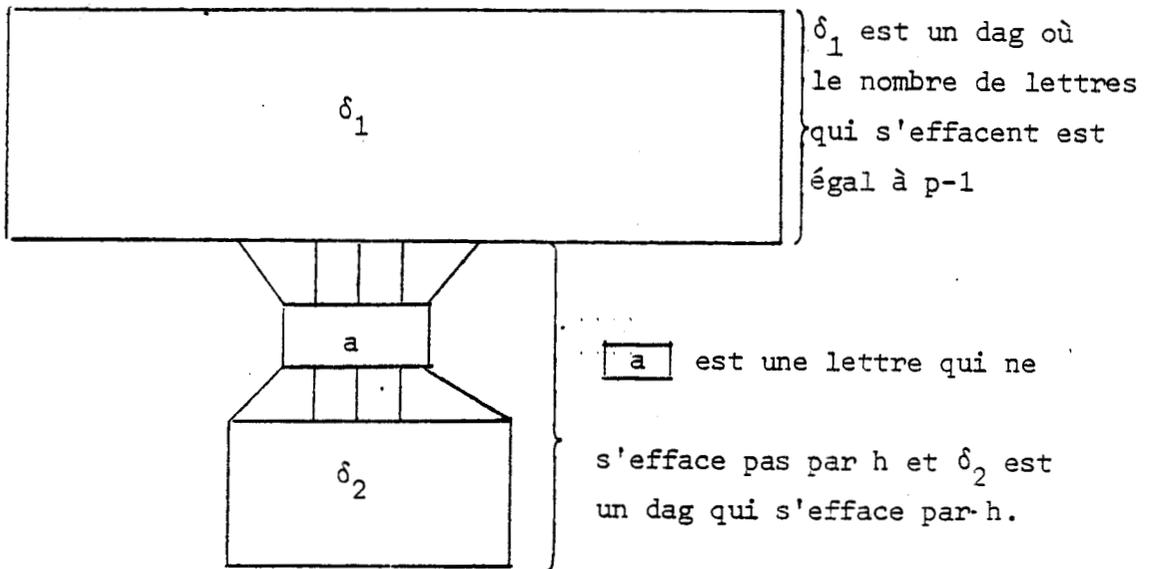
Par récurrence sur le nombre, ℓ , de lettres de δ , qui ne s'effacent pas

- $\underline{\ell = 0} \Rightarrow h(\delta) = e_n \Rightarrow \delta = \delta_1 \otimes \dots \otimes \delta_n$ avec $\delta_i \in \hat{D}\hat{P}(\Sigma)_1^1$ et $h(\delta_i) = e$, pour $i = 1, \dots, n$. Ceci, car l'homomorphisme h est connexe. Ainsi $q'_i = q_f$ et de par la définition de la substitution s , le lemme se vérifie.

Hypothèse de récurrence :

$\forall p \in \mathbb{N}^*, \forall \delta \in \hat{D}\hat{P}(\Sigma)$ où le nombre, ℓ , de lettres qui ne s'effacent pas est strictement inférieur à p , le lemme se vérifie.

- $\underline{\ell = p}$: dans ce cas, on peut considérer δ :



c'est-à-dire $\delta = \delta_1 \cdot (e_n \otimes a \cdot \delta_2 \otimes e_m)$.

Ainsi, $q_i \cdot \delta \stackrel{*}{\underset{M}{|}} \delta \cdot q_f$ et $\delta' = h(\delta)$

$\Leftrightarrow q_i \cdot \delta_1 \stackrel{*}{\underset{M}{|}} \delta_1 \cdot q_1$ et $q_1 \cdot (e_n \otimes a \cdot \delta_2 \otimes e_m) \stackrel{*}{\underset{M}{|}} (e_n \otimes a \cdot \delta_2 \otimes e_m) \cdot q_f$
et $\delta' = h(\delta)$ en utilisant l'hypothèse de récurrence pour δ_1

$\Leftrightarrow \exists q_i' \in s(q_i) / q_i' \cdot h(\delta_1) \stackrel{*}{\underset{M'}{|}} h(\delta_1) \cdot q_1$ et il existe une règle
 $r = m \cdot a \rightarrow a \cdot m' \in R$ telle que $h(a) \neq \epsilon$ pour obtenir la réécriture
 $q_1 \cdot (e_n \otimes a \otimes e_m) \stackrel{*}{\underset{M}{|}} (e_n \otimes a \otimes e_m) \cdot q_f$ et $\delta' = h(\delta)$.

en considérant la construction des règles en III.2.8.6. a) et b)

$\Leftrightarrow \exists q_i' \in s(q_i) / q_i' \cdot h(d_1) \stackrel{*}{\underset{M'}{|}} h(\delta_1) \cdot q_1$ et
 $q_1 \cdot h(e_n \otimes a \otimes e_m) \stackrel{*}{\underset{M'}{|}} h(e_n \otimes a \otimes e_m) \cdot q_f$, $\delta' = h(\delta)$

$\Leftrightarrow \exists q_i' \in s(q_i) / q_i' \cdot \delta' \stackrel{*}{\underset{M'}{|}} \delta' \cdot q_f$.

□

II.2.8.8. - Corollaire

REC est clos par homomorphisme alphabétique, non effaçant.

II.2.8.9. - Preuve du corollaire

Un homomorphisme alphabétique non effaçant est un homomorphisme connexe.

□

II.2.9. - Clôture par homomorphisme inverseII.2.9.1. - Propriété

REC est clos par homomorphisme inverse.

II.2.9.2. - Preuve

$\forall h : \check{D}P(\Sigma) \rightarrow \check{D}P(\Sigma')$, homomorphisme de magmaïde et $\forall D' \in \text{REC}$ sur Σ' , il existe effectivement un automate M sur Σ tel que $T(M) = h^{-1}(D')$.

II.2.9.3. - Notations

Notons $M = \langle \Sigma, Q, I, F, R \rangle$ l'automate que l'on cherche à construire tel que $T(M) = h^{-1}(D')$, et $M' = \langle \Sigma', Q', I', F', R' \rangle$ l'automate qui reconnaît D' .

II.2.9.4. - Construction

Seuls l'alphabet et les règles sont modifiés par rapport à l'automate M' et donc : $M = \langle \Sigma, Q', I', F', R' \rangle$ où

$$(q_1 \otimes \dots \otimes q_n) \cdot a \rightarrow a \cdot (q'_1 \otimes \dots \otimes q'_n) \in R$$

⇔

$$(q_1 \otimes \dots \otimes q_n) \cdot h(a) \rightarrow h(a) \cdot (q'_1 \otimes \dots \otimes q'_n)$$

pour $a \in \Sigma_{n'}^n$.

II.2.9.5. - Démonstration

En deux temps, on montre d'abord que la construction est effective, ensuite que c'est bien celle attendue, i.e. $T(M) = h^{-1}(D')$.

a) La construction est effective se déduit directement de :

- Q' et $h(\Sigma)$ étant des ensembles finis, l'ensemble des possibilités de réécritures $(q_1 \otimes \dots \otimes q_n) \cdot h(a) \stackrel{*}{\underset{M'}{\mid}} h(a) \cdot (q'_1 \otimes \dots \otimes q'_n)$, où $a \in \Sigma$, est fini.

- il est effectif de déterminer si une réécriture donnée est une réécriture de M' .

b) La construction proposée est bien celle attendue, se montre directement après avoir montré.

II.2.9.6. - Lemme

$\forall \omega \in \hat{D}P(\Sigma), \forall q_i \in Q'^{\otimes}, \forall q_f \in Q'^{\otimes}$, on a : $q_i \cdot \omega \stackrel{*}{\underset{M'}{\mid}} \omega \cdot q_f$
 $\Leftrightarrow q_i \cdot h(\omega) \stackrel{*}{\underset{M'}{\mid}} h(\omega) \cdot q_f$.

il vient

$\omega \in T(M) \Leftrightarrow \exists q_i \in I', \exists q_f \in F'$ tels que $q_i \cdot \omega \stackrel{*}{\underset{M'}{\mid}} \omega \cdot q_f$
 en utilisant le lemme

$\Leftrightarrow \exists q_i \in I', \exists q_f \in F'$ tels que $q_i \cdot h(\omega) \stackrel{*}{\underset{M'}{\mid}} h(\omega) \cdot q_f$
 en remarquant que les automates M et M' ont même contrôle initial et même contrôle final

$\Leftrightarrow h(\omega) \in D'$

$\Leftrightarrow \omega \in h^{-1}(D')$.

□

Preuve du lemme :

Par récurrence sur le nombre, ℓ , d'occurrences de lettres dans ω .

$\ell = 0 \Rightarrow q_i = q_f$ et $\omega = e_p \Leftrightarrow q_i = q_f$ et $h(\omega) = e_p$, le lemme se vérifie.

Hypothèse de récurrence :

$\forall r \in \mathbb{N}^*$, le lemme est vérifié pour $\omega \in \hat{DP}(\Sigma)$ tel que le nombre d'occurrences de lettres dans ω soit strictement inférieur à r .

$\ell = r \Rightarrow \omega = \omega_1 \cdot (e_i \otimes a \otimes e_j)$ où le nombre d'occurrences de lettres dans ω_1 est $\ell-1$.

$$\Rightarrow q_i \cdot \omega \Big|_{M}^* \omega \cdot q_f$$

$$\Leftrightarrow q_i \cdot \omega \Big|_{M}^* \omega_1 \cdot q \cdot (e_i \otimes a \otimes e_j) \Big|_{M} \omega \cdot q_f$$

en utilisant l'hypothèse de récurrence

$$\Leftrightarrow q_i \cdot h(\omega) \Big|_{M'}^* h(\omega_1) \cdot q \text{ et il existe } (q_1 \otimes \dots \otimes q_n) \cdot a \rightarrow a \cdot (q'_1 \otimes \dots \otimes q'_n) \in R / q = u \otimes q_1 \otimes \dots \otimes q_n \otimes v \text{ et } q_f = u \otimes q'_1 \otimes \dots \otimes q'_n \otimes v \text{ en considérant la construction de } M'$$

$$\Leftrightarrow q_i \cdot h(\omega) \Big|_{M'}^* h(\omega_1) \cdot q \cdot (e_i \otimes a \otimes e_j) \Big|_{M'} \omega \cdot q_f$$

$$\Leftrightarrow q_i \cdot h(\omega) \Big|_{M'}^* h(\omega) \cdot q_f.$$

□

II.2.10. - Problème ouvert : clôture de REC par complémentaire

Nous avons vu en [II.1.6.3. et II.1.6.4.] qu'il existe des automates non-déterministes pour lequel il n'existe pas d'automate déterministe, ni descendant, ni ascendant équivalent, (i.e. définissant le même taillis). Ceci nous interdit donc de reprendre l'idée de la démonstration de la clôture par complémentaire dans les arbres [] qui est d'utiliser la réductibilité du non-déterminisme pour les automates d'arbres, ascendant. On peut néanmoins établir à partir de cette constatation le résultat suivant :

Les taillis reconnus par un automate déterministe (faiblement et donc aussi fortement) et leurs complémentaires sont reconnaissables.

Esquisse de preuve :

Soit $M = \langle \Sigma, Q, I, F, R \rangle$, l'automate initial, on peut le construire totalement spécifié, i.e. où toutes les reconnaissances aboutissent ou plus formellement :

$$\forall m \in Q^*, \forall a \in \Sigma, \exists \text{ une règle telle sa partie gauche soit } m \cdot a.$$

La décision de l'appartenance d'un dag au taillis $T(M)$ est fonction du contrôle final. Ainsi l'automate $M' = \langle \Sigma, Q, I, F', R \rangle$, où F' est le complémentaire de F , reconnaît le complémentaire de $T(M)$, et de plus M' est aussi faiblement déterministe.

Exemple 1 :

Les finis et les co-finis (les finis sont ici des sous-ensembles *finis* de dags planaires, et les co-finis, leurs complémentaires) sont reconnaissables.

En effet, tout fini peut être reconnu par un automate faiblement déterministe.

Exemple 2 :

Les taillis, dont les dags possèdent au moins une occurrence d'une lettre donnée, et leurs complémentaires sont reconnaissables.

En effet, l'automate reconnaissant de tels taillis peut être faiblement déterministe. Le principe de sa construction est d'utiliser deux états q_0 et q_1 . L'état q_0 indique que jusqu'à présent aucune occurrence de la lettre donnée n'a été reconnue et q_1 , le contraire. Ainsi, soit $\begin{array}{c} \diagup \\ a \\ \diagdown \end{array}$, la lettre à chercher, on aura :

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

- $\forall \ell \in \Sigma \setminus \{a\} : (q_0 \otimes \dots \otimes q_0) \cdot \ell \rightarrow \ell \cdot (q_0 \otimes \dots \otimes q_0)$

- $(q_0 \otimes q_0) \cdot \underset{\diagup \quad \diagdown}{\underset{\diagup \quad \diagdown}{a}} \rightarrow \underset{\diagup \quad \diagdown}{\underset{\diagup \quad \diagdown}{a}} \cdot (q_1 \otimes q_0 \otimes q_0) : \text{reconnaissance de la lettre cherchée et génération d'un état } q_1 \text{ (1 seul suffit).}$

- $\forall \ell' \in \Sigma, \forall m \in Q^{\otimes} \{q_0\}^{\otimes}, \text{ i.e. } m \neq (q_0 \otimes \dots \otimes q_0), \text{ alors}$
 $m \cdot \ell' \rightarrow \ell' \cdot (q_1 \otimes q_0 \otimes \dots \otimes q_0) : \text{propagation du fait que la lettre } \underset{\diagup \quad \diagdown}{\underset{\diagup \quad \diagdown}{a}} \text{ a déjà été reconnue}$

- le contrôle initial $I = \{q_0\}^{\otimes}$

- le contrôle final $F = Q^{\otimes} \setminus I, \text{ i.e. le langage composé des mots possédant au moins une occurrence de l'état } q_1.$

CHAPITRE III

LES GRAMMAIRES REGULIERES DE DAGS PLANAIRES

PLAN DU CHAPITRE III

III.1. - DEFINITIONS ET EXEMPLES

III.1.1. - Grammaires linéaires à droite

III.1.2. - Grammaires régulières

III.1.3. - Critique des définitions

III.2. - THEOREMES D'IDENTITE DES CLASSES DE TAILLIS
TAILLIS RECONNAISSABLES = TAILLIS REGULIERS =
TAILLIS LINEAIRES A DROITEIII.2.1. - Théorème d'équivalence entre grammaires
régulières et grammaires linéaires à
droiteIII.2.2. - Théorème d'équivalence entre régularité
et reconnaissabilité dans les dags
planaires

III.3. - SCHEMA RECAPITULATIF

III.1. - DEFINITIONS ET EXEMPLESIII.1.1. - Grammaires linéaire à droiteIII.1.1.1. - Définition

Une grammaire linéaire à droite de dags planaires est la donnée d'un triplet $G = \langle \Sigma, V, R \rangle$ où :

a) Σ est un alphabet bigradué de lettres de co-arité et arité > 0 dit *alphabet terminal*.

b) V est un alphabet bigradué, disjoint de Σ , de lettres de co-arité et arité égaux à 1, dit *alphabet de variables* ou *alphabet auxiliaire*.

c) R est un sous-ensemble fini de règles, appelé production ou *règle de production*, inclus dans $V^{\otimes} \times \hat{DP}(\Sigma) \times (V \cup \{e\})^{\otimes}$ de la forme :

$$V_1 \otimes \dots \otimes V_n \rightarrow \delta \cdot (V'_1 \otimes \dots \otimes V'_n)$$

où $\delta \in \hat{DP}(\Sigma)_n^n$, et δ est un *dag connexe*

$$V_i \in V \text{ pour } i = 1, \dots, n$$

$$V'_i \in V \cup \{e\} \text{ pour } i = 1, \dots, n'$$

III.1.1.2. - La relation <<dérive directement de (dans G)>>

La relation <<dérive directement de (dans G)>> est définie dans $\hat{DP}(\Sigma \cup V)$ par : $\forall \delta \in \hat{DP}(\Sigma \cup V), \forall \delta' \in \hat{DP}(\Sigma \cup V)$.

δ' dérive directement de δ dans G , encore écrit $\delta \xrightarrow{G} \delta'$, si et seulement si δ' est obtenu en remplaçant un facteur m de δ par m' tel que $m \rightarrow m' \in R$.

III.1.1.3. - La relation <<dérive de (dans G)>>

La relation <<dérive de (dans G)>> est définie dans $\hat{DP}(\Sigma \cup V)$ comme la clôture réflexive et transitive de la relation \xrightarrow{G} , et est notée $\xrightarrow{*G}$.

III.1.1.4. - Ordre de dérivation

$\forall \delta \in \hat{D}\hat{P}(\Sigma \cup V)$, $\forall \delta' \in \hat{D}\hat{P}(\Sigma \cup V)$, il est classiquement établi que $\delta \xrightarrow[G]{*} \delta'$ si et seulement si il existe une suite finie $\delta_1, \delta_2, \dots, \delta_{k+1}$ de dags planaires sur $\Sigma \cup V$ tels que $\delta_1 = \delta$, $\delta_{k+1} = \delta'$ et

$$\forall i = 1, \dots, k \quad \delta_i \xrightarrow[G]{} \delta_{i+1}$$

Le nombre k est alors appelé *ordre de la dérivation* de u en u' dans G . Il représente le nombre de fois où on a utilisé une règle.

III.1.1.5. - Notations

On note, pour $\delta \in \hat{D}\hat{P}(\Sigma \cup V)$.

$$T(G, \delta) = \{\delta' \in \hat{D}\hat{P}(\Sigma) / \delta \xrightarrow[G]{*} \delta'\}$$

$$\hat{T}(G, \delta) = \{\delta' \in \hat{D}\hat{P}(\Sigma \cup V) / \delta \xrightarrow[G]{*} \delta'\}$$

On étend cette notation à $A \subseteq \hat{D}\hat{P}(\Sigma \cup V)$

$$T(G, A) = \{\delta' \in \hat{D}\hat{P}(\Sigma) / \exists \delta \in A \text{ et } \delta \xrightarrow[G]{*} \delta'\}$$

$$\hat{T}(G, A) = \{\delta' \in \hat{D}\hat{P}(\Sigma \cup V) / \exists \delta \in A \text{ et } \delta \xrightarrow[G]{*} \delta'\}$$

III.1.1.6. - Définition de taillis à partir d'une grammaire linéaire à droite

Dans le cas où A s'identifie à un langage rationnel de mots sur V^\emptyset alors $T(G, A)$ est appelé *taillis engendré* par la grammaire G à partir de l'axiome A (ou du langage rationnel d'axiomes) ; $\hat{T}(G, A)$ est appelé *taillis élargi engendré* par la grammaire G à partir de l'axiome A (ou du langage rationnel d'axiomes).

III.1.1.7. - Taillis linéaires à droite

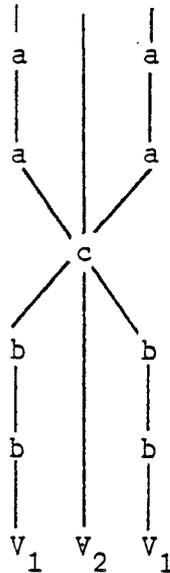
$T \subseteq \hat{DP}(\Sigma)$ est un taillis linéaire à droite si et seulement si il existe une grammaire linéaire à droite $G = \langle \Sigma, V, P \rangle$ et $A \subseteq V^*$, rationnel de mots tels que $T(G, A) = T$. L'ensemble des taillis linéaires à droite est noté LIN_D.

III.1.1.8. - Exemple

Soit la grammaire linéaire à droite $G = \langle \{a, b, c, d\}, \{V_1, V_2\}, R \rangle$ où R est composée de quatre règles.

- 2 règles pour générer

règle 1 : $V_1 \otimes V_2 \otimes V_1 \rightarrow$

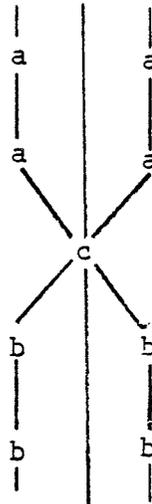


règle 2 : $V_2 \rightarrow d$



- 2 règles pour éliminer les variables, en fin de génération

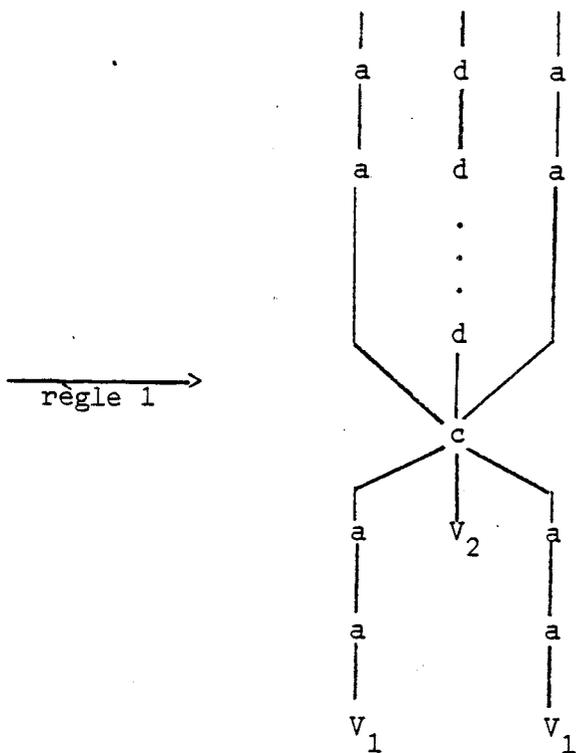
règle 3 : $V_1 \otimes V_2 \otimes V_1 \rightarrow$



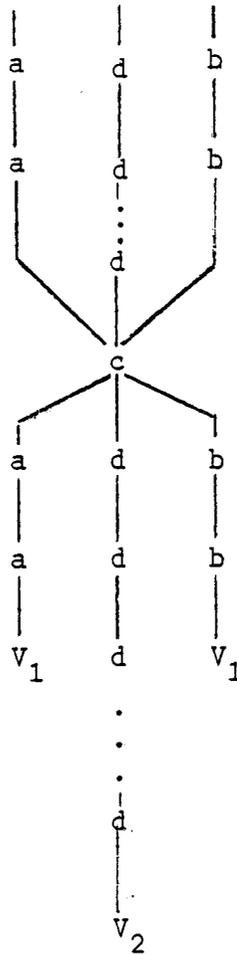
- règle 4 : $V_2 \rightarrow \begin{array}{|c} d \\ \hline \end{array}$

A partir de l'axiome $A = V_1 \otimes V_2 \otimes V_1$, nous aurons une succession de séquences de réécritures de la forme :

$$V_1 \otimes V_2 \otimes V_2 \xrightarrow[\text{(règle 2, plusieurs fois)}]{*} V_1 \otimes \begin{array}{|c} d \\ \hline \end{array} \otimes V_1$$

$$\begin{array}{|c} d \\ \hline \vdots \\ \hline d \\ \hline V_2 \\ \hline \end{array}$$


(règle 2, plusieurs fois) →



En répétant la séquence de réécritures, ci-dessus, puis en utilisant les règles 3 et 4 pour éliminer les variables, nous mettrons en évidence que $T(G, A)$ est le taillis de dags planaires de la forme d'une répétition du motif ci-dessus, où le nombre d'occurrences de lettres d est arbitraire.

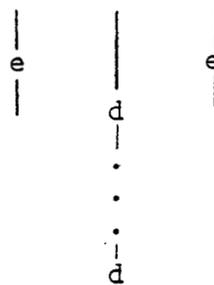
Exemple :



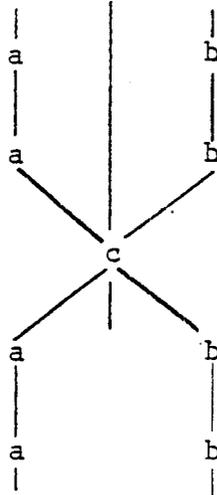
le nombre de d est arbitraire

Remarque :

Une autre manière de décrire $T(G, A)$ est de le considérer comme l'ensemble des dags de la forme



tant et autant de fois que l'on veut le motif



III.1.2. - Grammaires régulières

III.1.2.1. - Définition

Une grammaire régulière de dags planaires est la donnée d'un triplet $G = \langle \Sigma, V, R \rangle$ où

a) G est une grammaire linéaire à droite de dags planaires

b) $R \subseteq V^{\otimes} \otimes \Sigma \cup \{e\} \times (V \cup \{e\})^{\otimes}$ de la forme

$$\bullet \text{ soit } V_1 \otimes \dots \otimes V_n \rightarrow a \cdot V'_1 \otimes \dots \otimes V'_{n'}$$

$$\text{où } a \in \Sigma_n^n,$$

$$V_i \in V \text{ pour } i = 1, \dots, n$$

$$V'_i \in V \cup \{e\} \text{ pour } i = 1, \dots, n'$$

$$\bullet \text{ soit } V_1 \otimes \dots \otimes V_n \rightarrow e_n$$

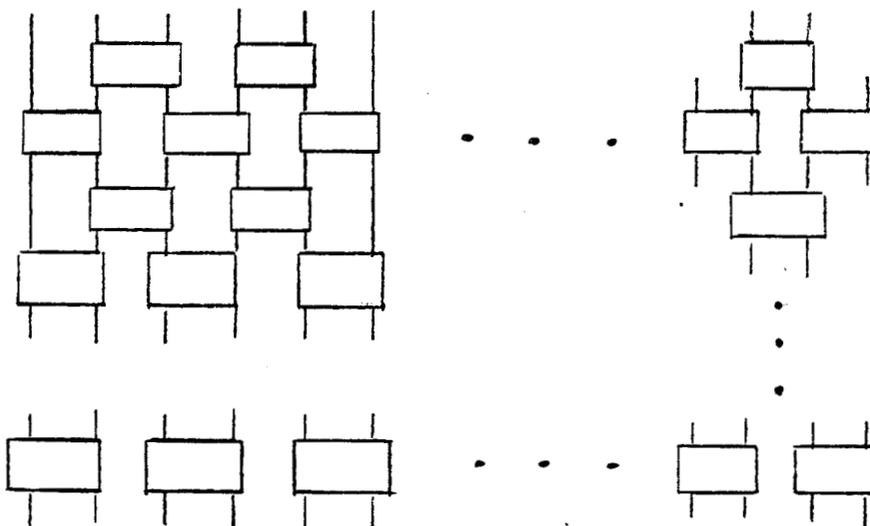
$$\text{où } V_i \in V \text{ pour } i = 1, \dots, n$$

III.1.2.2. - Taillis réguliers

$T \subseteq \overset{\vee}{DP}(\Sigma)$ est un taillis régulier si et seulement si il existe une grammaire régulière $G = \langle \Sigma, V, R \rangle$ et $A \subseteq V^{\otimes}$, rationnel de mots tels que $T(G, A) = T$. L'ensemble des taillis réguliers est noté REG.

III.1.2.3. - Exemple

L'ensemble des dags planaires de la forme de <<mur de toutes largeurs et de toutes longueurs>> est un taillis régulier



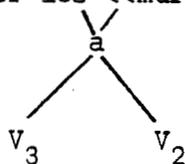
Il peut être généré à partir de l'axiome A

$$A = V_1 \otimes (V_2 \otimes V_3)^{\otimes} \otimes V_1$$

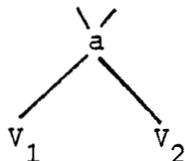
par la grammaire régulière $G = \langle \{a\}, \{V_1, V_2, V_3\}, R \rangle$ où R est composé des six règles :

- 3 règles pour générer les <<murs>> :

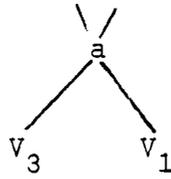
règle 1 : $V_2 \otimes V_3 \rightarrow$



règle 2 : $V_1 \otimes V_3 \rightarrow$



règle 3 : $V_2 \otimes V_1 \rightarrow$



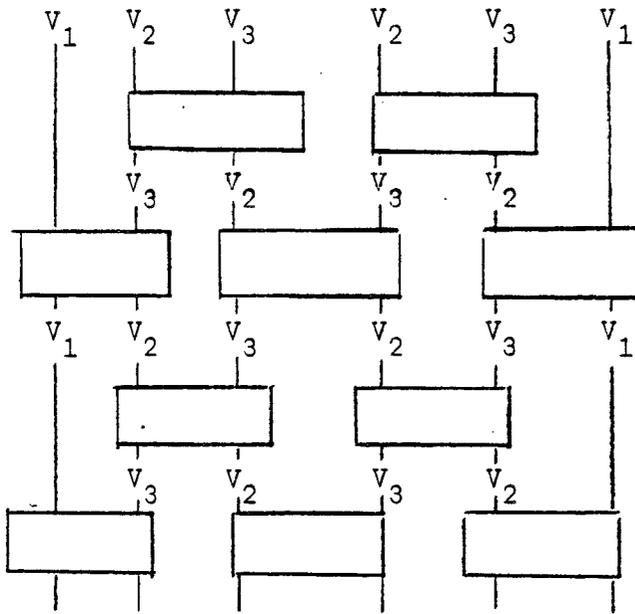
- 3 règles pour faire disparaître les variables et rendre les dags générés élément du taillis engendré :

règle 4 : $V_2 \otimes V_3 \rightarrow$

règle 5 : $V_1 \otimes V_3 \rightarrow$

règle 6 : $V_2 \otimes V_1 \rightarrow$

On peut schématiser une génération :



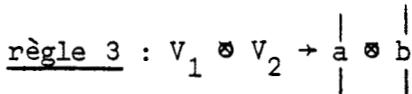
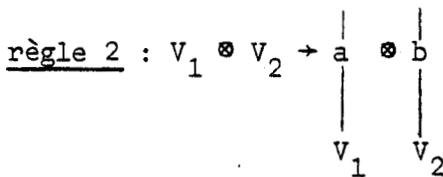
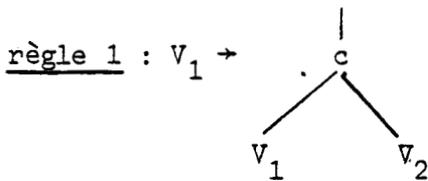
III.1.3. - Critique des définitions

La critique des définitions portera principalement sur celles relatives aux grammaires linéaires à droite, les grammaires régulières étant un cas particulier de grammaires linéaires à droite.

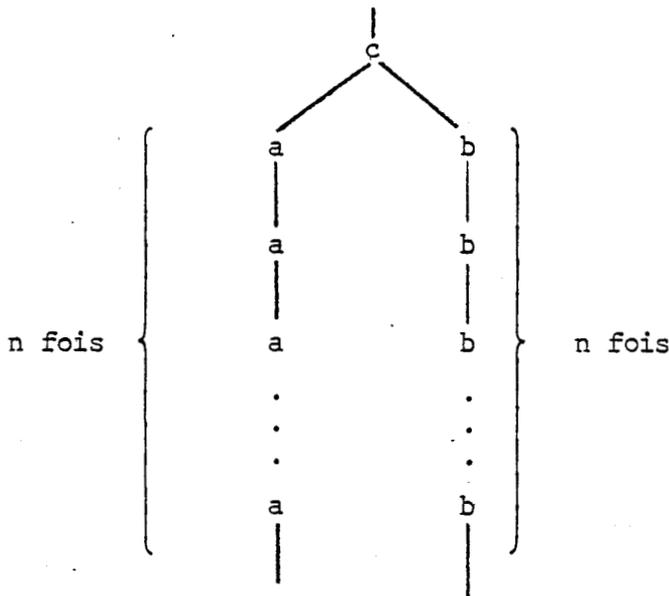
III.1.3.1. - Sur la forme des règles (génération de dags connexes)

Il semble naturel de retrouver, pour les dags planaires, la propriété $LIN_D = REC$ que l'on possède déjà dans les mots. Ainsi, imposer qu'un dag, généré par une règle, soit connexe s'avère essentiel. Dans le cas contraire, on pourrait engendrer des taillis non reconnaissables. Exemple :

$G = \langle \{a, b, c\}, \{V_1, V_2\}, R \rangle$, R définit par :



où les règles 2 et 3 génèrent chacune des dags non connexes. Alors $T(G, \{V_1\})$ est l'ensemble des dags de la forme :



et $T(G, V_1)$, dans ce cas, n'est pas un taillis reconnaissable car ce n'est pas une forêt reconnaissable.

Remarque :

Les règles de la forme $V \rightarrow e$, où V est une variable, sont admises comme règles régulières de grammaire de dags planaires, alors que dans les mots, toutes règles régulières réécrit *au moins une lettre*. Ainsi on aurait pu restreindre la définition des règles régulières de dags planaires à être inclus dans $V^{\otimes} \times \Sigma \times (V \cup \{e\})^{\otimes}$ et laisser les autres contraintes telles quelles.

Mais dans ce cas, il aurait fallu changer, d'une manière duale, la nature de l'axiome en acceptant qu'il soit inclus dans $(A \cup \{e\})^{\otimes}$ au lieu de A^{\otimes} , car sinon le dag $\{a \otimes e\}$ ne serait pas régulier.

D'entre les deux choix :

1. $R \subseteq V^{\otimes} \times (\Sigma \cup \{e\}) \times (V \cup \{e\})^{\otimes}$ et l'axiome $A \subseteq V^{\otimes}$

2. $R \subseteq V^{\otimes} \times \Sigma \times (V \cup \{e\})^{\otimes}$ et l'axiome $A \subseteq (V \cup \{e\})^{\otimes}$

Notre choix s'est porté sur la première possibilité, ce qui ne change fondamentalement rien.

III.1.3.2. - Sur la nature de l'axiome (langage rationnel de mots sur V)

III.1.3.2.1. - La nature, langage rationnel de mots de variables, de l'axiome pour les grammaires de dags planaires, rompt avec la nature de l'axiome définit pour les grammaires de mots ou d'arbres, où l'axiome est une variable. Cette différence de nature des axiomes correspond à la différence de dimension des objets considérés :

III.1.3. - Critique des définitions

La critique des définitions portera principalement sur celles relatives aux grammaires linéaires à droite, les grammaires régulières étant un cas particulier de grammaires linéaires à droite.

III.1.3.1. - Sur la forme des règles (génération de dags connexes)

Il semble naturel de retrouver, pour les dags planaires, la propriété $LIN_D = REC$ que l'on possède déjà dans les mots. Ainsi, imposer qu'un dag, généra par *une* règle, soit connexe s'avère essentiel. Dans le cas contraire, on pourrait engendrer des taillis non reconnaissable. Exemple :

$G = \langle \{a, b, c\}, \{V_1, V_2\}, R \rangle$, R définit par :

règle 1 : $V_1 \rightarrow$

```

      |
      c
     / \
    V_1 V_2
  
```

règle 2 : $V_1 \otimes V_2 \rightarrow$

```

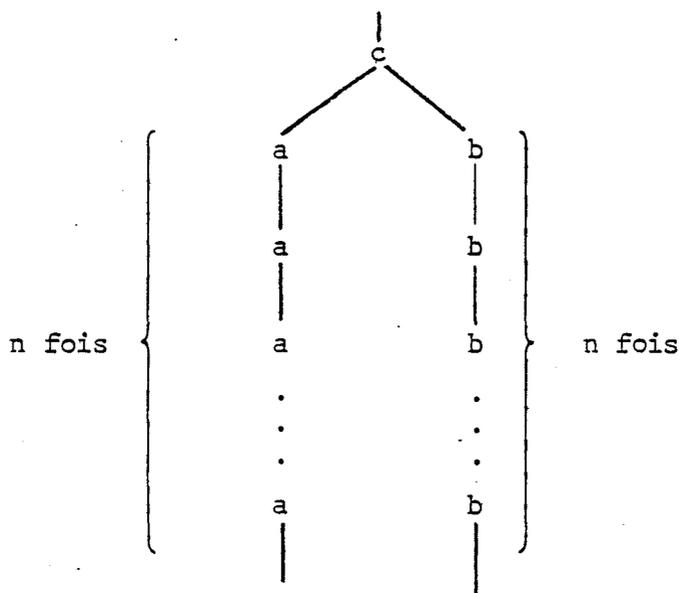
      |   |
      a   b
      |   |
      V_1 V_2
  
```

règle 3 : $V_1 \otimes V_2 \rightarrow$

```

      |   |
      a   b
      |   |
      |   |
  
```

où les règles 2 et 3 génèrent chacune des dags non connexes. Alors $T(G, \{V_1\})$ est l'ensemble des dags de la forme :



et $T(G, V_1)$, dans ce cas, n'est pas un taillis reconnaissable car ce n'est pas une forêt reconnaissable.

Remarque :

Les règles de la forme $V \rightarrow e$, où V est une variable, sont admises comme règles régulières de grammaire de dags planaires, alors que dans les mots, toutes règles régulières réécrit *au moins une lettre*. Ainsi on aurait pu restreindre la définition des règles régulières de dags planaires à être inclus dans $V^{\otimes} \times \Sigma \times (V \cup \{e\})^{\otimes}$ et laisser les autres contraintes telles quelles.

Mais dans ce cas, il aurait fallu changer, d'une manière duale, la nature de l'axiome en acceptant qu'il soit inclus dans $(A \cup \{e\})^{\otimes}$ au lieu de A^{\otimes} , car sinon le dag $\{a \otimes e\}$ ne serait pas régulier.

D'entre les deux choix :

1. $R \subseteq V^{\otimes} \times (\Sigma \cup \{e\}) \times (V \cup \{e\})^{\otimes}$ et l'axiome $A \subseteq V^{\otimes}$
2. $R \subseteq V^{\otimes} \times \Sigma \times (V \cup \{e\})^{\otimes}$ et l'axiome $A \subseteq (V \cup \{e\})^{\otimes}$

Notre choix s'est porté sur la première possibilité, ce qui ne change fondamentalement rien.

III.1.3.2. - Sur la nature de l'axiome (langage rationnel de mots sur V)

III.1.3.2.1. - La nature, langage rationnel de mots de variables, de l'axiome pour les grammaires de dags planaires, rompt avec la nature de l'axiome définit pour les grammaires de mots ou d'arbres, où l'axiome est une variable. Cette différence de nature des axiomes correspond à la différence de dimension des objets considérés :

- un mot est un objet de dimension 1 : un seul opérateur, la concaténation.
- un arbre est un objet de dimension 1-2 : deux opérateurs, le produit de composition et le produit tensoriel, mais la racine d'un arbre est réduite à *une* lettre de co-arité 1.
- un dag planaire est un objet de dimension 2 : deux opérateurs, les mêmes que pour les arbres, mais pas de contrainte sur le "haut" des objets.

III.1.3.2.2. - Cependant, en remarquant qu'un langage rationnel de mots peut être engendré par une grammaire régulière de mots. On peut conjecturer que la proposition de la définition de grammaire de dags planaires, ci-dessous, permet de définir, à partir d'un axiome réduit à une variable, exactement les mêmes taillis que nos grammaires linéaires à droite à partir d'un axiome, langage rationnel de mots de variables.

Proposition de définition :

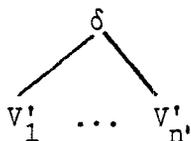
$G = \langle \Sigma, H, V, R \rangle$ où

- a) Σ est un alphabet bigradué de lettres de co-arité et arité > 0 , dit terminal.
- b) H est un alphabet bigradué de lettres de co-arité et arité égaux à 1, disjoint de Σ .
- c) V est un alphabet bigradué de lettres de co-arité et arité égaux à 1, disjoint de Σ et de H .
- d) R est un ensemble fini de règles de la forme

soit $A \rightarrow V_1 \otimes A'$

où $A \in H$; $A' \in H \cup \{e_0\}$ et $V_1 \in V$

soit $V_1 \otimes \dots \otimes V_n \rightarrow$



où $\delta \in \overset{\vee}{\text{DP}}(\Sigma)_n^n$, et δ est un dag connexe

$V_i \in V$, pour $i = 1, \dots, n$

$V'_i \in V \cup \{e\}$, pour $i = 1, \dots, n'$

Les règles de R de la forme $A \rightarrow V_1 \otimes A'$ étant destinées à générer le langage rationnel de mots de variables qui sert d'axiome, et ensuite les règles de la forme $V_1 \otimes \dots \otimes V_n \rightarrow \delta \cdot (V'_1 \otimes \dots \otimes V'_{n'})$ étant destinées à générer le taillis proprement dit.

La définition [III.1.1.] a été préférée pour des raisons de simplicité d'écriture.

Remarque :

Dans la proposition de définition ci-dessus, la génération du langage rationnel de mots sur V , destiné à servir d'axiome est faite indépendamment de la génération du taillis proprement dit. Pour avoir les deux générations imbriquées, et ainsi peut-être avoir une définition, de taillis, plus modulaire, on peut étendre R en admettant aussi les règles à

$$V_1 \otimes \dots \otimes V_n \otimes A \rightarrow \begin{array}{c} \delta \\ \swarrow \quad \searrow \\ V'_1 \quad \dots \quad V'_{n'} \end{array} \otimes A'$$

où $\delta \in \overset{\vee}{\text{DP}}(\Sigma)_n^n$, et δ est un dag connexe

$V_i \in V$ pour $i = 1, \dots, n$

$A \in H$

$V'_i \in V \cup \{e\}$ pour $i = 1, \dots, n'$

$A' \in H \cup \{e_0\}$.

Dans la suite, nous ne considérerons que la définition donnée initialement [III.1.1.].

III.2. - THEOREMES D'IDENTITE DES CLASSES DE TAILLIS : TAILLIS
RECONNAISSABLES = TAILLIS REGULIERS = TAILLIS LINEAIRES
A DROITE

III.2.1. - Théorème d'équivalence entre grammaire régulière et
grammaire linéaire à droite

Toute grammaire régulière est par définition une grammaire linéaire à droite. Nous allons montrer que, inversement, toute règle linéaire à droite peut se décomposer en un ensemble de règles régulières. Ce qui nous permet d'écrire le théorème suivant :

III.2.1.1. - Enoncé

Dans les dags planaires, $REG = LIN_D$.

III.2.1.2. - Preuve

(i) Toute grammaire régulière est par définition une grammaire linéaire à droite, et donc $REG \subseteq LIN_D$.

(ii) $\forall G = \langle \Sigma, V, R \rangle$, grammaire linéaire à droite, il existe effectivement $G' = \langle \Sigma, V \cup V', R' \rangle$, grammaire régulière telle que $\forall A \subseteq V^*$, $T(G, A) = T(G', A)$. Et donc $LIN_D \subseteq REG$.

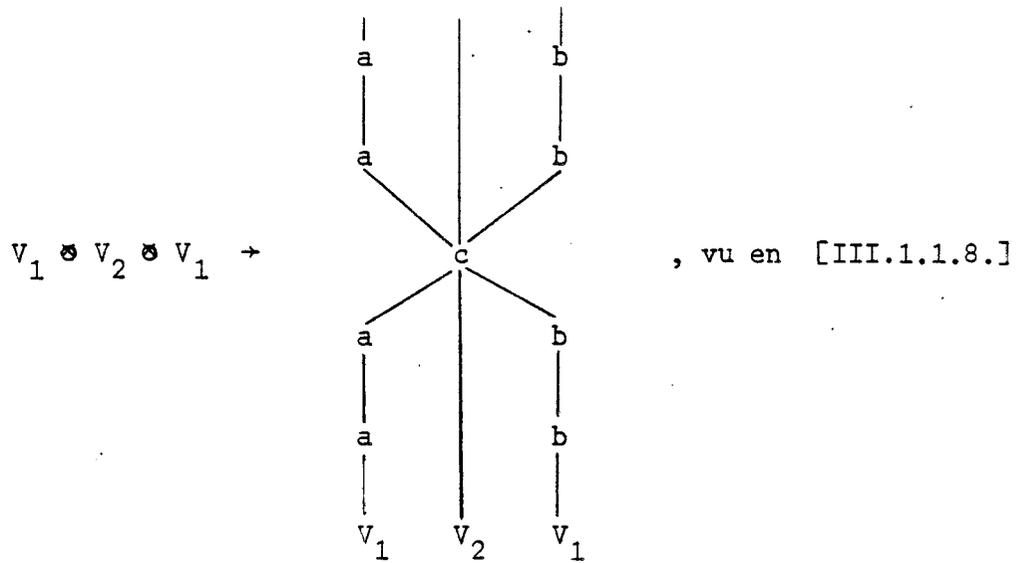
Le (i) ne nécessitant pas de démonstration, regardons celle de (ii).

III.2.1.3. - Idée de construction de (ii)

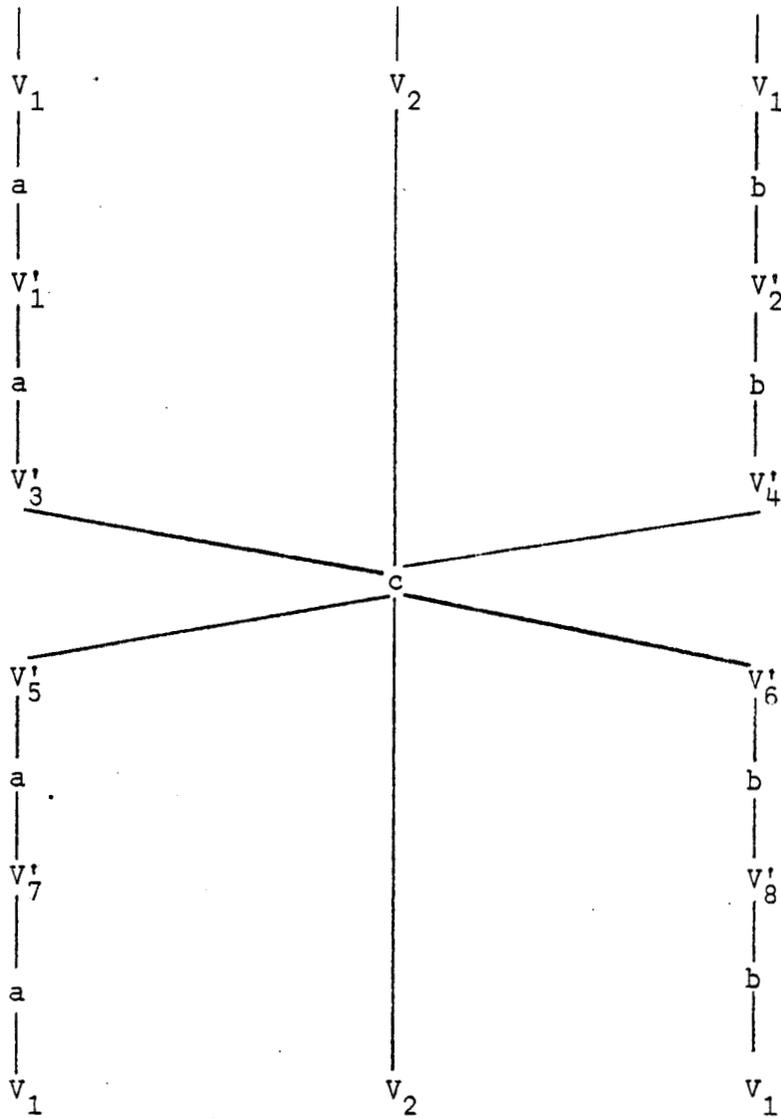
La seule différence entre ces deux types de grammaire est qu'une règle d'une grammaire linéaire à droite peut générer, en une seule réécriture un dag connexe, tandis qu'une règle régulière ne peut générer qu'une lettre. Nous allons donc remplacer chaque règle $r \in R$, par un ensemble, R_r , de règles régulières simulant cette règle r . Afin que les nouvelles règles produites n'interfèrent pas entre-elles, il s'avèrera nécessaire de définir chaque ensemble de règles R_r sur $(V_r \cup V)^* \rightarrow \Sigma \cdot (V_r \cup V \cup \{e\})^*$, où pour $r \in R$, les ensembles de variables sont deux à deux disjoints, et disjoints aussi de V . Pour la règle r , V_r est appelé ensemble de variables intermédiaires pour simuler la règle r .

III.2.1.4. - Exemple

Soit la règle linéaire à droite :



nous allons la remplacer par neuf règles régulières, en prenant comme principe d'écrire une règle régulière par occurrences de lettres terminales générées, ici quatre occurrences de a, quatre occurrences de b et une occurrence de c. Les neuf règles à écrire pour simuler la règle considérée, peuvent se schématiser :



où $V'_1, V'_2, V'_3, V'_4, V'_5, V'_6, V'_7, V'_8$ sont de nouvelles variables, ils forment pour la règle simulée, l'ensemble des variables intermédiaire.

III.2.1.5. - Construction

En deux temps :

a) à chaque règle $r \in R : v \rightarrow \delta \cdot v'$, on associe

- un ensemble de variables, V_r , tel que les V_r soient deux à deux disjoints et disjoints de V .

- un ensemble de règles régulières, R_r , définies sur $(V_r \cup V)^\omega \rightarrow \Sigma \cdot (V_r \cup V \cup \{e\})^\omega$ tel que :

$$v_1 \xrightarrow[r]{*} \delta' \cdot v_2 \iff v_1 \xrightarrow[R_r]{*} \delta' \cdot v_2$$

pour $v_1, v_2 \in V^{\otimes}$

$$\delta' \in \hat{DP}(\Sigma).$$

$$b) G' = \langle \Sigma, \bigcup_{r \in R} V_r \cup V, \bigcup_{r \in R} R_r \rangle.$$

III.2.1.6. - Démonstration

En reprenant, III.2.1.3. et III.2.1.4., il est clair que la construction a) est effective. Montrer alors que la grammaire régulière G' est telle que :

$\forall A \subseteq V^{\otimes}, T(G', A) = T(G, A)$ est immédiat :

$\forall \delta \in T(G', A) \iff \exists m \in A$ tel que $m \xrightarrow[G']{*} \delta$, d'après la construction des règles r et R_r , notamment qu'ils engendrent strictement les mêmes dags à partir d'un même tenseur de variables de V .

$$\iff m \xrightarrow[G]{*} \delta \text{ avec } m \in A$$

$$\iff m \in T(G, A).$$

□

III.2.2. - Théorème d'équivalence entre régularité et reconnaissabilité dans les dags planaires

Dans les mots et dans les arbres, nous avons le résultat bien connu que la reconnaissance par automate d'états finis et la génération par grammaire régulière sont deux approches d'une seule et même classe d'objets : les reconnaissables ou réguliers. Nous allons montrer qu'il en est de même dans les dags planaires.

III.2.2.1. - Enoncé

Dans les dags planaires, REC = REG.

C'est-à-dire qu'un taillis est reconnaissable si et seulement s'il est régulier.

III.2.2.2. - Preuve

(i) $\forall M$, automate d'états finis de dags planaires, il existe effectivement G , grammaire régulière de dags planaires et A , langage rationnel de mots tels que :

$$T(M) = T(G, A).$$

(ii) $\forall G$, grammaire régulière de dags planaires et A , langage rationnel de mots, il existe effectivement M , automate d'états finis de dags planaires tel que :

$$T(G, A) = T(M).$$

III.2.2.2.1. - Preuve de (i) (passage automate \rightarrow grammaire)

III.2.2.2.1.1. - Idée de construction

Soit $M = \langle \Sigma, Q, I, F, R \rangle$, on peut supposer [théorème II.1.1.4.] que $F = \{q_f\}^{\otimes}$. Ainsi la construction peut s'expliquer en trois points :

- a) Identifier les variables de la grammaire aux états de l'automate, ainsi on peut associer à toute règle de l'automate, une règle de la grammaire.
- b) Identifier l'état terminal de l'automate à un effacement de variable dans la grammaire, c'est-à-dire ajouter la règle $q_f \rightarrow e$.
- c) Prendre comme axiome le contrôle haut de l'automate.

III.2.2.2.1.2. - Construction

Plus formellement, on a :

$$G : \langle \Sigma, Q, RG \rangle \text{ où } RG = R \cup \{q_f \rightarrow e\}$$

$$A = I$$

III.2.2.2.1.3. - Démonstration

Immédiate,

• en constatant que la construction de la grammaire régulière est effective

• en montrant :

Lemme :

$$\forall m \in Q^{\otimes}, \forall m' \in Q^{\otimes}, \forall \delta \in \hat{D}\hat{P}(\Sigma) :$$

$$m \cdot \delta \xrightarrow{M}^* \delta \cdot m' \iff m \xrightarrow{G}^* \delta \cdot m'$$

et de plus si $m' \in \{q_f\}^{\otimes}$ alors

$$m \cdot \delta \xrightarrow{M}^* \delta \cdot m' \iff m \xrightarrow{G}^* \delta.$$

Les justifications formelles de la démonstration et du lemme sont évidentes.

□

III.2.2.2.1.4. - Exemple

Soit l'automate $M = \langle \{a, b\}, \{q_1, q_f\}, I = \{q_1\}, F = \{q_f\}, R \rangle$ où R est l'ensemble des transitions :

$$q_1 \cdot a \vdash a \cdot (q_1 \otimes q_1)$$

$$q_1 \cdot a \vdash a \cdot (q_f \otimes q_f)$$

$$(q_f \otimes q_f) \cdot b \vdash b \cdot q_f$$

En appliquant, la construction proposée, on obtient la grammaire régulière $G = \langle \Sigma, Q, RG \rangle$ et l'axiome $A = I = \{q_1\}$ où RG .

$$q_1 \rightarrow a \cdot (q_1 \otimes q_1)$$

$$q_1 \rightarrow a \cdot (q_f \otimes q_f)$$

$$q_f \otimes q_f \rightarrow b \cdot q_f$$

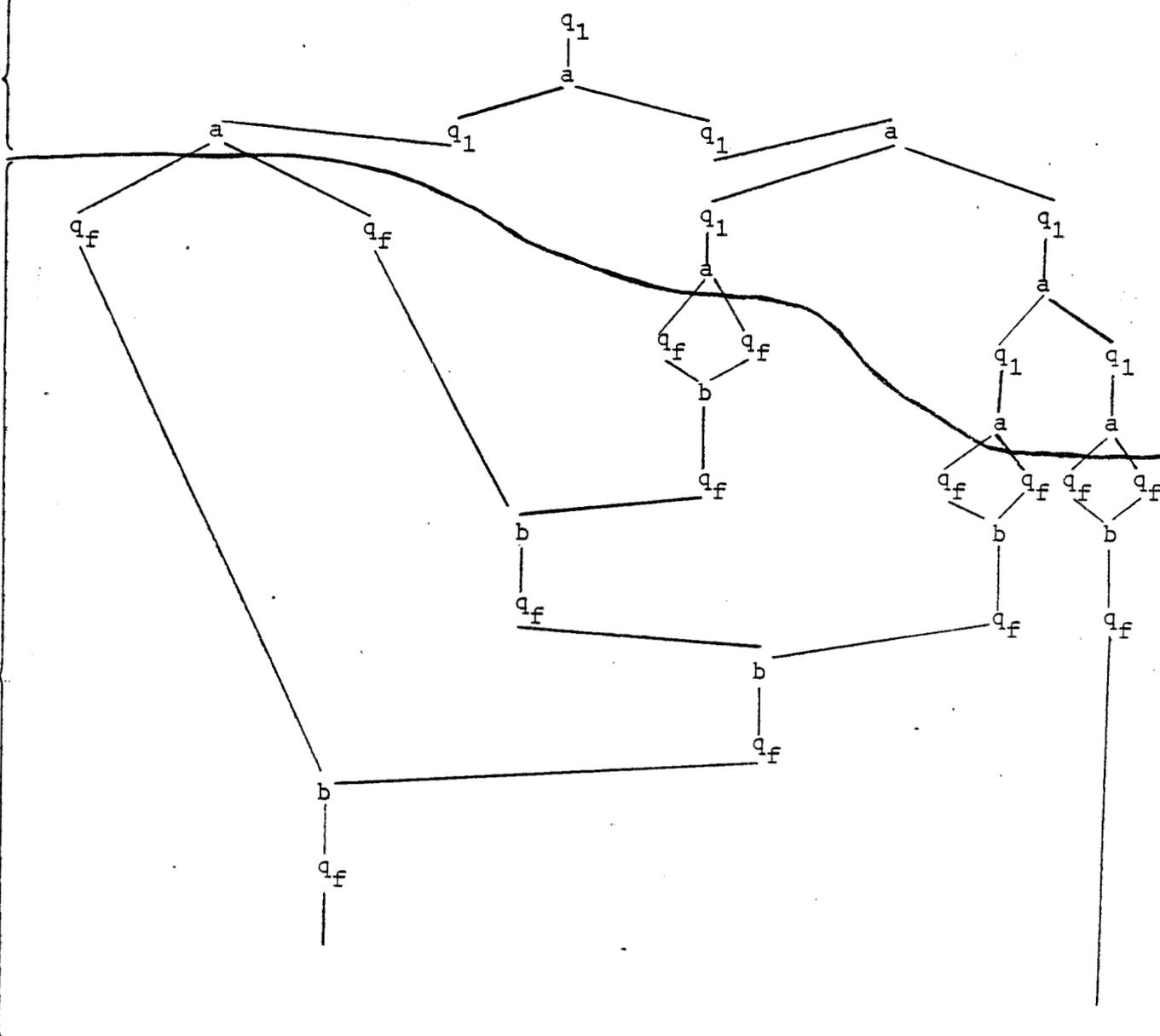
$$q_f \rightarrow e$$

a) Les dags reconnus par l'automate peuvent être générés par la grammaire, exemple, pour la reconnaissance :

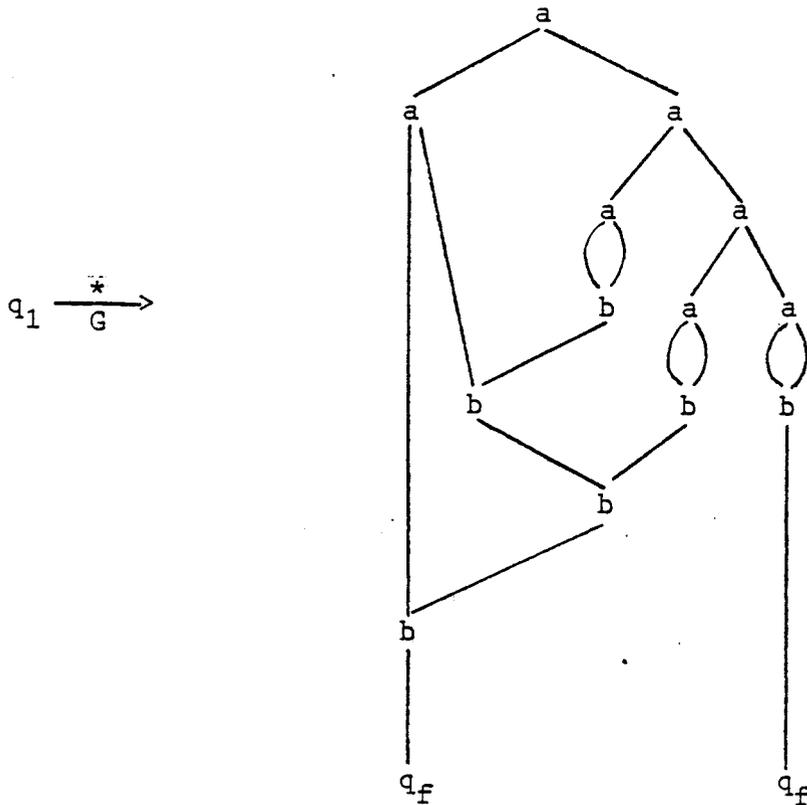
ère par-
ie formée



ème par-
e formée

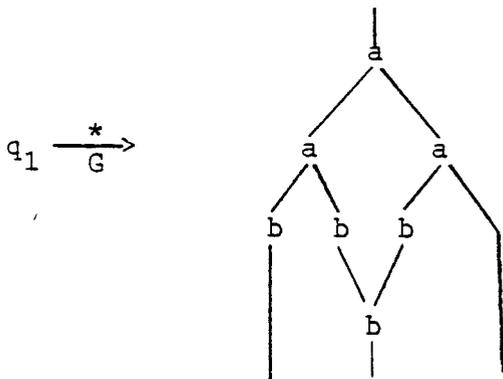


• Au formalisme près, R étant inclus dans RG , on a forcément

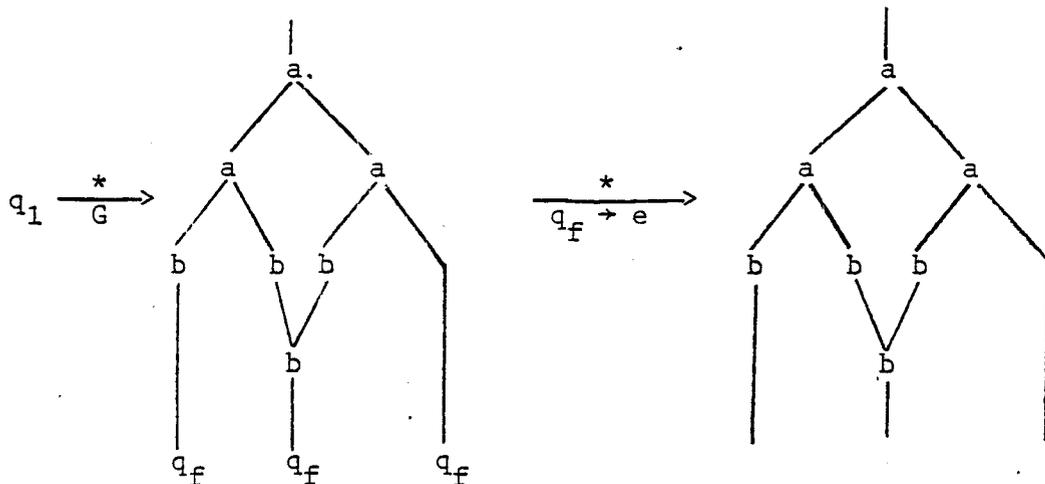


De plus, en utilisant deux fois la règle $q_f \rightarrow \epsilon$, on obtient le dag reconnu précédemment.

b) Inversement, tout dag généré par la grammaire régulière peut être reconnu par l'automate ; exemple :



peut se décomposer en deux parties.



La première partie est faite en utilisant la partie $R \subseteq RG$, la seconde partie grâce à la règle $q_f \rightarrow e$. Il est facile alors d'associer à la première partie de la génération, une reconnaissance acceptée par l'automate M .

III.2.2.2.2. - Preuve de (ii) (passage grammaire régulière \rightarrow automate)

III.2.2.2.2.1. - Idée de construction

Nous allons utiliser une construction symétrique à la précédente. Cette construction peut, elle aussi, s'expliquer en trois points.

a) Identifier les états de l'automate aux variables de la grammaire, ainsi on peut associer à toute règle de la grammaire dont la partie droite n'"efface" pas de variable, une transition de l'automate.

b) Introduire un nouvel état : q_f et pour chaque règle de G où apparaît soit e , soit une variable qui peut s'effacer, ajouter les règles de transitions identiques à la règle de B concernée mais

- en remplaçant chaque e par q_f (sauf $V_1 \rightarrow e$)
- en donnant la possibilité de faire apparaître q_f là où peut disparaître une variable, exemple :

1. Pour la règle $(V_1 \otimes V_2) \rightarrow a \cdot (V_1' \otimes e \otimes e)$, on ajoutera la transition

$$(V_1 \otimes V_2) \cdot a \rightarrow a \cdot (V_1' \otimes q_f \otimes q_f).$$

2. Pour les règles $(V_1 \otimes V_2) \rightarrow a \cdot (V'_1 \otimes V'_2 \otimes V'_3)$

$$V'_2 \rightarrow e$$

$$V'_3 \rightarrow e$$

On ajoutera à la transition $(V_1 \otimes V_2) \cdot a \rightarrow a \cdot (V'_1 \otimes V'_2 \otimes V'_3)$ déjà lors de l'identification des états aux variables, et les trois transitions

$$(V_1 \otimes V_2) \cdot a \rightarrow a \cdot (V'_1 \otimes q_f \otimes q_f)$$

$$(V_1 \otimes V_2) \cdot a \rightarrow a \cdot (V'_1 \otimes V'_2 \otimes q_f)$$

$$(V_1 \otimes V_2) \cdot a \rightarrow a \cdot (V'_1 \otimes q_f \otimes V'_3)$$

c) Prendre comme contrôle haut : l'axiome A mais comme en b) : chaque fois que pourra apparaître, grâce aux règles $v \rightarrow e$, l'élément e , on fera apparaître q_f , dans le contrôle haut.

III.2.2.2.2.2. - Construction

Plus formellement : soit $G = \langle \Sigma, V, RG \rangle$ pour l'axiome A on a :

$$M = \langle \Sigma, V \cup \{q_f\}, s^{-1}(A), \{q_f\}^{\otimes}, R \rangle \text{ où } q_f \notin V \text{ et}$$

$$R = \{(V_1 \otimes \dots \otimes V_n) \cdot a \rightarrow a \cdot s^{-1}(V'_1 \otimes \dots \otimes V'_n)\} \text{ tel que}$$

$$V_1 \otimes \dots \otimes V_n \rightarrow a \cdot (V'_1 \otimes \dots \otimes V'_n) \in RG\}$$

où

$s : (V \cup \{q_f\})^{\otimes} \rightarrow P(V \cup \{e\})^{\otimes}$, substitution finie définie par :

$$s(v) = \{v, q_f\} \iff v \in V \text{ et } v \rightarrow e \in R$$

$$s(v) = \{v\} \iff v \in V \text{ et } v \rightarrow e \notin R$$

$$s(q_f) = \{q_f\}$$

III.2.2.2.2.3. - Démonstration

Immédiatement,

• en constatant que la construction de l'automate d'états finis proposée est effective

• en montrant que

Lemme

$\forall m \in (V \cup \{q_f\})^*$, $\forall m' \in (V \cup \{q_f\})^*$, $\forall \delta \in \tilde{D}P(\Sigma)$:

$$m \cdot \delta \xrightarrow[M]{*} \delta \cdot m' \Leftrightarrow s^{-1}(m) \xrightarrow[G]{*} \delta \cdot s^{-1}(m')$$

où s est la substitution finie définie précédemment.

Les justifications formelles sont évidentes.

□

III.2.2.2.2.4. - Exemple

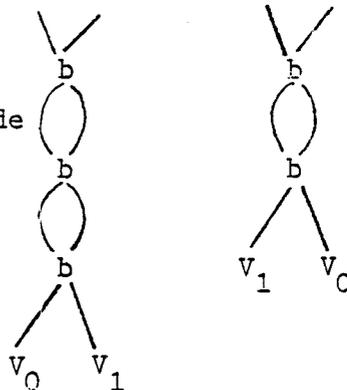
a) Considérons la grammaire régulière $G = \langle b, \{V_0, V_1, V_2\}, RG \rangle$ et l'axiome $A = \{V_0 \otimes V_1 \otimes V_1 \otimes V_0\}$ où RG est composée

de 5 règles pour générer

$$V_0 \otimes V_1 \rightarrow b \cdot (V_0 \otimes V_1)$$

$$V_1 \otimes V_0 \rightarrow b \cdot (V_1 \otimes V_0)$$

: génération de



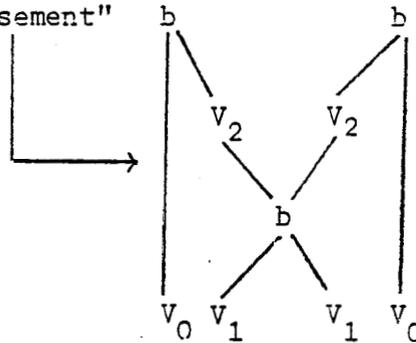
$$V_0 \otimes V_1 \otimes b \cdot (V_0 \otimes V_2)$$

: transformation des V_1 en V_2 , pour

$$V_1 \otimes V_0 \otimes b \cdot (V_2 \otimes V_0)$$

préparer un "croisement"

$$V_2 \otimes V_2 \otimes b \cdot (V_1 \otimes V_1) : \text{"croisement"}$$



et réapparition des V_1 pour revenir à la condition initiale : $V_0 \times V_1 \times V_1 \times V_0$

de 3 règles pour faire disparaître les variables

$$V_0 \otimes V_1 \rightarrow b \cdot (e \otimes V_1)$$

$$V_1 \otimes V_0 \rightarrow b \cdot (V_1 \otimes e)$$

$$V_1 \rightarrow e$$

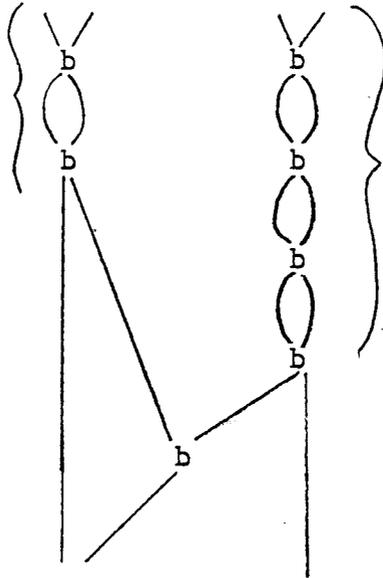
Remarque :

Le choix des 3 règles, ci-dessus, a été fait pour mieux illustrer la construction proposée, il aurait été équivalent de choisir :

$$V_0 \rightarrow e ; V_1 \rightarrow e \text{ et } V_2 \rightarrow e.$$

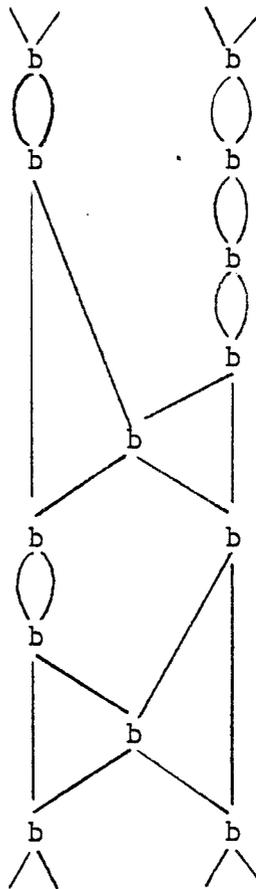
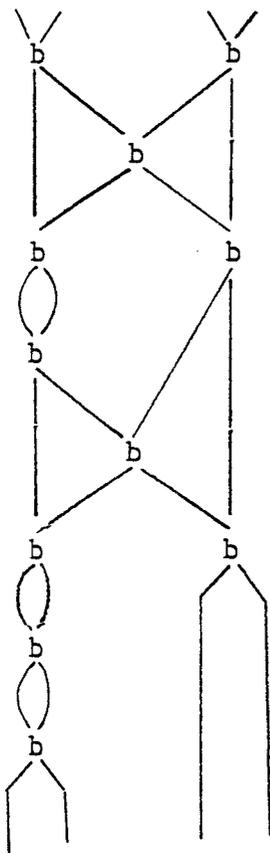
Le taillis généré par cette grammaire G à partir de l'axiome A est une répétition du motif :

suite de une
à n lettres b



suite de une à n' lettres
b

où l'on ne s'impose pas de terminer les dags par un croisement, exemples :



← "croisement"



b) En reprenant les trois points de la construction proposée en III.2. on obtient :

$$M = \langle \{b\}, \{V_0, V_1, V_2, q_f\}, I = s^{-1}(A), F = \{q_f\}^{\otimes}, R \rangle \text{ où}$$

R est l'ensemble de transitions composé

des transitions identifiables aux règles de G (pas de e en partie droite)

$$(V_0 \otimes V_1) \cdot b \vdash b \cdot (V_0 \otimes V_1)$$

$$(V_1 \otimes V_0) \cdot b \vdash b \cdot (V_1 \otimes V_0)$$

$$(V_0 \otimes V_1) \cdot b \vdash b \cdot (V_0 \otimes V_2)$$

$$(V_1 \otimes V_0) \cdot b \vdash b \cdot (V_2 \otimes V_0)$$

$$(V_1 \otimes V_0) \cdot b \vdash b \cdot (V_2 \otimes V_0)$$

$$(V_2 \otimes V_2) \cdot b \vdash b \cdot (V_1 \otimes V_1)$$

des transitions destinée à prendre en compte les effacements de variables

(e en partie droite)

• remplacement des $\begin{array}{c} | \\ e \\ | \end{array}$ par q_f sauf pour $V_1 \rightarrow e$

$$V_0 \otimes V_1 \cdot b \vdash b \cdot (q_f \otimes V_1)$$

$$V_1 \otimes V_0 \cdot b \vdash b \cdot (V_1 \otimes q_f)$$

• prise en compte de la règle $V_1 \rightarrow e$

$$(V_0 \otimes V_1) \cdot b \vdash b \cdot (V_0 \otimes q_f)$$

$$(V_1 \otimes V_0) \cdot b \vdash b \cdot (q_f \otimes V_0)$$

$$(V_2 \otimes V_2) \cdot b \vdash b \cdot (V_1 \otimes q_f)$$

$$(V_2 \otimes V_2) \cdot b \vdash b \cdot (q_f \otimes V_1)$$

$$(V_2 \otimes V_2) \cdot b \vdash b \cdot (q_f \otimes q_f)$$

où $s(V_0) = \{V_0\}$

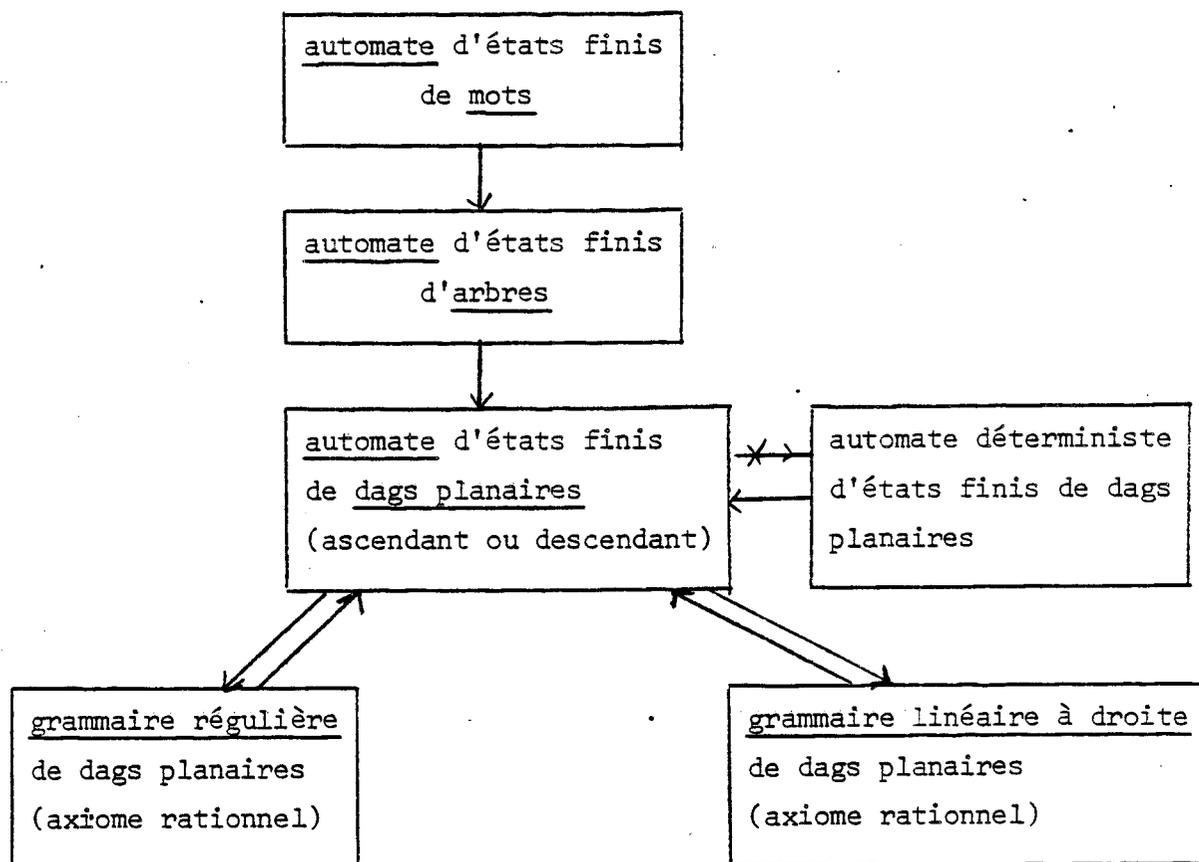
$$s(V_1) = \{V_1, q_f\}$$

$$s(V_2) = \{V_2\} \text{ et } s(q_f) = \{q_f\}$$

et $I = s^{-1}(A) = s^{-1}\{V_0 \otimes V_1 \otimes V_1 \otimes V_0\}$

$$I = \{V_0 \otimes V_1 \otimes V_1 \otimes V_0, V_0 \otimes q_f \otimes V_1 \otimes V_0, V_0 \otimes V_1 \otimes q_f \otimes V_0, V_0 \otimes q_f \otimes q_f \otimes V_0\}$$

3 - Schéma récapitulatif



CHAPITRE IV

LES DAGS DE DERIVATION

PLAN DU CHAPITRE IV

IV.1. - DEFINITIONS

IV.2. - THEOREME : LES TAILLIS DE DAGS DE DERIVATIONS
SONT REGULIERS

IV.2.1. - Exemple introductif

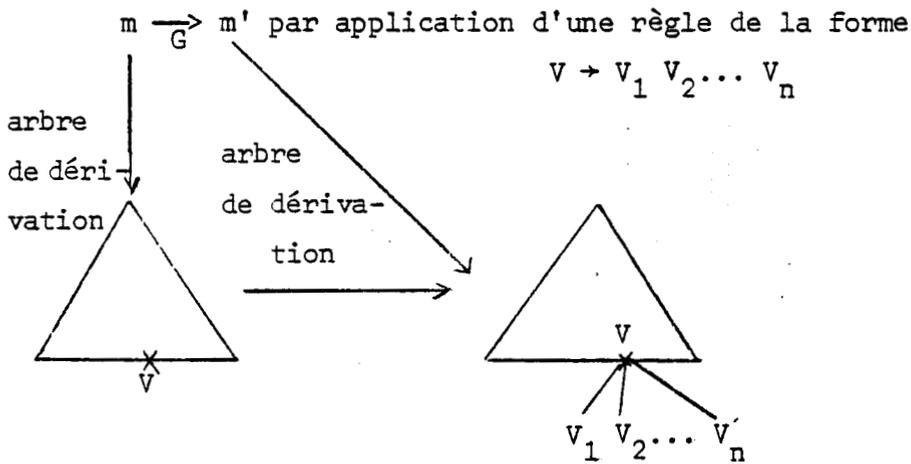
IV.2.2. - Généralisation de la construction

IV.2.3. - Démonstration du théorème

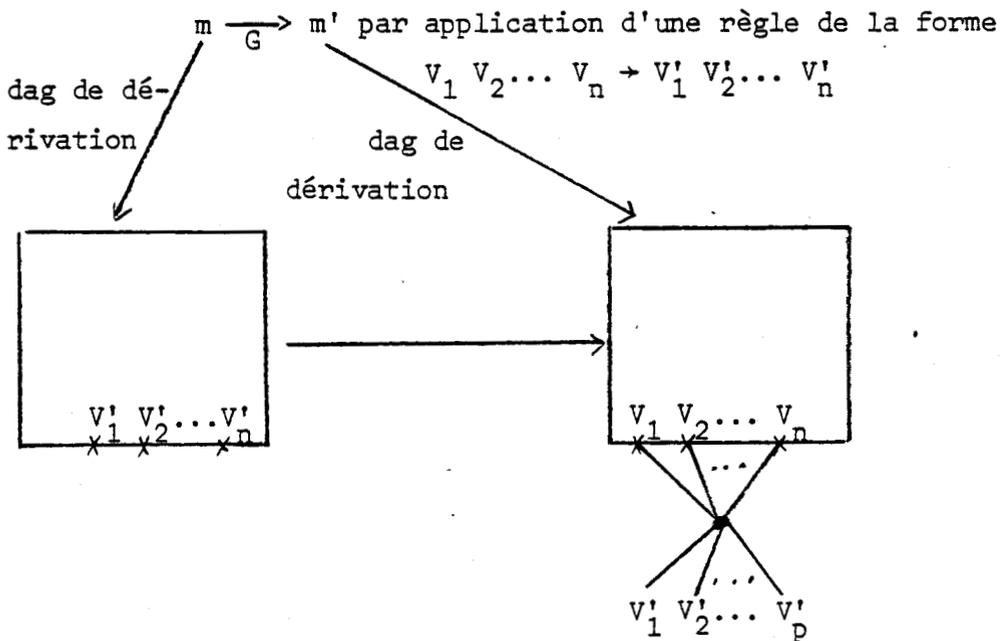
IV.3. - L'INDECIDABILITE DE L'EGALITE DE DEUX TAILLIS
REGULIERS

INTRODUCTION

On a pu observer lors des parties précédentes, la parfaite continuité dans laquelle s'inscrivent les définitions d'automate de dags, de grammaire de dags vis à vis des notions correspondantes qui existent dans les mots et les arbres. On peut alors légitimement envisager de trouver certaines propriétés des dags planaires et plus particulièrement des taillis réguliers, en transposant certaines propriétés observées sur les arbres. On peut notamment étendre la notion d'arbre de dérivation d'une grammaire algébrique, en dag de dérivation d'une grammaire de mots quelconques. En effet, si une réécriture dans une grammaire algébrique peut être assimilée à un greffe dans un arbre :



Une réécriture dans une grammaire quelconque, peut être traduite graphiquement par une connexion dans un dag planaire :



On sait que l'ensemble des arbres de dérivations d'une grammaire algébrique est une forêt régulière d'arbres, et on montrera que l'ensemble des dags de dérivations d'une grammaire à structure de phrases sans ϵ -règles est un taillis régulier. De ce résultat, on pourra déduire que l'égalité de deux taillis réguliers est indécidable. Notons, que sur les dags réguliers, on ne parvient pas à mettre en évidence des lemmes d'itérations aussi généraux et aussi simples que dans les mots, ou les arbres.

IV. - LES DAGS DE DERIVATION D'UNE GRAMMAIRE DE MOTS

IV.1. - DEFINITIONS

IV.1.1. - Définition

On appelle grammaire à structures de phrases sans ϵ -règles la donnée d'un triplet $G = (X, N, P)$ avec :

X ensemble de lettres terminales

N ensemble de lettres non terminales

P ensemble de règles de production, sous-ensemble de $(X \cup N)^* \cdot N \cdot (X \cup N)^* \times (X \cup N)^+$

IV.1.2. - Définition

On appelle le langage engendré par $G = (X, N, P)$, à partir de l'axiome $A \in N$, le sous-ensemble de X^* noté $L(G, A)$ défini de la façon suivante :

$$L(G, A) = \{g \in X^* / A \xrightarrow{*}_G g\}$$

La relation notée $\xrightarrow{*}_G$ est la clôture réflexive et transitive de la relation \xrightarrow{G} définie sur l'ensemble $(X \cup N)^*$ comme suit :

$f \xrightarrow{G} g$ si et seulement si $\exists m_1, m_2 \in (X \cup N)^*$ et $\exists v \vdash m \in P$ tels que $f = m_1 \cdot m \cdot m_2$ et $g = m_1 \cdot v \cdot m_2$.

Remarque :

L'ensemble des langages tels qu'il existe une grammaire à structure de phrases qui les engendre, est l'ensemble des langages récursivement énumérables.

IV.1.3. - Définition

On appelle dérivation dans $G = (X, N, P)$ toute suite :

$$\delta : f_0 \xrightarrow{G} f_1 \xrightarrow{G} \dots \xrightarrow{G} f_i \xrightarrow{G} f_{i+1} \dots \xrightarrow{G} f_n$$

Cette dérivation est de longueur n (n applications de règles de G pour obtenir f_n à partir de f_0).

IV.1.4. - Définition

$\mathcal{D}_G(\delta)$: le dag de dérivation associé à la dérivation δ dans G , est un dag de $\hat{DP}(\Sigma)$ dont le feuillage $\phi(\mathcal{D}_G(\delta))$ est égal au terme de la réécriture δ , et qui retrace la façon dont a été généré ce feuillage, à la manière des arbres des dérivations associés aux grammaires algébriques de mots.

Σ est un alphabet bigradué tel que :

- $\forall x \in X ; x \in \Sigma_1^1$
- $\forall n \in N ; n \in \Sigma_1^1$
- $\forall u \in N$ tel que $\exists v \in P$ alors $u \in \Sigma_{l(v)}^1$
- $\forall u \xrightarrow{G} v \in P$ tel que $l(u) > 1$ alors $c_{l(v)}^{l(u)} \in \Sigma_{l(v)}^{l(u)}$.

Les c_j^i sont de nouveaux symboles appelés connecteurs.

Et $\mathcal{D}_G(\delta)$ est défini de façon récursive par :

- δ est une dérivation de longueur nulle : δ est donc un mot ω de $(X \cup N)^*$ alors :

$\mathcal{D}_G(\delta) = i(\omega)$; avec i l'injection canonique du monoïde $(X \cup N, \cdot)$ sur le monoïde (Σ, \otimes) . On confondra par la suite ω et $i(\omega)$; $\phi(\mathcal{D}_G(\delta))$ est bien égal à ω .

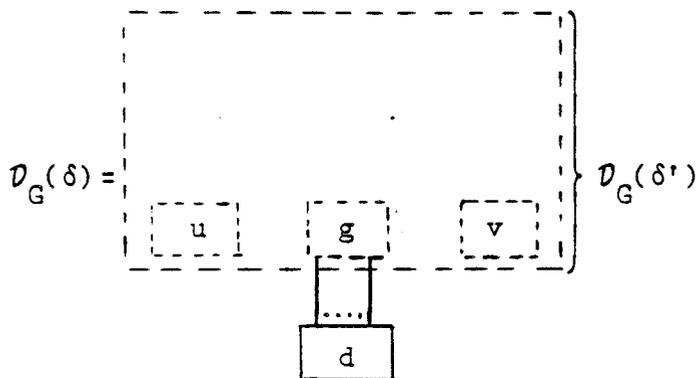
- $\delta : f_0 \xrightarrow{G^*} f_{n-1} \xrightarrow{G} f_n$ alors :

On peut affirmer que :

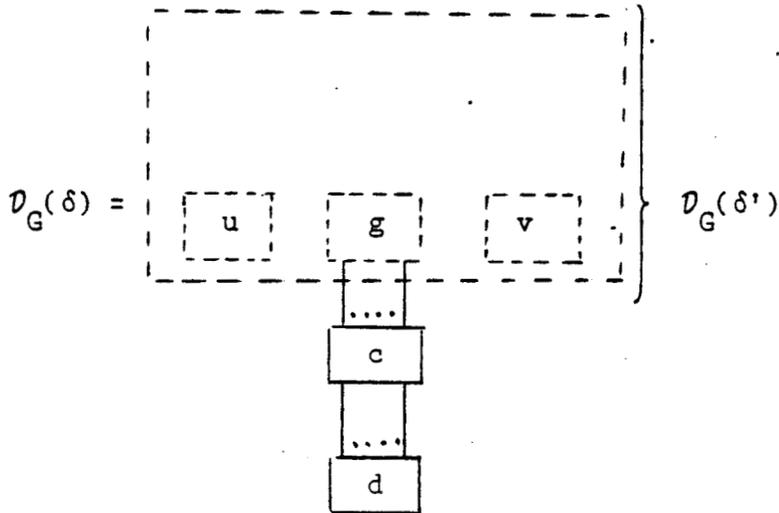
$\exists u, v, g, d \in (X \cup N)^*$ tels $f_{n-1} = ugv$, $f_n = udg$ et $g \xrightarrow{P} d \in P$.

Appelons $\delta' : f_0 \xrightarrow{G^*} f_{n-1}$; alors $\mathcal{D}_G(\delta')$ est un dag de $\hat{DP}(\Sigma)$ dont le feuillage est égal à ugv (par définition) $\mathcal{D}_G(\delta)$ sera :

1^{er} cas : $g \in N$, donc $g \in \Sigma_{\ell(d)}^1$



2^{ème} cas : $l(g) > 1$ alors $c \in \Sigma_{\frac{l(g)}{l(d)}}$



On a bien dans ces deux cas $\phi(\mathcal{D}_G(\delta)) = f_n$.

Remarque :

On a distinguer deux cas :

- Le premier qui rend compte de règles de type algébrique que nous traitons de façon classique. Ainsi, en présence d'une grammaire algébrique, nôtre système élabore des arbres de dérivations.

- Le second qui nous plonge dans une structure de graphe planaire, et qui rend compte de la transformation d'un mot en un autre mot.

IV.1.5. - Définition

On appelle $\mathcal{D}(G, A)$ le taillis des dags de dérivations associés à des réécritures dans G dont le premier terme est A .

$$\mathcal{D}(G, A) = \{ \mathcal{D}_G(\delta) / \delta : A \xrightarrow[G]{*} w, w \in (X \cup N)^* \}$$

On appelle $\mathcal{D}_u(G, A)$ le taillis des dags de dérivations associés à des réécritures terminales dans G dont le premier terme est A .

$$\mathcal{D}_u(G, A) = \{\mathcal{D}_G(\delta) / \delta : A \xrightarrow[G]{*} w, w \in X^*\}.$$

IV.2. - THEOREME : LES TAILLIS DE DAGS DE DERIVATIONS SONT REGULIERS

Enoncé :

$\forall G = (X, N, R)$ grammaire à structures de phrases sans ϵ -règles et $A \in N$ alors $\mathcal{D}(G, A)$ et $\mathcal{D}_u(G, A)$ sont des taillis réguliers de dags.

IV.2.1. - Exemple introductif à la démonstration du théorème

Soit la grammaire à contexte lié suivante $G = (X, N, R)$

$$X = \{a, b, c\}$$

$$N = \{A, C\}$$

$$R = \{A \xrightarrow[1]{\rightarrow} aAbC ; A \xrightarrow[2]{\rightarrow} C ; Cb \xrightarrow[3]{\rightarrow} bC ; CC \xrightarrow[4]{\rightarrow} Cc ; bC \xrightarrow[5]{\rightarrow} bc\}.$$

Regardons d'abord quel est le dag de dérivation associé à la dérivation suivante :

$$\delta : \underline{A} \xrightarrow{(1)} \underline{aAbC} \xrightarrow{(1)} \underline{aaAbCbC} \xrightarrow{(2)} \underline{aaCbCbC} \xrightarrow{(3)} \underline{aabCCbC} \xrightarrow{(3)}$$

$$\underline{aabCbCC} \xrightarrow{(3)} \underline{aabbCCC} \xrightarrow{(4)} \underline{aabbCCc} \xrightarrow{(4)} \underline{aabbCcc} \xrightarrow{(5)} \underline{aabbccc}$$

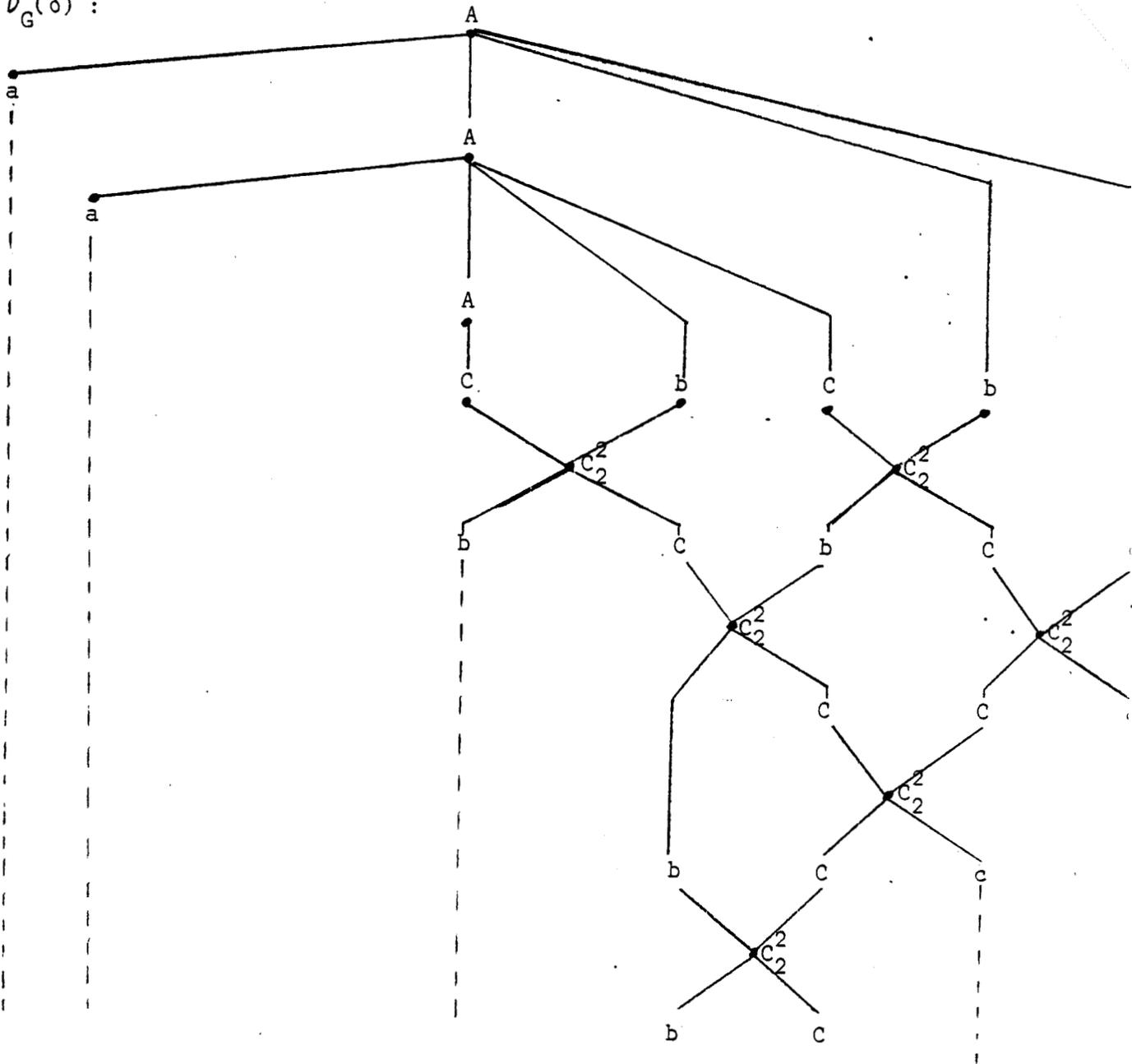
Le dag de dérivation $\mathcal{D}_G(\delta)$ est construit à partir de l'alphabet Σ

$$\Sigma_1^1 = \{a, b, c, A, C\}$$

$$\Sigma_4^1 = \{A\}$$

$$\Sigma_2^2 = \{C_2^2\} ; \text{ le connecteur } 2, 2$$

$\mathcal{D}_G(\delta) :$



Soit la grammaire régulière de DAG suivante $G' = (\Sigma, V, P)$ où Σ est l'alphabet de lettres bigraduées défini précédemment

$\cdot v = \{\overline{A}, \tilde{b}, \tilde{c}\}$



$$\bullet P = \{\bar{A} \xrightarrow{(1)} A \cdot (a \otimes (A/\bar{A}) \otimes (b \cdot \tilde{b}/b) \otimes (C \cdot \tilde{C}/C)) ;$$

$$\bar{A} \xrightarrow{(2)} A \cdot (C \cdot \tilde{C}/C) ;$$

$$\tilde{C} \tilde{b} \xrightarrow{(3)} c_2^2 \cdot (b \cdot \tilde{b}/b \otimes C \cdot \tilde{C}/C) ;$$

$$\tilde{C} \tilde{C} \xrightarrow{(4)} c_2^2 \cdot (C \cdot \tilde{C}/C \otimes c) ;$$

$$\tilde{b} \tilde{C} \xrightarrow{(5)} c_2^2 \cdot (b \cdot \tilde{b}/b \otimes c)$$

Notations :

• La notation "/" dans l'énoncé des règles précédentes symbolise un "ou" exclusif.

Ainsi la notation $\bar{A} \rightarrow A \cdot (C \cdot \tilde{C}/C)$ représente les deux règles

$$- \bar{A} \rightarrow A \cdot C \cdot \tilde{C}$$

$$- \bar{A} \rightarrow A \cdot C$$

• Le symbole λ désigne l'élément neutre du monoïde $(X \cup N)^*$

Remarques sur la forme des règles de G' :

• Chaque variable de V correspond à une lettre de $X \cup N$:

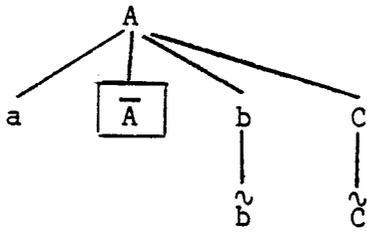
$$\bar{v} \in V \iff \exists v \mid \rightarrow m \in R$$

$$\tilde{v} \in V \iff \exists u', u'', w \in (X \cup N)^* \text{ tel que } u' \cdot u'' \neq \lambda \text{ et } u' \cdot v \cdot u'' \mid \rightarrow w \in R.$$

• On peut constater que chacune des règles de P est directement associé à une règle de R.

Construisons la dérivation dans G' dont le dernier terme est $\mathcal{D}_G(\delta)$:

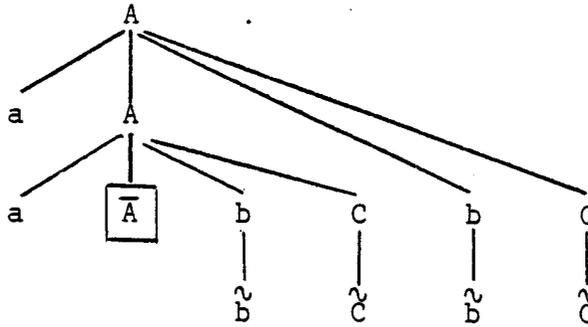
$\bar{A} \xrightarrow{G'} \rightarrow$



par application de

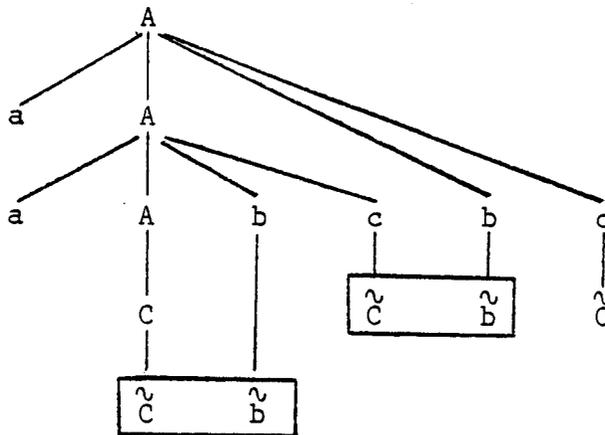
$$\bar{A} \xrightarrow{G'} \rightarrow A \cdot (a \otimes \bar{A} \otimes b \cdot \tilde{b} \otimes C \cdot \tilde{C})$$

$\xrightarrow{G'}$



idem

$\xrightarrow{G'}$



par application de
 $\bar{A} \xrightarrow{G'} \rightarrow A \cdot (C \cdot \tilde{C})$

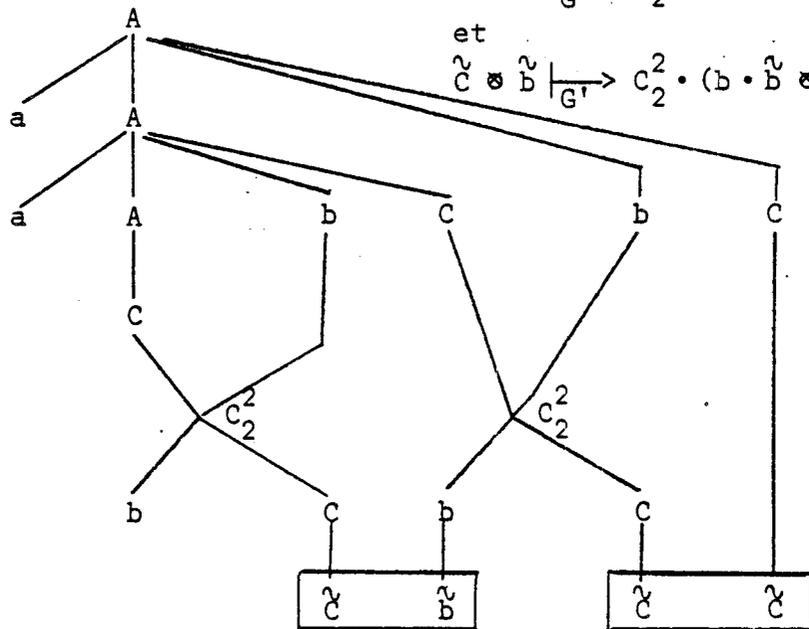
$\xrightarrow{G'}$

par application de

$$\tilde{c} \otimes \tilde{b} \xrightarrow{G'} c_2^2 \cdot (b \otimes c \cdot \tilde{c})$$

et

$$\tilde{c} \otimes \tilde{b} \xrightarrow{G'} c_2^2 \cdot (b \cdot \tilde{b} \otimes c \cdot \tilde{c})$$

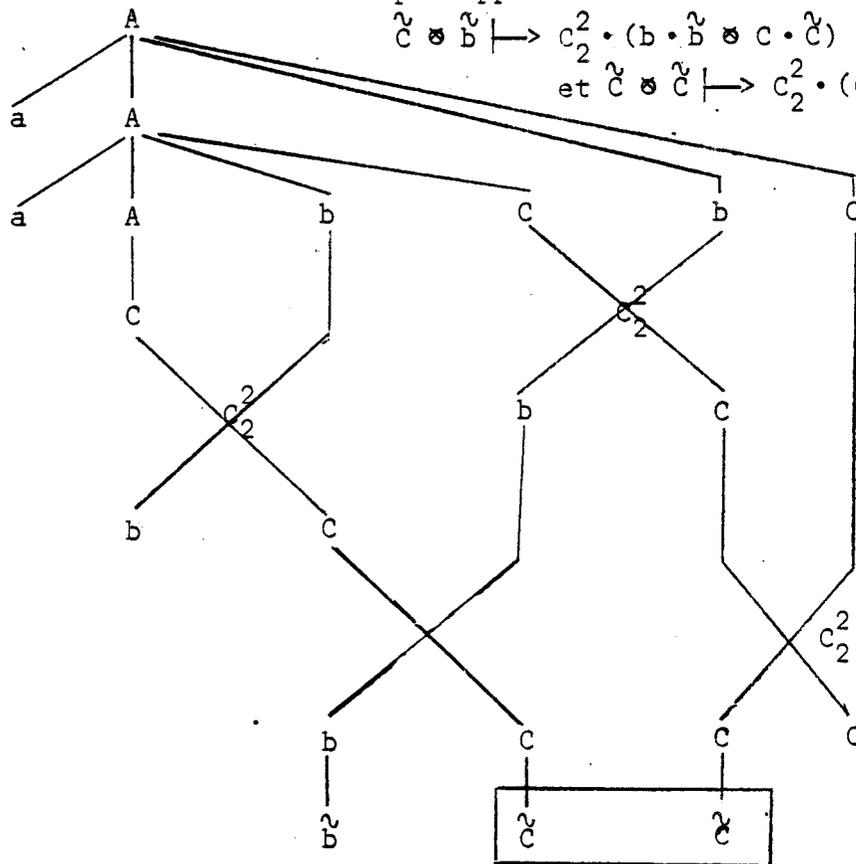


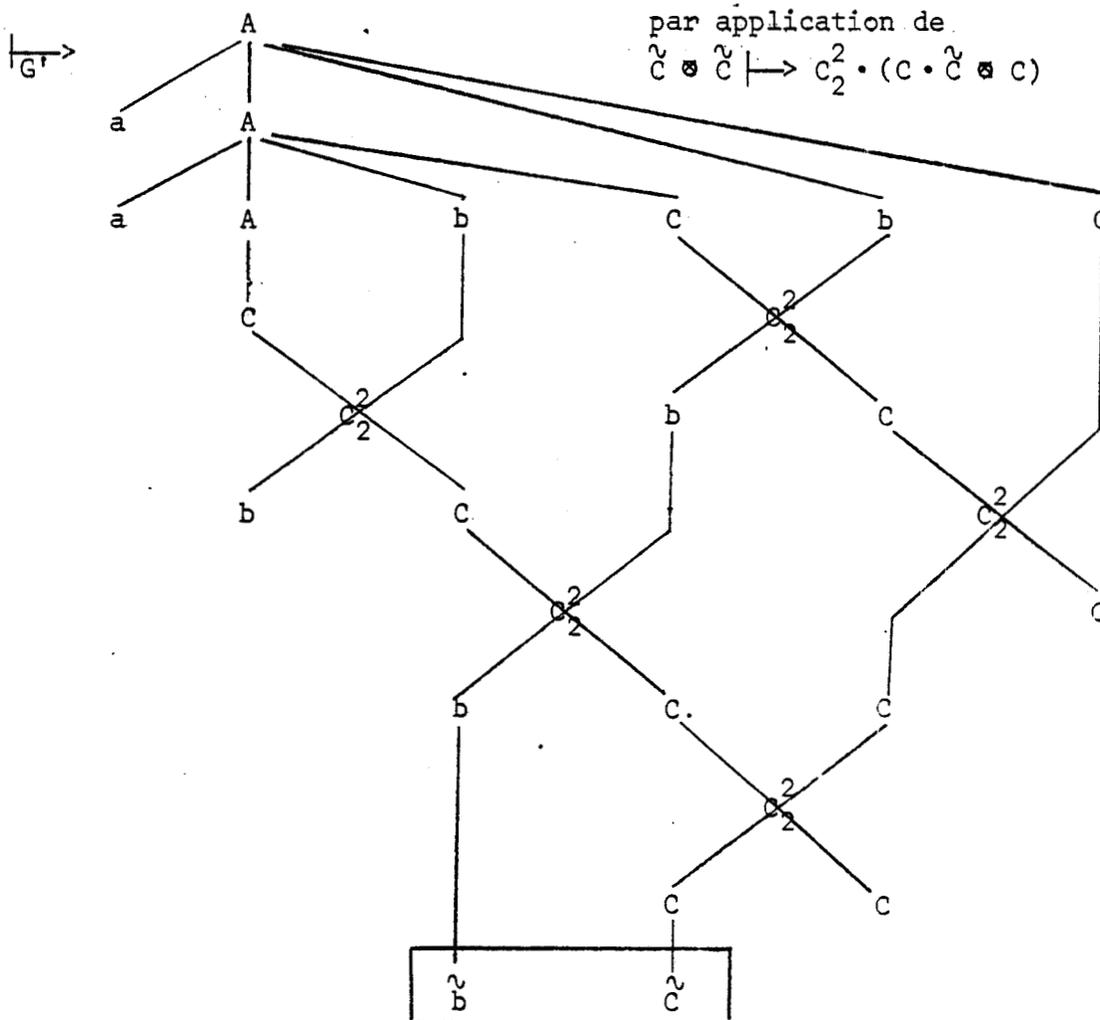
$\xrightarrow{G'}$

par application de

$$\tilde{c} \otimes \tilde{b} \xrightarrow{G'} c_2^2 \cdot (b \cdot \tilde{b} \otimes c \cdot \tilde{c})$$

$$\text{et } \tilde{c} \otimes \tilde{c} \xrightarrow{G'} c_2^2 \cdot (c \cdot \tilde{c} \otimes c)$$





et qui par application de la règle $\tilde{b} \otimes \tilde{c} \mapsto c_2^2 \cdot (b \otimes c)$, donne $\mathcal{D}_G(\delta)$.

IV.2.2. - Généralisation : Construction d'une grammaire régulière de dags G' qui engendre le taillis-des dags de dérivation d'une grammaire à structure de phrases

Soit $G = (X, N, R)$ grammaire de mots à structure de phrases sans ϵ -règles, où

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

$$N = \{N_1, N_2, \dots, N_r\}, A \in N$$

$$R = \{u_i \rightarrow v_i ; u_i \in (X \cup N)^* N (X \cup N)^* \text{ et } v_i \in (X \cup N)^+\}_{i \in [r]}$$

Nous allons construire une grammaire régulière de DAG, $G' = (\Sigma, V, P)$ telle que $T(G', \bar{A}) = \mathcal{D}(G, A)$.

- Construction de Σ : voir définition IV.4.

- Construction de V :

$\bar{N}_i \in V \iff N_i$ est la partie gauche d'une règle de R .

\tilde{X}_i (resp. \tilde{N}_i) $\in V \iff X_i$ (resp. N_i) apparaît dans une partie gauche de règle de R .

- Construction de P :

A chaque règle de R , on va associer une famille de règles de P .

Préliminaire 1

On appellera $E = X \cup N \cup V \cup \{x \cdot \tilde{x} / \tilde{x} \in V\}$ et

$E^* = (E, \otimes, e_0)$ le monoïde libre construit à partir de l'alphabet E et de l'opération \otimes .

Préliminaire 2

Soit s la substitution rationnelle de $(X \cup N)^*$ dans E^* telle que :

$s(x) = \{x, x \cdot \tilde{x}\}$ ssi $x \in X$ et x apparaît en partie gauche d'une règle de P .

$s(x) = \{x\}$ ssi $x \in X$ et x n'apparaît pas en partie gauche d'une règle de P .

$s(n) = \{n, \bar{n}\}$ ssi $n \in N$ et n apparaît uniquement comme membre gauche de règle de P .

$s(n) = \{n, n \cdot \tilde{n}\}$ ssi $n \in N$ et n apparaît uniquement en partie gauche de règle de P .

$s(n) = \{n, \bar{n}, n \cdot \tilde{n}\}$ ssi $n \in N$ et n apparaît comme membre gauche de règle de P et en partie gauche de règle de P .

- Premier type de règle de P

- Pour une règle de R de la forme

$$u_1 u_2 \dots u_k \xrightarrow{G} w ; k > 1 ; u_j \in X \cup V$$

on crée les règles de P suivantes :

$$\tilde{u}_{i_1} \otimes \tilde{u}_{i_2} \otimes \dots \otimes \tilde{u}_{i_k} \xrightarrow{G'} C_{\ell(w)}^k \cdot s(w).$$

Exemple :

A la règle $Cb \rightarrow bC$, de l'exemple introductif correspondent les règles de P :

$$\tilde{C} \otimes \tilde{b} \xrightarrow{} C_2^2 \cdot s(b \otimes C)$$

$$\text{Soit } \tilde{C} \otimes \tilde{b} \xrightarrow{} C_2^2 \cdot (b/b \cdot \tilde{b} \otimes C/C \cdot \tilde{C})$$

- Second type de règle de P

- Pour une règle de R de la forme :

$$N_i \xrightarrow{} w ; N_i \in N$$

on crée les règles de P suivantes :

$$\bar{N}_i \xrightarrow{} N_i \cdot s(w)$$

Exemple :

A la règle $A \rightarrow C$ de R de l'exemple introductif correspondent les règles de P :

$$\bar{A} \rightarrow A \cdot (C/C \cdot \tilde{C})$$

- On rajoute la règle $\bar{A} \rightarrow A$, si A est l'axiome considéré.

IV.2.3. - Démonstration du théorème

Il suffit de démontrer que $\mathcal{D}(G, A) = T(G', \bar{A})$.

- Preuve de $\mathcal{D}(G, A) \subseteq T(G', \bar{A})$:

- $\mathcal{D}(G, A) = \{\mathcal{D}_G(\delta) \text{ telle que } \delta \text{ est une dérivation dans } G \text{ dont le premier terme est } A\}$.

Nous allons démontrer par récurrence sur la longueur des dérivations δ débutant par A , que $\mathcal{D}_G(\delta) \in T(G', \bar{A})$.

Cas d'une dérivation de longueur nulle :

$l(\delta) = 0 \Rightarrow \mathcal{D}_G(\delta) = A$; dag obtenu par l'application de la règle $\bar{A} \xrightarrow{G} A$.

Hypothèse de Récurrence :

$\forall \delta ; l(\delta) \leq n$ alors $\mathcal{D}_G(\delta) \in T(G', \bar{A})$.

Soit δ' telle que $l(\delta') = n+1$.

δ' est donc de la forme $A \xrightarrow{G}^n ugv \xrightarrow{G} udv$ avec $g \rightarrow d \in R$.

Appelons δ la dérivation de longueur n : $A \xrightarrow{G}^n ugv$; d'après notre hypothèse de récurrence, et les propriétés des dags de dérivations, on a :

$\mathcal{D}_G(\delta) \in T(G', \bar{A})$, ou $\bar{A} \xrightarrow{G'}^* \mathcal{D}_A(\delta)$

et

$\phi(\mathcal{D}_G(\delta)) = ugv$.

D'après la construction de la grammaire G' , on est assuré de $1'$ dans P de la règle suivante :

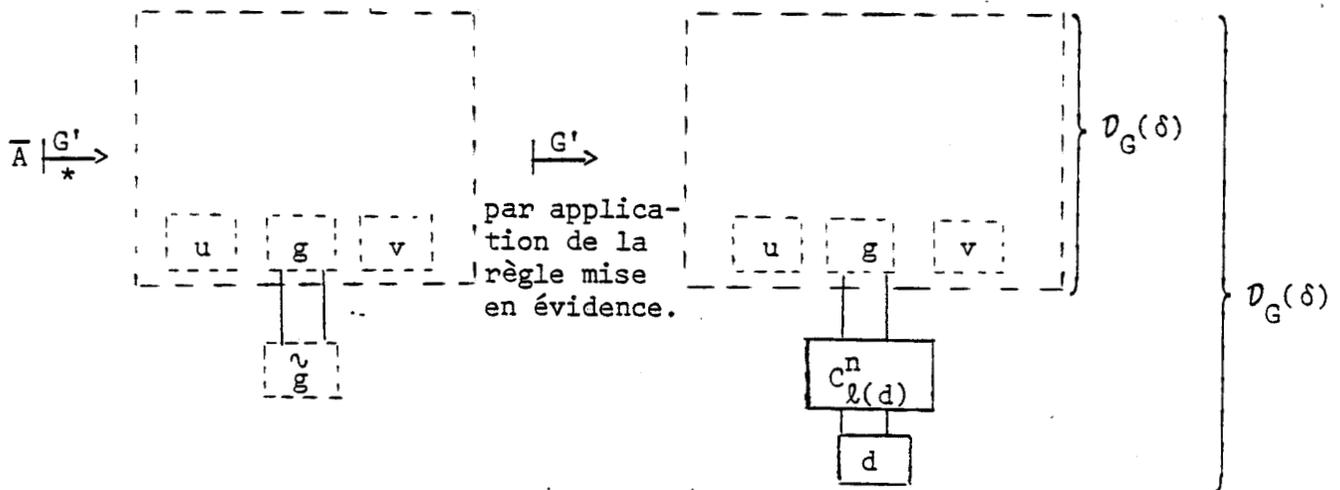
cas 1 - si $\ell(g) > 1$ $\tilde{g}_1 \otimes \tilde{g}_2 \otimes \dots \otimes \tilde{g}_n \xrightarrow{G'} C_{\ell(d)}^n \cdot d$

cas 2 - si $\ell(g) = 1$ $\bar{g} \xrightarrow{G'} g \cdot d$

Considérons dans la dérivation $\bar{A} \xrightarrow{G'}^* \mathcal{D}_A(\delta)$, toutes les applications de règles qui font apparaître une lettre de g . On peut leur substituer des applications de règles ayant naturellement la même partie gauche, mais dont la partie droite ferait apparaître les lettres de g suivi de la variable qui leur correspond \tilde{g}_i dans le cas 1 ou dans laquelle \bar{g} remplacerait g dans le cas 2.

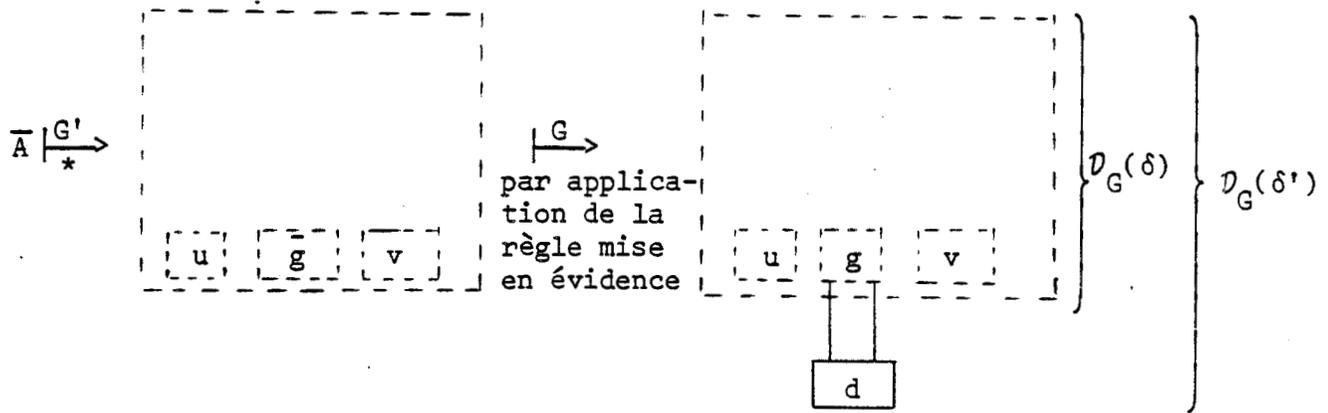
On aurait ainsi :

cas 1 :



$\Rightarrow \mathcal{D}_G(\delta') \in T(G', \bar{A})$.

cas 2 :



$\Rightarrow \mathcal{D}_G(\delta') \in T(G', \bar{A})$.

□

- Preuve que $T(G', \bar{A}) \subset \mathcal{D}(G, A)$:

Nous allons pour ce faire démontrer que :

avec h l'homomorphisme de $(\Sigma \cup V)^*$ dans $(\{x \in \Sigma ; x \in X\} \cup \{e_1^1\})^*$ tel que

$$\begin{aligned} \forall X \in \Sigma \quad h(X) &= X \\ \forall \tilde{x} \in V \quad h(\tilde{x}) &= e_1^1 \\ \forall \bar{N} \in V \quad h(\bar{N}) &= N \end{aligned}$$

On a la propriété suivante :

$$\forall d \in \hat{D}P(\Sigma \cup V) \text{ tel que } \bar{A} \xrightarrow[G']{*} d$$

si $d = m \cdot \phi(d)$ alors $m \cdot h(\phi(d)) \in \mathcal{D}(G, A)$.

Nous allons démontrer cette propriété par récurrence sur la longueur des dérivations dans G' .

Cas d'une dérivation de longueur 1 à partir de \bar{A} : il ne peut s'agir que de la règle $\bar{A} \xrightarrow[G']{*} A ; A \in \mathcal{D}(G, A)$. □

Ou d'une règle de la forme $\bar{A} \xrightarrow[G']{*} A \cdot s(w)$ avec $A \rightarrow w \in R$.

$$A \cdot h(s(w)) = A \cdot w ; A \cdot w \in \mathcal{D}(G, A). \quad \square$$

Hypothèse de récurrence :

$$\forall d \in \overset{\vee}{\mathcal{D}}\mathcal{P}(\Sigma \cup V) \forall \ell \leftarrow n \text{ tel que } \bar{A} \xrightarrow[\mathcal{G}]{\ell} d$$

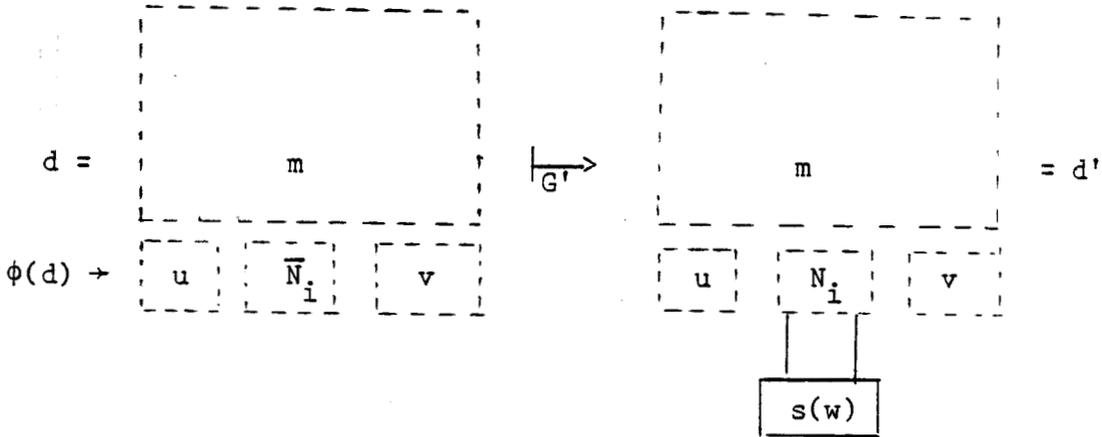
si $d = m \cdot \phi(d)$ alors $m \cdot h(\phi(d)) \in \mathcal{D}(G, A)$.

Soit $d \in \overset{\vee}{\mathcal{D}}\mathcal{P}(\Sigma \cup V)$ tel que $\bar{A} \xrightarrow[\mathcal{G}]{\ell} d$.

Envisageons les différents types de règles applicables à partir de d .

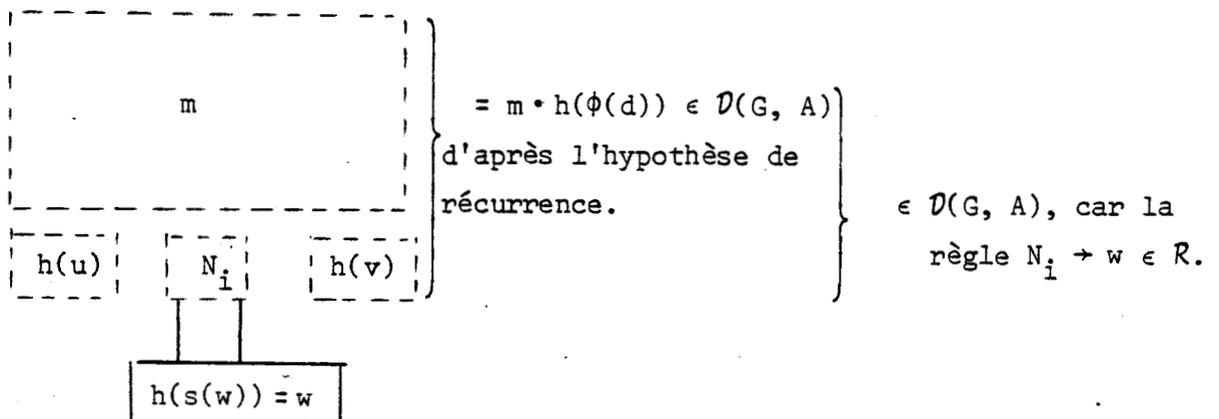
cas 1 :

$$\bar{N}_i \xrightarrow[\mathcal{G}]{\ell} N_i \cdot s(w) \quad (\text{i.e. } N_i \xrightarrow[\mathcal{G}]{\ell} w \in R).$$



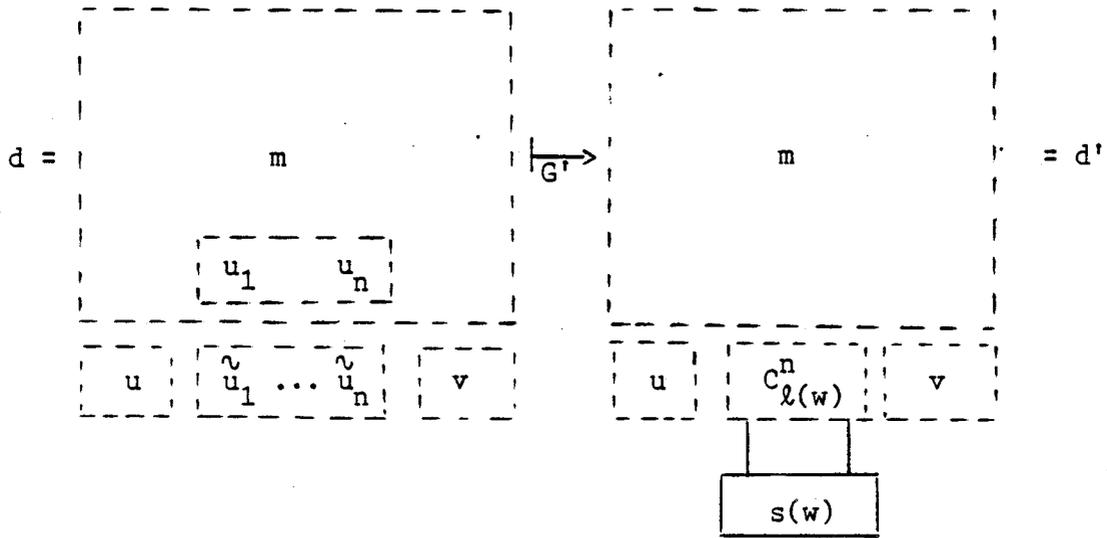
Regardons quel est le dag obtenu en appliquant au feuillage de d' , l'homomorphisme h .

Ce dag est le suivant :



cas 2 :

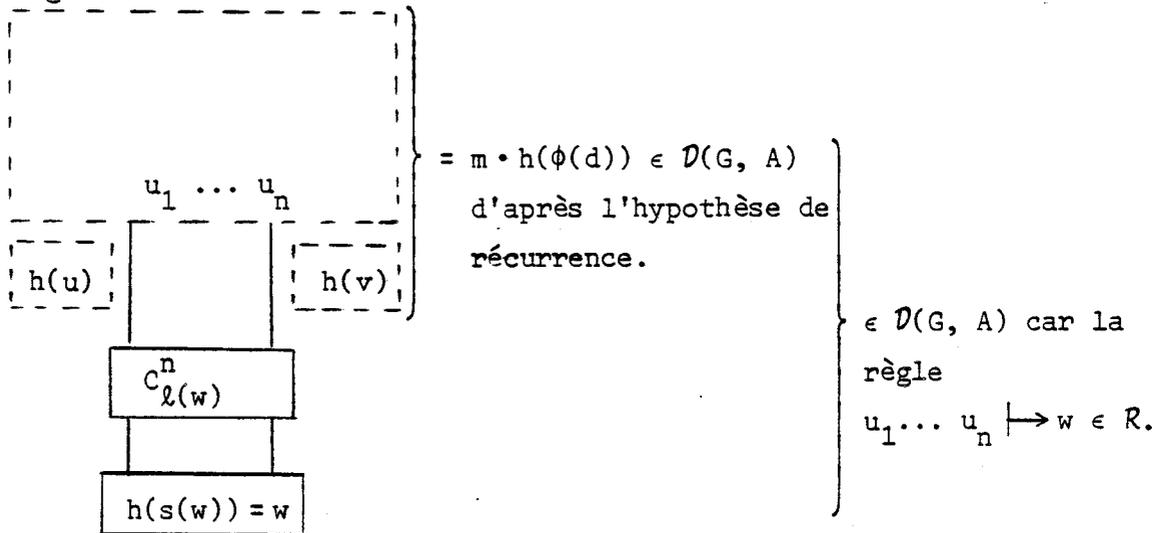
$$\tilde{u}_1 \otimes \tilde{u}_2 \otimes \dots \otimes \tilde{u}_n \xrightarrow{G'} C_{\ell(w)}^n \cdot s(w) \quad (\text{et } u_1 \dots u_n \vdash w \in R).$$



La présence de $\{u_1 \dots u_n\}$ "au dessous" de $\{\tilde{u}_1 \dots \tilde{u}_n\}$ résulte de la forme des règles qui génère uniquement des $u_i \cdot \tilde{u}_i$.

Regardons quel est le dag obtenu en appliquant au feuillage de d' l'homomorphisme h .

Ce dag est le suivant :



Cette propriété nous permet de conclure que $T(G', \bar{A}) \subseteq \mathcal{D}(G, A)$, car tout dag de $T(G', \bar{A})$ est conservé par application à son feuillage de l'application h .

Ceci conclut la première partie du théorème : $\mathcal{D}(G, A)$ est un taillis régulier. \square

La démonstration de la seconde partie du théorème : $\mathcal{D}_u(G, A)$ est un taillis régulier, repose sur la construction d'une grammaire régulière très proche de la précédente. La restriction que l'on s'impose, est que chaque lettre de Σ correspondant à une lettre quelconque de N devra être suivi dans les membres droits des règles de P , de la variable de V qui lui est associée (son apparition ne donnera donc jamais de dag terminal).

On peut alors reprendre la même construction, en modifiant la définition de s de la manière suivante :

- $x \in X$ et x apparaît en partie gauche de règle de $R \Rightarrow s(x) = \{x, x \cdot \tilde{x}\}$
- $x \in X$ et x n'apparaît pas en partie gauche de règle de $R \Rightarrow s(x) = \{x\}$
- $n \in N$ et n apparaît uniquement comme membre gauche d'une règle de $R \Rightarrow s(n) = \{\bar{n}\}$
- $n \in N$ et n apparaît uniquement en partie gauche de règle de $R \Rightarrow s(n) = \{n \cdot \tilde{n}\}$
- $n \in N$ et n apparaît en partie gauche de règle de R , et comme membre gauche de règle de $R \Rightarrow s(n) = \{n \cdot \tilde{n}, \bar{n}\}$

et en supprimant de P la règle $\bar{A} \rightarrow A$.

Exemple :

La règle $A \xrightarrow{G} aABC$ de R , n'exige plus que l'existence des règles.

$$\bar{A} \xrightarrow{G'} A \cdot (a \otimes \bar{A} \otimes b \otimes C \cdot \tilde{C})$$

$$\bar{A} \xrightarrow{G'} A \cdot (a \otimes \bar{A} \otimes b \cdot \tilde{b} \otimes C \cdot \tilde{C}) \text{ dans } P.$$

Remarque :

Ce théorème prolonge un autre résultat bien connu qui nous dit que les arbres de dérivations d'une grammaire algébrique constitue une forêt régulière d'arbres. Pour y parvenir on a dû définir une notion de régularité dans une structure plus riche que les arbres : les dags planaires. Comme dans le cas des arbres où un arbre de dérivation caractérise une classe de dérivations "équivalentes", un dag de dérivation caractérise une classe de dérivation "équivalentes" pour une grammaire à structures de phrases.

IV.3. - L'INDECIDABILITE DE L'EGALITE DE DEUX TAILLIS REGULIERS

On obtient comme corollaire du théorème précédent que l'égalité de deux taillis réguliers est indécidable par le raisonnement suivant :

Si l'égalité de deux taillis réguliers était décidable ; on pourrait décider l'égalité au vide d'un taillis régulier.

Dans cette hypothèse, pour $G = (X, N, R)$ grammaire à structure de phrases sans ϵ -règles et $A \in N$, l'égalité $\mathcal{D}_u(G, A) = \emptyset$ serait décidable.

Or $\mathcal{D}_u(G, A) = \emptyset \iff L(G, A) = \emptyset$, l'égalité $L(G, A) = \emptyset$ deviendrait également décidable, ce qui s'oppose au résultat établi que l'égalité au vide du langage engendré par une grammaire à structure de phrases sans ϵ -règles est indécidable.

La difficulté de résoudre les problèmes d'égalité de taillis réguliers peut être illustrée par l'exemple suivant :

Exemple :

Soit $M_1 = (\Sigma, Q_1, I_1, F_1, R_1)$ et $M_2 = (\Sigma, Q_2, I_2, F_2, R_2)$ automates de dags planaires ; avec

$$\Sigma = \Sigma_2^2 = \{x\}$$

$$Q_1 = \{g, d, q\} ; Q_2 = \{q_1, q_2\}$$

$$I_1 = q \otimes d \otimes q + q \otimes g \otimes q ; I_2 = q_1 \otimes q_1 \otimes q_2 + q_1 \otimes q_2 \otimes q_2$$

$$F_2 = q_1^* ; F_2 = q_2^*$$

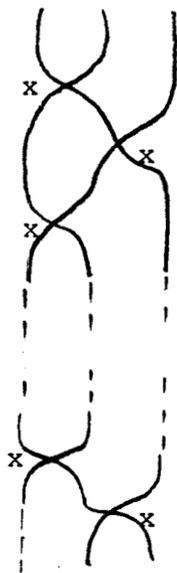
$$R_1 = \{(g \otimes q) \cdot x \mid x \cdot (d \otimes q) ; (q \otimes d) \cdot x \mid x \cdot (q \otimes g)\}$$

$$R_2 = \{(q_1 \otimes q_2) \cdot x \mid x \cdot (q_1 \otimes q_1) ; (q_1 \otimes q_2) \cdot x \mid x \cdot (q_2 \otimes q_2)\}$$

Tout éloigne, ces deux taillis reconnaissables dans leur mode de reconnaissance :

- nombre d'états des automates
- un automate déterministe en montant et l'autre en descendant
- forme des règles.

Or manifestement $T(M_1) = T(M_2) =$ ensembles des "tresses à trois brins" de longueurs quelconques, c'est-à-dire les dags qui ont l'allure suivante :



CHAPITRE V



CONNEXITE DANS LES DAGS PLANAIRE

PLAN DU CHAPITRE V

V.1. - L'AUTOMATE DE DAGS PLANAIRES DG

V.2. - PROPRIETE DE L'AUTOMATE DG EN RAPPORT AVEC LA CONNEXITE

V.3. - CONCLUSION

V. - CONNEXITE DANS LES DAGS PLANAIRES

La notion de connexité tient dans l'étude des dags, une place de premier plan. Ainsi la clôture par homomorphisme n'est obtenue que pour les homomorphismes connexes. Nous verrons dans ce chapitre qu'un taillis reconnaissable de dags connexes peut être reconnu par un automate dont les contrôles haut et bas se réduisent à des états initiaux et finaux. Nous observerons également, lors de la démonstration de l'inclusion de RAT dans REC, le rôle primordial que jouent les parties connexes des dags envisagés. La notion de connexité considérée ici répond à la définition classique de connexité sur les graphes : un graphe connexe est tel que pour toute paire de sommets distincts, ceux-ci sont reliés par une suite d'arcs.

Nous présenterons dans ce chapitre un outil particulièrement utile dans l'analyse de la structure d'un dag : l'automate de dags que nous appellerons DG.

V.1. - L'AUTOMATE DE DAG PLANAIRE DG

$DG = \langle \Sigma, H, I, F, P \rangle$ où

- Σ alphabet bigradué sans lettre de arité ou co-arité nulle

- $H = \{g, c, d, 0\}$

- $I = H^*$

- $F = 0^*$

- $\forall a \in \Sigma$ i) $(g \otimes c^*) \cdot a \vdash a \cdot (g \otimes c^* \otimes d + 0)^* \otimes g \otimes c^* \in P$

- ii) $c^* \cdot a \vdash a \cdot (c^* + c^* \otimes d \otimes (g \otimes c^* \otimes d + 0)^* \otimes g \otimes c^*) \in P$

$$\text{iii) } (c^* \otimes d) \cdot a \vdash a \cdot (c^* \otimes d \otimes (g \otimes c^* \otimes d + 0)^*) \in \mathcal{P}$$

$$\text{iv) } 0 \cdot a \vdash a \cdot (g \otimes c^* \otimes d + 0)^* \in \mathcal{P}$$

$$\text{v) } (g \otimes c^* \otimes d) \cdot a \vdash a \cdot (g \otimes c^* \otimes d + 0)^* \in \mathcal{P}$$

Note :

* désigne ici l'itération de l'opération \otimes dans le monoïde libre construit à partir de H et de l'opération \otimes .

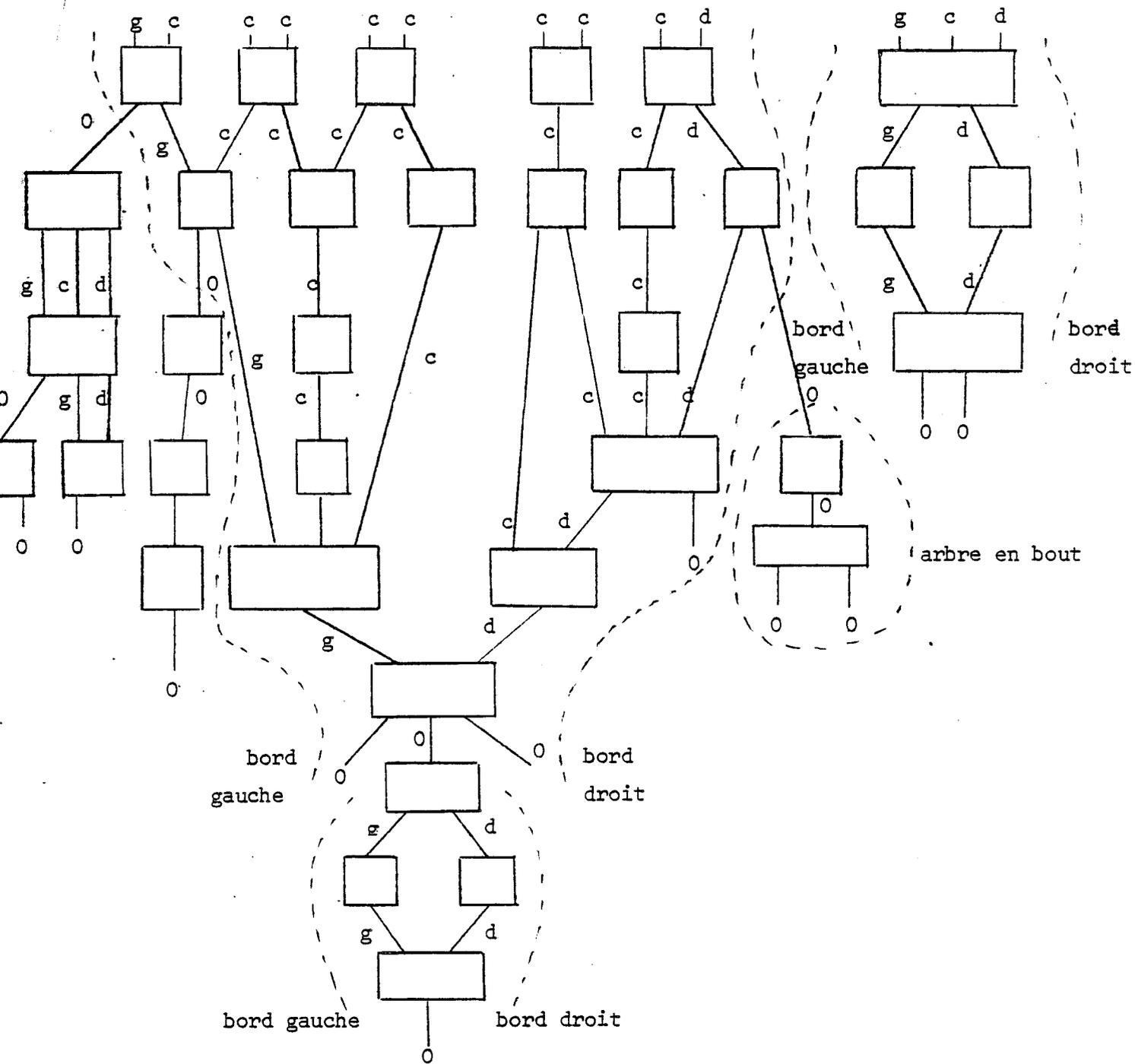
Remarque :

Cet automate est un automate ascendant déterministe, d'où $\forall \delta \in T(DG)$
 $\exists ! w \in H^*$ tel que $w \cdot \delta \stackrel{DG}{\vdash} \delta \cdot w'$; $w' \in 0^*$

Les propriétés qui suivent, explicitent formellement le comportement de cet automate sur les dags, et témoignent de son intérêt quant à l'étude de la structure des dags et en particulier de ses composantes connexes.

V.2. - PROPRIETES DE L'AUTOMATE DG EN RAPPORT AVEC LA CONNEXITE

Cet automate effectue au fil de la reconnaissance d'un dag, un "marquage" des bords droits et gauches des sous-dags connexes (états d et g), et celui des sous-arbres extrêmes (état 0) comme nous le constatons sur l'exemple suivant :



V.2.1. - Propriété

$\forall \delta \in \overset{\vee}{DP}(\Sigma) \quad \exists w \in (0 + (g \otimes c^* \otimes d))^*$ et $w' \in 0^*$ tel que

$$w \cdot \delta \stackrel{*}{\underset{DG}{|}} \delta \cdot w'.$$

Preuve :

Par récurrence sur le nombre de lettre du dag δ .

- δ est composé d'une seule lettre, alors il est reconnu par une règle de la forme (v) dans les conditions prescrites.

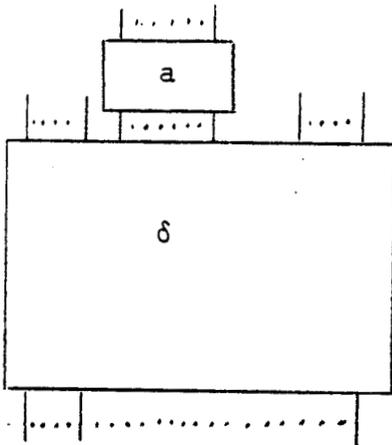
Hypothèse de récurrence :

$\forall \delta \in \hat{D}\hat{P}(\Sigma)$ tel que δ est composé de moins de n lettres alors

$\exists w \in (O + (g \otimes c^* \otimes d))^*$ et $w' \in O^*$ tel que

$$w \cdot \delta \stackrel{*}{\underset{DG}{\mid}} \delta \cdot w'$$

soit $\delta' =$



où $a \in \Sigma$ et δ dag de n lettres

il existe alors d'après l'hypothèse de récurrence $w \in (O + (g \otimes c^* \otimes d))^*$ et $w' \in O^*$ tel que : $w \cdot \delta \stackrel{*}{\underset{DG}{\mid}} \delta \cdot w'$.

w peut se décomposer en $V_1 \otimes V_2 \otimes V_3$ de manière à ce que $(V_1 \otimes (a \cdot V_2) \otimes V_3) \cdot \delta \stackrel{*}{\underset{DG}{\mid}} \delta' \cdot w'$.

V_2 est un facteur d'un mot de $(O + g \otimes c^* \otimes d)^* \Rightarrow$

V_2 appartient à l'un des langages suivants :

$$\begin{array}{l}
 L_1 = (g \otimes c^* \otimes d + 0)^* \otimes g \otimes c^* \\
 L_2 = c^* \otimes d \otimes (g \otimes c^* \otimes d + 0)^* \\
 L_3 = c^* \otimes d \otimes (g \otimes c^* \otimes d + 0)^* \otimes g \otimes c^* \\
 L_4 = c^* \\
 L_5 = (g \otimes c^* \otimes d + 0)^*
 \end{array}
 \left. \vphantom{\begin{array}{l} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \end{array}} \right\} \Rightarrow$$

$a \cdot V_2$ est la partie gauche d'une règle de P

Soit $u_2 \cdot a \vdash a \cdot V_2$ cette règle

$$(V_1 \otimes u_2 \otimes V_3) \cdot \delta' \stackrel{*}{\vdash}_{DG} \delta' \cdot w'$$

de plus, il est évident au vu de la forme des règles de P que le résultat de la substitution de v_2 par u_2 dans w est lui-même un mot de $(0 + g \otimes c^* \otimes d)^*$

Comme corollaire de cette propriété on a : $T(DG) = \hat{DP}(\Sigma)$.

Tout dag de $\hat{DP}(\Sigma)$ est donc reconnu par l'automate DG ; nous allons examiner comment la reconnaissance par l'automate DG peut nous donner des informations sur la structure du dag. Nous constaterons dans un premier cas, que les états de cet automate sont porteurs d'une interprétation graphique.

V.2.2. - Propriété

$$\forall \delta \in \hat{DP}(\Sigma).$$

$$1) \exists w \in H^* \otimes d \otimes H^* \quad \exists w' \in O^* \text{ tel que } w \cdot \delta \stackrel{*}{\vdash}_{DG} \delta \cdot w' \Rightarrow$$

$$\delta = \delta_1 \otimes \delta_2$$

$$w = w_1 \otimes d \otimes w_2$$

$$w' = w'_1 \otimes w'_2$$

$$\text{tels que } (w_1 \otimes d) \cdot \delta_1 \stackrel{*}{\vdash}_{DG} \delta_1 \cdot w'_1.$$

Cela signifie que l'état d "marque" l'arête droite d'un dag.

$$2) \exists w \in H^* \otimes g \otimes H^* \quad \exists w' \in O^* \text{ tels que } w \cdot \delta \Big|_{DG}^* \delta \cdot w' \Rightarrow$$

$$\delta = \delta_1 \otimes \delta_2$$

$$w = w_1 \otimes g \otimes w_2$$

$$w' = w'_1 \otimes w'_2$$

$$\text{tels que } (g \otimes w_2) \cdot \delta_2 \Big|_{DG}^* \delta_2 \cdot w'_2$$

Cela signifie que l'état g "marque" l'arête gauche d'un dag.

$$3) \exists w \in H^* \otimes O \otimes H^* \quad \exists w' \in O^* \text{ tels que } w \cdot \delta \Big|_{DG}^* \delta \cdot w' \Rightarrow$$

$$\delta = \delta_1 \otimes \delta_2 \otimes \delta_3$$

$$w = w_1 \otimes O \otimes w_3$$

$$w' = w'_1 \otimes w'_2 \otimes w'_3$$

$$\text{tels que } O \cdot \delta_2 \Big|_{DG}^* \delta_2 \cdot w'_2.$$

Cela signifie que l'état O marque des arcs qui sont à la fois arête gauche et arête droite d'un dag.

Preuve :

L'analyse de la forme des règles nous fait observer que :

dans les parties gauches de règles, l'état d est toujours au-dessus de l'extrémité droite d'une lettre (confère les règles (iii) et v)) ; et que par application de ces règles (iii) et (v), on retrouve sous l'extrémité droite de la lettre un d ou un O.

dans les parties gauches de règles, l'état g est toujours au-dessus de l'extrémité gauche d'une lettre (confère les règles (i) et (v)) ; et que par application de ces règles (i) et (v), on retrouve sous l'extrémité gauche de la lettre un g ou un O.

dans les parties gauches de règles, l'état 0 est au dessus d'une lettre de co-arité un (confère règles (iv)) ; c'est-à-dire simultanément au dessus de l'extrémité droite et gauche d'une lettre ; et que par application de ces règles (iv), on retrouve sous l'extrémité gauche de la lettre un g ou un 0 , et sous l'extrémité droite de la lettre un d ou un 0 .

Ces trois constatations suffisent pour induire les énoncés précédents. \square

La propriété qui suit, constitue en quelque sorte une caractérisation des dags planaires connexes à partir de l'automate DG. Elle établit que tout dag connexe est reconnu par l'automate DG à partir d'un mot unique (car l'automate est ascendant déterministe) du langage $0 + g \otimes c^* \otimes d$.

Accessoirement, cette propriété établit que le taillis des dags connexes est reconnaissable.

V.2.3. - Propriété

$$\forall \delta \in \hat{DP}(\Sigma)$$

$$\exists w_1 \in (0 + g \otimes c^* \otimes d) \quad \exists w_2 \in 0^* \text{ tels que } w_1 \cdot \delta \stackrel{*}{\underset{DG}{|}} \delta \cdot w_2 \iff \delta \text{ est connexe}$$

Preuve :

\Rightarrow : • Soit $\delta \in \hat{DP}(\Sigma)$, $w_1 \in g \otimes c^* \otimes d$, $w_2 \in 0^*$ tels que $w_1 \cdot \delta \stackrel{*}{\underset{DG}{|}} \delta \cdot w_2$.
Montrons par l'absurde que δ est nécessairement connexe :

Si δ ne l'était pas, on aurait alors $\delta = \delta' \otimes \delta''$, $w_1 = w_1' \otimes w_1''$ et $w_2 = w_2' \otimes w_2''$ tels que

$$w_1' \cdot \delta' \stackrel{*}{\underset{DG}{|}} \delta' \cdot w_2' \text{ avec } w_1' \in g \otimes c^* \text{ et } w_2' \in 0^*$$

$$w_1'' \cdot \delta'' \stackrel{*}{\underset{DG}{|}} \delta'' \cdot w_2'' \text{ avec } w_1'' \in c^* \otimes d \text{ et } w_2'' \in 0^*$$

Or, dans w_1' on a (nombre de g - nombre de d) = 1, et l'application d'une règle quelconque de DG conserve cette quantité, w_2' ne peut donc appartenir à O^* .

De même, dans w_1'' on a (nombre de d - nombre de g) = 1, et l'application d'une règle quelconque de DG conserve cette quantité, w_2'' ne peut donc appartenir à O^* \square

• Soit $d \in \hat{D}\hat{P}(\Sigma)$, $w_2 \in O^*$ et $O \cdot \delta \stackrel{*}{\underset{DG}{|}} \delta \cdot w_2$: δ est connexe puisque de co-arité 1. \square

\Leftarrow : • Soit δ connexe alors $\exists w \in (O + g \otimes c^* \otimes d)^*$, $\exists w' \in O^*$ tels que $q \cdot \delta \stackrel{*}{\underset{DG}{|}} \delta \cdot w'$ (Propriété VI.2.2.1.).

- Si δ est de co-arité 1, alors $w = O$ (seul mot de longueur un du langage $(O + g \otimes c^* \otimes d)^*$). \square

- Si δ est de co-arité et de co-arité strictement à 1

- δ a une et une seule arête droite ;
- δ a une et une seule arête gauche ;
- l'arête droite et l'arête gauche de δ sont distinctes et sur les bords.

On conclut, d'après la Propriété V.2.2. que $w \in g \otimes c^* \otimes d$. \square

Pour établir la propriété suivante, attestant que les taillis connexes reconnaissables peuvent être reconnus par des automates possédant comme contrôles haut et bas des états initiaux et finaux, on doit s'astreindre à des préliminaires fastidieux.

Préliminaires à la propriété V.2.4. :

1) Quelques notations

a) Soit $M = \langle \Sigma, Q, I, F, R \rangle$ un automate d'états finis de dags

- On appellera $A_I = \langle Q, V, \{v_o\}, \{v_f\}, \mu \rangle$ un automate d'états finis de mots qui reconnaît I et tel que :

- V_0 est un état strictement initial, c'est-à-dire que

$$V \times Q \times \{v_0\} \cap \mu = \emptyset$$

- V_f est un état final "puits", c'est-à-dire que

$$\{v_f\} \times Q \times V \cap \mu = \emptyset$$

Soit $V_1 = V \setminus \{V_0, V_f\}$ (i.e. les états intermédiaires).

- $\forall v \in V$ il existe une suite de transitions de μ conduisant de cet état v à l'état final V_f

et

il existe une suite de transitions de μ conduisant de l'état V_0 à l'état v .

• On appellera

$$\Delta v_i v_j = \{(v_i, q_1, v_1) \otimes (v_1, q_2, v_2) \dots \otimes (v_{n-1}, q_n, v_j)\}$$

l'ensemble des suites de transitions chaînées qui démarrent par v_i et se terminent par v_f ; $v_1, v_2 \dots v_{n-1} \in V_1$ (i.e. on ne passe jamais par v_0 , ou par v_f).

De façon tout-à-fait symétrique, on notera

$A_F = \langle Q, K, \{k_0\}, \{k_f\}, \lambda \rangle$ un automate d'états finis de mots qui reconnaît F avec les mêmes contraintes sur l'ensemble K .

$\Delta k_i k_j$ désignera alors l'ensemble des suites de transitions (de $K \times Q \times K$) chaînées qui commencent par k_i et finissent par k_j .

b) Soit $Q_1 \times Q_2 \times \dots \times Q_n$ un ensemble produit d'ensembles d'états.

On notera Π_{Q_i} la projection de $Q_1 \times \dots \times Q_n$ sur Q_i :

$$\Pi_{Q_i} : Q_1 \times Q_2 \times \dots \times Q_n \rightarrow Q_i$$

$$(q_1, q_2, \dots, q_n) \rightsquigarrow q_i$$

et par extension

on notera $\Pi_{Q_{i_1} \times Q_{i_2} \times \dots \times Q_{i_n}}$ l'application de

$$Q_1 \times Q_2 \times \dots \times Q_p \rightarrow Q_{i_1} \times Q_{i_2} \times \dots \times Q_{i_n}$$

$$w \rightsquigarrow (\Pi_{Q_{i_1}}(w), \Pi_{Q_{i_2}}(w), \dots, \Pi_{Q_{i_n}}(w))$$

2) Rappels de construction connue (cf. II.1.4.)

a) Soit $M = \langle \Sigma, Q, I, F, R \rangle$ un automate de dag.

On peut construire les deux automates de dags suivants :

$$\bullet M_1 = \langle \Sigma, K \times Q \times K, \Delta_{K_0, K_f} \cap \Pi_Q^{-1}(I), \lambda^*, R_1 \rangle$$

où $R_1 = \{m \cdot a \vdash a \cdot m' / \exists k_i, k_j \in K \text{ tels que } m \text{ et } m' \in \Delta_{k_i, k_j} \text{ et}$

$$\Pi_Q(m) \cdot a \vdash a \cdot \Pi_Q(m') \in R\}$$

$$\bullet M_2 = \langle \Sigma, V \times Q \times V, \mu^*, \Delta_{V_0, V_f} \cap \Pi_Q^{-1}(F), R_2 \rangle$$

où $R_2 = \{m \cdot a \vdash a \cdot m' / \exists v_i, v_j \in V \text{ tels que } m \text{ et } m' \in \Delta_{v_i, v_j} \text{ et}$

$$\Pi_Q(m) \cdot a \vdash a \cdot \Pi_Q(m') \in R\}$$

Ces deux automates sont tels que $T(M) = T(M_1) = T(M_2)$.

3) Une propriété intéressante

$\forall \delta \in \hat{D}\mathcal{P}(\Sigma)$ connexe alors

$\exists m, m' \in (K \times Q \times K)^*$ tels que $m \cdot \delta \stackrel{*}{\underset{R_1}{\mid}} \delta \cdot m' \Rightarrow$

$\exists k_i, k_j \in K$ tels que m et $m' \in \Delta_{k_i, k_j}$.

Preuve :

Par récurrence sur le nombre de lettres de δ

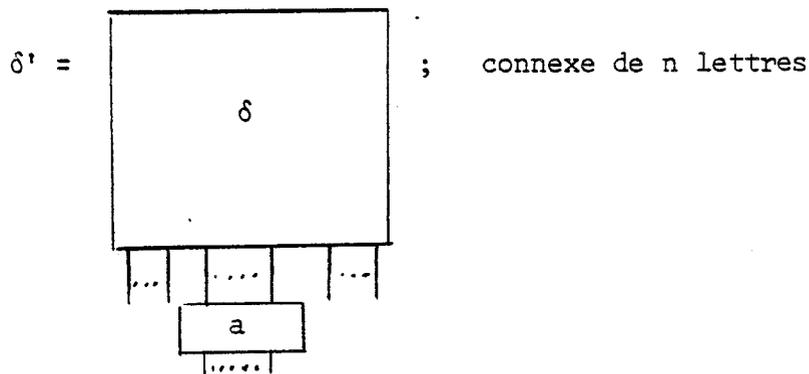
• $\delta \in \Sigma \Rightarrow m \cdot \delta \stackrel{*}{\underset{R_1}{\mid}} \delta \cdot m' \in R_1$ et les règles de R_1 sont telles que m et m' appartiennent à un même Δ_{k_i, k_j} .

• Hypothèse de récurrence : $\forall \delta \in \hat{D}\mathcal{P}(\Sigma)$, δ composé de moins de n lettres

$$m \cdot \delta \stackrel{*}{\underset{R_1}{\mid}} \delta \cdot m' \Rightarrow m \text{ et } m' \in \Delta_{k_i, k_j}$$

• Soit δ' composé de $n+1$ lettres, tel qu'il existe m et $m' \in (K \times Q \times K)^*$ et $m \cdot \delta' \stackrel{*}{\underset{R_1}{\mid}} \delta' \cdot m'$;

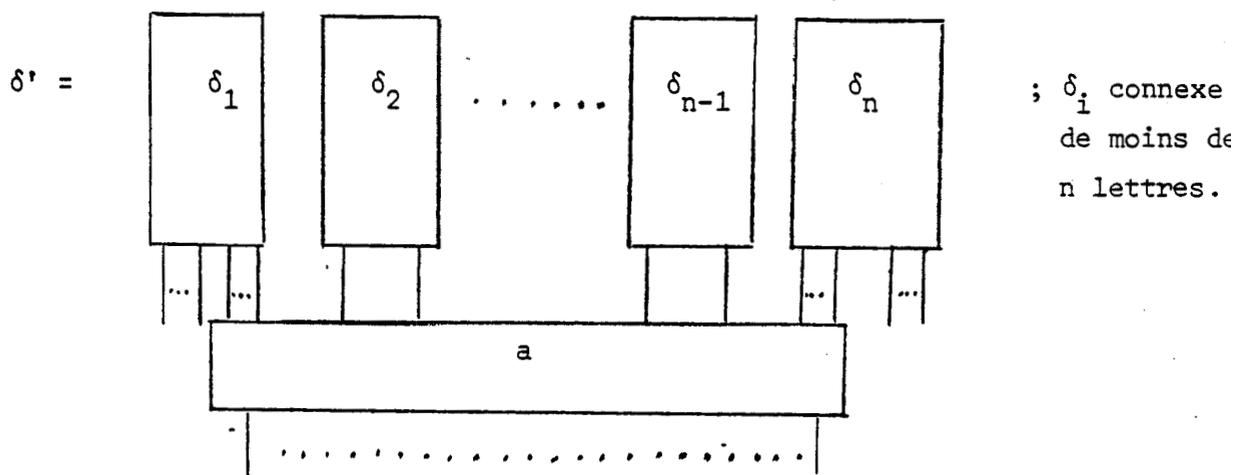
1^{er} cas :



• $m \cdot \delta \stackrel{*}{\underset{R_1}{\mid}} \delta \cdot m'_1$ avec (d'après H.R) m et m'_1 appartenant à un même Δ_{k_i, k_j}

• m' est obtenu à partir de m'_1 sans briser la chaîne, d'où m et m' appartiennent au même Δ_{k_i, k_j} .

2^{ème} cas :



d'après l'hypothèse de récurrence, on a :

$$\bullet m = m_1 \otimes m_2 \otimes \dots \otimes m_n$$

$$(m_1 \otimes m_2 \dots \otimes m_n) \cdot (\delta_1 \otimes \delta_2 \dots \otimes \delta_n) \stackrel{*}{\underset{R_1}{|}} (\delta_1 \otimes \delta_2 \otimes \dots \otimes \delta_n) \cdot (m'_1 \otimes m'_2 \otimes \dots \otimes m'_n)$$

avec $\forall i \in \{1, 2, \dots, n\} \exists k_{1,i}, k_{2,i} \in K$ tels que m_i et $m'_i \in \Delta_{k_{1,i}, k_{2,i}}$

• l'application d'une règle de R_1 sur la lettre a impose que

$$\forall i \in \{1, 2, \dots, n-1\} \quad k_{2,i} = k_{1,i+1}$$

d'où $m_1 \otimes m_2 \dots \otimes m_n$ et $m'_1 \otimes m'_2 \dots \otimes m'_n \in \Delta_{k_{1,1}, k_{2,n}}$

$\Rightarrow m$ et $m' \in \Delta_{k_{1,1}, k_{2,n}}$. \square

V.2.4. - Propriété

Quelque soit T taillis reconnaissable connexe (i.e. $\forall \delta \in T \quad \delta$ connexe), il existe un automate $M = \langle \Sigma, Q, I, F, R \rangle$ où $I = Q_0^*$, $Q_0 \subset Q$ et $F = Q_f^*$, $Q_f \subset Q$; tel que $T(M) = T$.

Preuve :

Comme T est reconnaissable, il existe par définition un automate $M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$, tel que $T(M_1) = T$.

Le Théorème II.1.4. nous assure que l'on peut construire à partir de M_1 , un automate $M_2 = \langle \Sigma, Q_2, I_2, F_2, R_2 \rangle$ reconnaissant le même taillis et tel que $F_2 = Q_{2f}^*$, $Q_{2f} \subset Q_2$.

Soit $A = \langle Q_2, V, \{v_0\}, \{v_f\}, \mu \rangle$ l'automate de mots qui reconnaît I_2 , avec les caractéristiques énoncées en Préliminaires 1.

On notera : μ_0 l'ensemble des transitions commençant par v_0 (i.e. $\mu_0 = \mu \cap \{v_0\} \times Q_2 \times V$) ; μ_f l'ensemble des transitions aboutissant à v_f (i.e. $\mu_f = \mu \cap V \times Q_2 \times \{v_f\}$) ; μ_{of} l'ensemble des transitions permettant de passer de v_0 à v_f (i.e. $\mu_{of} = \mu \cap \{v_0\} \times Q_2 \times \{v_f\}$). L'automate DG est caractérisé par le quintuplet $\langle \Sigma, H, H^*, \{0\}^*, P \rangle$ (cf. V.1.).

Nous allons montrer que :

l'automate $M = \langle \Sigma, Q, I, F, R \rangle$ où

$$Q = V \times Q_2 \times V \times H$$

$$I = ((\mu_0 \times \{g\}) \cup (\mu_f \times \{d\}) \cup (\mu \times \{c\}) \cup (\mu_{of} \times \{0\}))^*$$

$$F = (V \times Q_{2f} \times V \times \{0\})^*.$$

$R = \{m \cdot a \vdash a \cdot m' \text{ telles que :}$

$$\bullet \Pi_{q_2}(m) \cdot a \vdash a \cdot \Pi_{q_2}(m') \in R_2$$

$$\bullet \Pi_H(m) \cdot a \vdash a \cdot \Pi_H(m') \in P$$

$$\bullet \Pi_{V \times Q_2 \times V}(m) \text{ et } \Pi_{V \times Q_2 \times V}(m') \text{ appartiennent au même } \Delta_{v_i v_j}$$

reconnait le taillis T .

a) $T(M) \subset T$

$$\forall \delta \in T(M) \quad \exists m \in I \text{ et } m' \in F \text{ tels que } m \cdot \delta \stackrel{*}{\mid} \delta \cdot m'$$

• Par projection sur la composante H des états on a :

$$\Pi_H(m) \cdot \delta \stackrel{*}{\mid} \delta \cdot \Pi_H(m')$$

Comme $m' \in F$, $\Pi_H(m') \in \mathcal{O}^*$. On en conclut donc, étant donné que δ est connexe que $\Pi_H(m) \in (\mathcal{O} + g \otimes c^* \otimes d)$. (cf. Propriété V.2.3.).

• Par projection sur les composantes $V \times Q_2 \times V$ des états on a :

$$\Pi_{V \times Q_2 \times V}(m) \cdot \delta \stackrel{*}{\mid} \delta \cdot \Pi_{V \times Q_2 \times V}(m')$$

Attendu que δ est connexe, on en déduit que $\Pi_{V \times Q_2 \times V}(m)$ est suite de transitions chaînées de μ (i.e. $\exists v_i, v_j \in V$ tels que $\Pi_{V \times Q_2 \times V}(m) \in \Delta_{v_i v_j}$) (cf. Propriété intéressante - Préliminaires 3).

De ces constatations, il résulte que :

$$\left. \begin{array}{l} \cdot \Pi_{V \times Q_2 \times V}(m) \in \Delta_{v_i v_j} \\ \cdot \Pi_H(m) \in (\mathcal{O} + g \times c^* \times d) \\ \cdot m \in I \end{array} \right\} \Rightarrow \Pi_{V \times Q_2 \times V}(m) \in \Delta_{v_o v_f}$$

Or, si $\Pi_{V \times Q_2 \times V}(m)$ est une suite de transitions chaînées de μ , démarrant en v_o , et aboutissant en v_f , cela signifie que $\Pi_{Q_2}(m)$ appartient au langage reconnu par A, soit I_2 .

Il suffit, maintenant de regarder ce que l'on obtient par projection sur la composante Q_2 des états :

$$\left. \begin{array}{l} \cdot \Pi_{Q_2}(m) \cdot \delta \Big|_{R_2}^* \delta \cdot \Pi_{Q_2}(m') \\ \cdot \Pi_{Q_2}(m) \in I_2 \\ \cdot \Pi_{Q_2}(m') \in (Q_{2f})^* \end{array} \right\} \Rightarrow \delta \in (T(M_2) = T) \quad \square$$

b) $T \subset T(M)$

Ceci revient à démontrer que $T(M_2) \subset T(M)$, ce qui est trivialement vérifié. \square

V.3. - CONCLUSION

On constate à travers ce chapitre, l'intérêt d'étudier les dags connexes comme composantes des dags plus généraux. On découvre, en particulier, que dans les dags connexes, la notion de reconnaissabilité peut être épurée des contrôles haut et bas. Ceci exprime bien le fait que sur des dags connexes on peut toujours parvenir à faire coïncider deux informations par un mécanisme de transmission tel que les règles de transitions locales ; alors que ceci est manifestement impossible pour des dags non connexes (l'information ne pouvant passer d'une composante connexe à une autre) : ces informations sont donc consignés dans les contrôles haut et bas.

CHAPITRE VI

LA RATIONALITE DANS LES DAGS PLANAIRES

PLAN DU CHAPITRE VI

VI.1. - LES MAGMOÏDES TYPES

VI.2. - LES MORPHISMES DE MAGMOÏDES TYPES

VI.3. - LES MAGMOÏDES TYPES LIBRES

VI.4. - RATIONALITE DANS LES DAGS PLANAIRES : DEFINITIONS

VI.5. - JUSTIFICATION DE LA DEFINITION DE RATIONALITE
DANS LES DAGS PLANAIRES

INTRODUCTION

Il ne nous reste, pour compléter notre étude influencée par les résultats obtenus sur les mots, et les arbres, à définir sur les dags planaires une notion de rationalité. L'ensemble des taillis reconnaissables est-il, comme le sont les ensembles des langages et de forêts régulières, la clôture par certaines opérations des ensembles finis de dags, ou encore peut-on par combinaison d'un nombre fini d'opérateurs sur des sous-ensembles finis obtenir tous les taillis reconnaissables. Dans les mots, le problème est bien connu ; tout langage régulier est solution d'un système algébrique que l'on peut résoudre ; mais bien que les forêts régulières d'arbres soient également solution de système algébrique, leur résolution nécessite l'introduction d'opérateurs typés. Dans les dags, on se verra contraint, à bityper les lettres à partir desquelles on fabriquera les expressions rationnelles. Un cadre algébrique nouveau sera adopté pour manipuler ces expressions : le magmoïde typé.

A travers cette définition de rationalité sur les dags, notre but est de parvenir au même théorème classique que dans les mots et les arbres : l'équivalence entre les notions de reconnaissabilité, de régularité et de rationalité.

VI.1. - LES MAGMOÏDES TYPES

Nous allons définir à la manière de Jacob et Corbeel dans [] la notion de magmoïde typé. Pour ce faire, nous allons particulariser les liaisons qui peuvent exister entre ses éléments. Les éléments du magmoïde typé auront deux caractéristiques particulières : une co-signature et une signature mot d'un monoïde T^* , T étant un ensemble fini de symboles de types. Et on imposera au produit de composition d'être compatible avec les signatures et co-signatures des éléments du magmoïde.

IV.1.1. - Définitions formelles

Soit T un ensemble fini de symboles de type.

• Un alphabet bitypé (Σ, T) est un ensemble fini de lettres qui sont des triplets de la forme (t, a, t') avec :

- $t \in T^*$ appelée co-signature de cette lettre
- $a \in \Sigma$ appelée étiquette de cette lettre
- $t' \in T^*$ appelée signature de cette lettre.

On notera aussi parfois ces triplets $a_{t'}^t$.

• Un magmoïde typé est un sextuplet $\langle M, T, \cdot, \otimes, E, e_\epsilon \rangle$ qui sera noté (M, T) , où M est un ensemble, T un ensemble fini de symbole de types, \cdot une opération partielle binaire sur M , \otimes une opération binaire sur M , E un sous-ensemble distingué de M et e_ϵ un élément distingué de M ; et qui vérifié les axiomes suivants :

A1 : $\forall u \in T^*, \forall v \in T^*$ il existe une partie M_{v}^u de M , appelée fibre u - v de M .
Les fibres de M constituent une partition de M .

M est donc un ensemble bitypé, tout élément de M_{v}^u sera dit de signature v et de co-signature u .

A2 : $\forall u \in T^*, \forall v \in T^*, \forall u' \in T^*, \forall v' \in T^*, \forall m \in M_{v}^u, \forall m' \in M_{v'}^{u'}$,

$m \cdot m'$ est défini ssi $v = u'$. On a alors $m \cdot m' \in M_{v'}^u$.

Cette opération \cdot est appelée produit de composition (typé).

A'2 : Le produit de composition (typé) est associatif.

A3 : $\forall u \in T^*, \forall v \in T^*, \forall u' \in T^*, \forall v' \in T^*, \forall m \in M_{v}^u, \forall m' \in M_{v'}^{u'}$

$$m \otimes m' \in M_{v \cdot v'}^{u \cdot u'}$$

Cette opération \otimes est appelée produit tensoriel.

A'3 : Le produit tensoriel est associatif.

A4 : $\forall m_1, m_2, m'_1, m'_2 \in M$

$$(m_1 \cdot m_2) \otimes (m'_1 \cdot m'_2) \text{ défini} \Rightarrow (m_1 \otimes m'_1) \cdot (m_2 \otimes m'_2) = (m_1 \cdot m_2) \otimes (m'_1 \cdot m'_2)$$

A5 :

A tout symbole de type $t \in T$, est associé un et un seul élément de E qui a comme signature et co-signature t . Nous noterons e_t l'élément de E associé à t .

En posant $e_u = e_{u_1} \cdot e_{u_2} \dots \cdot e_{u_n}$ tel que $u = u_1 \cdot u_2 \dots u_n$, $u_i \in T$

on a : $\forall m \in M_{\mathcal{V}}^u \in e_u \cdot m = m$ et $m \cdot e_{\mathcal{V}} = m$

$e_{\varepsilon} \in M_{\varepsilon}^{\varepsilon}$, où ε désigne l'élément neutre du monoïde T^* et $\forall m \in M_{\mathcal{V}}^u$
 $e_{\varepsilon} \otimes m = m \otimes e_{\varepsilon} = m$.

Les éléments e_u sont donc neutres pour la composition ; e_{ε} est également neutre pour le produit tensoriel.

VI.1.2. - Exemple de magmoïde typé

Nous avons illustré la définition de magmoïde en donnant comme exemple l'ensemble des relations définies de E^i sur E^j (E ensemble donné), muni de la composition des relations et de "la juxtaposition" de relations.

Par contre, cette structure algébrique de magmoïde ne peut s'appliquer à l'ensemble des relations définies à l'aide de deux ensembles E et F muni des mêmes lois de composition des relations et de "juxtaposition" des relations. On n' peut plus se contenter pour composer deux relations de vérifier que le nombre de composantes de l'ensemble d'arrivée de la première relation est égal au nombre de composantes de l'ensemble de départ de la deuxième relation (respect de la graduation), mais il faut s'assurer que chacune des composantes appartienne au même ensemble (respect d'un type : l'ensemble de référence).

Soit $T = \{E, F\}$

$$R(E, F) = \bigcup_{u, v \in T^*} R_v^u(E, F) \text{ avec}$$

$$R_v^u(E, F) = \{\text{ensemble des relations de } u_1 \times u_2 \dots \times u_n \text{ dans } v_1 \times v_2 \dots \times v_p \\ u = u_1 \cdot u_2 \dots u_n; u_i \in T \text{ et } v = v_1 \cdot v_2 \dots v_p, v_j \in T\}.$$

$(R(E, F), T)$ muni des lois de composition et de juxtaposition des relations, de l'identité de E sur E , de l'identité de F sur F , et de la relation définie du vide sur le vide (qui correspond à e_ϵ) est un magmoïde typé.

Remarque :

La notion de magmoïde typé est une généralisation de la notion de magmoïde. En effet, tout magmoïde peut être considéré comme un magmoïde typé dont l'ensemble des symboles de type est réduit à un seul élément.

On montre que tout magmoïde typé $(M, \{t\})$ est isomorphe à un magmoïde M .

Soit ℓ l'isomorphisme de t^* dans \mathbb{N} tel que

$$\forall u \in t^* \quad \ell(u) = \text{longueur du mot } u.$$

Tout fibre M_v^u du magmoïde $(M, \{t\})$ s'identifie de façon unique à une fibre $M_{\ell(u)}^{\ell(v)}$ et les axiomes du magmoïde typé se réduisent alors aux axiomes du magmoïde.

VI.2. - LES MORPHISMES DE MAGMOÏDES TYPES

Soient $\langle M, T, \cdot, \emptyset, E, e_\epsilon \rangle$ et $\langle M', T', \cdot, \emptyset, E', e_{\epsilon'} \rangle$ deux magmoïdes typés, une application ϕ de (M, T) dans (M', T') est un morphisme de magmoïdes typés si :

C1 : ϕ se comporte comme un morphisme vis-à-vis des types ; i.e. il existe un morphisme h de T^* dans T'^* .

$$\forall u \in T^*, \forall v \in T^* \quad \forall m \in M_v^u \quad \phi(m) \in M_{h(v)}^{h(u)}.$$

C2 : ϕ préserve les éléments neutres :

$$\forall t \in T \quad \phi(e_t) = e_{h(t)}$$

C3 : ϕ est compatible avec les opérations, i.e.

$$\forall m \in M_V^u, \forall m' \in M_{V'}^{u'} \quad \phi(m \cdot m') = \phi(m) \cdot \phi(m') \text{ et } \phi(m \otimes m') = \phi(m) \otimes \phi(m')$$

Remarquons que la condition 1 assure que lorsque $m \cdot m'$ est défini $\phi(m) \cdot \phi(m')$ l'est aussi.

VI.3. - LE MAGMOÏDE LIBRE TYPE ENGENDRE PAR UN ALPHABET BITYPE (Σ, T)

Soit (Σ, T) un alphabet bitypé.

On construit alors de façon standard le magmoïde libre typé engendré par (Σ, T) comme l'ensemble des expressions syntaxiques bien formées bâties à partir des lettres de (Σ, T) quotienté par les axiomes du magmoïde typé.

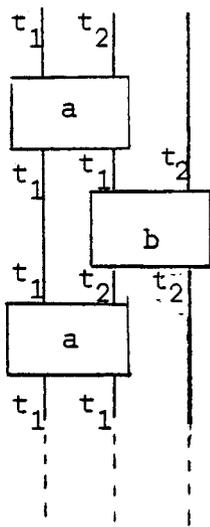
Le magmoïde libre engendré par (Σ, T) se note $\hat{DP}(\Sigma, T)$.

Les éléments de ce magmoïde peuvent s'interpréter comme des dags planaires dont les noeuds, et les arcs seraient étiquetés. La jonction de deux arcs n'est possible qu'à la condition qu'ils portent la même étiquette.

Exemple :

$$\text{Soit } T = \{t_1, t_2\}, (\Sigma, T) = \{(t_1 t_2, a, t_1 t_1), (t_1 t_2, b, t_2 t_2)\}$$

Les éléments de $\hat{DP}(\Sigma, T)$ sont en terme de dags planaires :

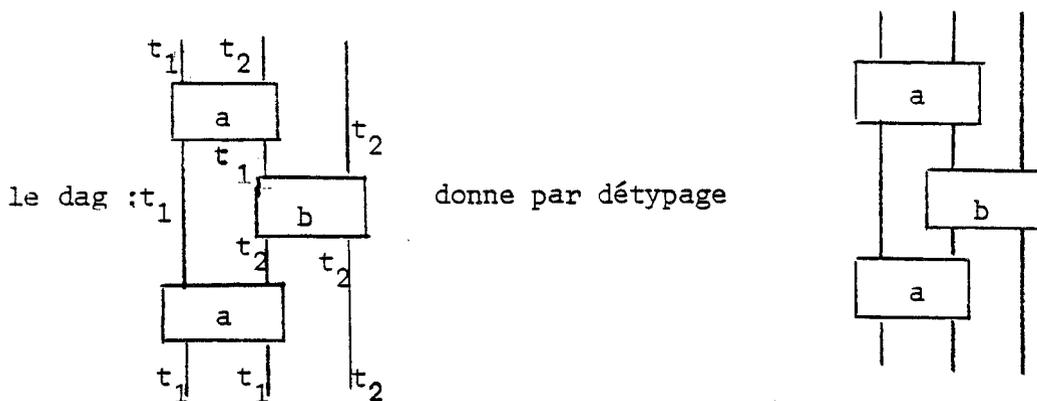


VI.3.1. - L'opération de détypage

Le dag obtenu "en gommant" les types de l'interprétation graphique d'un élément de $\hat{DP}(\Sigma, T)$, est manifestement l'interprétation graphique d'un élément de $\hat{DP}(\Sigma)$. On dira que l'élément de $\hat{DP}(\Sigma)$ ainsi obtenu, l'a été par détypage de l'élément de $\hat{DP}(\Sigma, T)$.

Exemple :

Reprenons T et (Σ, T) tels qu'ils ont été définis ci-dessus.



Formellement, cette opération de détypage est un morphisme $\bar{\tau}$ de $\hat{DP}(\Sigma, T)$ dans $\hat{DP}(\Sigma)$ définie par :

$$- \forall (t, a, t') \in (\Sigma, T) \quad \bar{\tau}((t, a, t')) = a \in \hat{DP}(\Sigma)_{\ell(t')}^{\ell(t)}$$

Ce morphisme n'est pas d'une manière générale injectif.

VI.3.2. - Propriété du magmoïde typé libre

Quelque soit le magmoïde typé libre, son détypage est un taillis reconnaissable.

$\hat{VDP}(\Sigma, T) \quad \bar{\tau}(\hat{DP}(\Sigma, T)) = \{\bar{\tau}(m) / m \in \hat{DP}(\Sigma, T)\}$ est un taillis reconnaissable.

Preuve :

Soit $M = (\Sigma, T, T^*, T^*, R)$ automate de dag planaire

où $R = \{(t_1 \otimes \dots \otimes t_n) \cdot a \vdash a \cdot (t'_1 \otimes \dots \otimes t'_p) / (t_1 \dots t_n, a, t'_1 \dots t'_p) \in (\Sigma, T)\}$

On se convaint aisément que le taillis reconnu par cet automate est bien le sous-ensemble de $\hat{DP}(\Sigma)$ obtenu en détypant $\hat{DP}(\Sigma, T)$.

Remarque :

Cette propriété établit l'analogie parfaite qui existe entre le rôle des types dans le magmoïde typé et le rôle des états dans un automate d'états finis de dags.

Imaginons qu'au fil d'une reconnaissance, on marque les lettres par la transformation d'états résultant de la règle appliquée sur la lettre, on obtiendrait alors un élément de magmoïde typé.

De même, on conçoit aisément que les concordances de types à l'intérieur du magmoïde typés puissent être vérifiées par un automate d'états finis.

On peut bien évidemment mettre en évidence des ensembles à structure de magmoïde typé, qui ne soient pas des structures libres :

VI.3.3. - Un exemple de magmoïde typé non libre

Soit (Σ, T) un alphabet bitypé.

Soit M le sous-ensemble de $T^* \times \hat{D}\hat{P}(\Sigma) \times T^*$ défini par :

$$\forall a_{t,t'}^t \in (\Sigma, T) \quad (t, a, t') \in M$$

On appellera M_v^u , le sous-ensemble de M dont la première composante des éléments est u et la seconde v .

$$\bullet \forall (u, m, v) \in M_v^u \quad \forall (u', m', v') \in M_{v'}^{u'}$$

$$(u, m, v) \otimes (u', m', v') = (uu', m \otimes m', vv')$$

$$\bullet \forall (u, m, v) \in M_v^u \quad \forall (u', m', v') \in M_{v'}^{u'}$$

$$(u, m, v) \cdot (u', m', v') \text{ existe ssi } v = u', \\ \text{et est alors égal à } (u, m \cdot m', u')$$

$$\bullet \forall t \in T \quad (t, e, t) \in M_t^t$$

$$\bullet (\varepsilon, e_0, \varepsilon) \in M_\varepsilon^\varepsilon.$$

L'ensemble M défini comme $\bigcup_{u, v \in T^*} M_v^u$ est un magmoïde typé, contenant (Σ, T)

mais n'est pas le magmoïde libre typé engendré par (Σ, T) .

Preuve :

S'il en était ainsi, on pourrait définir un morphisme de M dans $\hat{D}\hat{P}(\Sigma)$ par sa restriction à (Σ, T) et ceci de façon unique ; ce qui est contredit par l'exemple suivant :

$$\bullet (\Sigma, T) = \{a_{t_1}^{t_1}, a_{t_1}^{t_0}, b_{t_1}^{t_1}, b_{t_1}^{t_0}\}$$

• Soit l'application ϕ de (Σ, T) dans $\hat{D}\mathcal{P}(\Sigma)$ telle que

$$\phi(a_{t_1}^{t_1}) = a ; \phi(b_{t_1}^{t_1}) = a$$

$$\phi(a_{t_0}^{t_1}) = b ; \phi(b_{t_1}^{t_0}) = b$$

• On a $(t_1, ab, t_2) = (t_1, a, t_0) \cdot (t_0, b, t_1) = (t_1, a, t_1) \cdot (t_1, b, t_1)$ dans M.

Le calcul de $\phi((t_1, ab, t_2))$ par extension de ϕ en un morphisme donnerait soit $b \cdot b$, ou $a \cdot a$. \square

VI.4. - RATIONALITE DANS LES DAGS PLANAIRES : DEFINITIONS

Lors des chapitres précédents, nous avons mis en évidence deux familles de taillis incluses dans $\hat{D}\mathcal{P}(\Sigma)$. REC (les taillis reconnaissables) et REG (les taillis réguliers). Un taillis reconnaisable est caractérisé par l'existence d'un algorithme de décision de l'appartenance d'un élément de $\hat{D}\mathcal{P}(\Sigma)$ à ce taillis (automate d'états finis).

Un taillis régulier est caractérisé par l'existence d'un procédé de génération (grammaire régulière) des éléments de ce taillis.

Et nous avons montré que ces deux familles coïncident (REC = REG). Ce premier théorème s'inscrit dans la lignée de théorèmes similaires sur des classes d'objets plus pauvres : les mots et les arbres.

Sur les mots, on dispose d'un résultat supplémentaire : la classe des langages réguliers (ou reconnaissables) est la plus petite classe de langages contenant les langages finis et close par union, produit et étoile (Théorème de Kleene). Ce théorème offre une troisième caractérisation des langages réguliers (ou reconnaissables) par l'existence d'expressions décrivant les mots de ces langages : les expressions rationnelles de mots.

Dans cette partie, nous établirons un résultat de même nature concernant la classe des taillis réguliers (ou reconnaissables) ; nous montrerons qu'à tout taillis de REC (ou REG), on peut associer une expression rationnelle de dag et vice-versa.

VI.4.1. - Expressions rationnelles de dag

VI.4.1.1. - Définitions

Soit (Σ, T) un alphabet bitypé.

L'ensemble $Ex(\Sigma, T)$ des expressions simples construites à partir de l'alphabet (Σ, T) est défini récursivement par :

$$- \forall a_{\substack{t \\ t \\ s}}^c \in (\Sigma, T) \quad a_{\substack{t \\ t \\ s}}^c \in Ex(\Sigma, T)$$

$$- \forall exp1 \in Ex(\Sigma, T) \quad \forall exp2 \in Ex(\Sigma, T)$$

- $(exp1 + exp2) \in Ex(\Sigma, T)$
- $(exp1 \cdot exp2) \in Ex(\Sigma, T)$
- $(exp1 \circledast exp2) \in Ex(\Sigma, T)$

$$- \forall exp \in Ex(\Sigma, T)$$

- $exp^* \in Ex(\Sigma, T)$
- $exp^{\circledast} \in Ex(\Sigma, T)$

On appelle expression rationnelle de dags (sur (Σ, T)), tout triplet (I, exp, F) où I et F sont des langages rationnels inclus dans T^* et exp est une expression simple de $Ex(\Sigma, T)$.

On note $RAT(\Sigma, T)$, l'ensemble des expressions rationnelles de dags (sur (Σ, T)).

Par la suite, la référence systématique à l'alphabet bitypé (Σ, T) sera implicite, et on abrègera les notations $Ex(\Sigma, T)$ en Ex et $Rat(\Sigma, T)$ en Rat .

Notre objectif étant de faire correspondre les expressions de RAT et les taillis de REC (ou REG) ; il nous faut interpréter les expressions rationnelles comme sous-ensembles de $\hat{DP}(\Sigma)$.

VI.4.1.2. - Interprétation des expressions rationnelles de dags

a) Interprétation d'une expression simple

$$I : Ex \rightarrow 2^{\hat{DP}(\Sigma, T)}$$

définie par :

$$- \forall exp1 \in Ex \quad \forall exp2 \in Ex$$

$$I(exp1 \cdot exp2) = I(exp1) \cdot I(exp2)$$

$$= \{m_1 \cdot m_2 \in \hat{DP}(\Sigma, T) / m_1 \in I(exp1) \text{ et } m_2 \in I(exp2)\}$$

Propriété :

$$\forall exp1, exp2, exp3 \in Ex$$

$$I((exp1 \cdot exp2) \cdot exp3) = I(exp1 \cdot (exp2 \cdot exp3))$$

du fait de l'associativité du produit de composition dans $\hat{DP}(\Sigma, T)$.

Cette propriété nous autorise de façon classique une réduction du nombre de parenthèses dans l'écriture des expressions simples de dags.

Propriété :

On déduit immédiatement de la propriété précédente que :

$$\forall n \in \mathbb{N} \quad \forall i \in [1 \dots n] \quad \forall exp_i \in Ex$$

$$\tau(\text{exp1} \cdot \text{exp2} \cdot \dots \cdot \text{expn}) = I(\text{exp1}) \cdot \dots \cdot I(\text{expn})$$

- $\forall \text{exp} \in \text{Ex}$

$$I(\text{exp}^*) = \{e_p / p \in T^*\} \cup \bigcup_{i \in \mathbb{N}^*} I(\text{exp}^i)$$

où $\text{exp}^i = \underbrace{\text{exp} \cdot \text{exp} \cdot \dots \cdot \text{exp}}_{i \text{ fois}}$

- $\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex}$

$$I(\text{exp1} \odot \text{exp2}) = I(\text{exp1}) \otimes I(\text{exp2}) \cup$$

$$I(\text{exp1}) \otimes \{e_p / p \in T^*\} \cup$$

$$\{e_p / p \in T^*\} \otimes I(\text{exp2})$$

Propriété :

$\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex} \quad \forall \text{exp3} \in \text{Ex}.$

$$I(\text{exp1} \odot (\text{exp2} \odot \text{exp3})) = I((\text{exp1} \odot \text{exp2}) \odot \text{exp3})$$

du fait de l'associativité du produit tensoriel dans $\hat{D}\mathcal{P}(\Sigma, T)$.

Cette propriété nous autorise de façon classique une réduction du nombre de parenthèses dans l'écriture des expressions simples de dags.

- $\forall \text{exp} \in \text{Ex}$

$$I(\text{exp}^{\odot}) = \{e_p / p \in T^*\} \cup \bigcup_{i \in \mathbb{N}^*} \underbrace{I(\text{exp} \odot \text{exp} \odot \dots \odot \text{exp})}_{i \text{ fois}}$$

- $\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex}$

$$I(\text{exp1} + \text{exp2}) = I(\text{exp1}) \cup I(\text{exp2})$$

Propriété :

$$\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex} \quad \forall \text{exp3} \in \text{Ex}$$

$$I(\text{exp1} + (\text{exp2} + \text{exp3})) = I((\text{exp1} + \text{exp2}) + \text{exp3})$$

La propriété précédente amène la possibilité d'une réduction du nombre de parenthèses.

Propriété :

$$\forall n \in \mathbb{N}^* \quad \forall i \in [1 \dots n] \quad \forall \text{exp}_i \in \text{Ex}$$

$$I(\text{exp}_1 + \text{exp}_2 + \dots + \text{exp}_n) = \bigcup_{i \in [1 \dots n]} I(\text{exp}_i)$$

- l'interprétation d'une lettre se confond avec l'identité sur (Σ, T) .

b) Interprétation d'une expression rationnelle de dag

$$I : \text{RAT} \rightarrow 2^{\text{DP}(\Sigma)}$$

$(I, \text{exp}, F) \rightsquigarrow \{\bar{t}(m) / m \in I(\text{exp}) \text{ dont la co-signature appartient à } I \text{ et la signature à } F\}$.

Par la suite RAT désignera aussi bien l'ensemble des expressions rationnelles de dags que l'ensemble des taillis qu'elles ont comme interprétations.

c) Exemple d'expression rationnelle et de son interprétation

Soit $T = \{t_0, t_1\}$, ensemble fini de types.

Soit $(\Sigma, T) = \left\{ \begin{matrix} t_0 & t_0 t_1 \\ a_{t_0 t_1} & b_{t_0} \end{matrix} \right\}$, alphabet bitypé

et $r = (t_0, ((a_{t_0 t_1}^{\circledast})^{\circledast})^{\circledast} \cdot ((b_{t_0}^{\circledast t_1})^{\circledast})^{\circledast}, t_0) \in \text{RAT}$.

Etudions la composition du taillis qu'elle représente :

Pour ce faire, nous allons analyser la composition des éléments de l'interprétation de l'expression simple

$$\text{exp} = ((a_{t_0 t_1}^{t_0})^{\circlearrowleft})^* \cdot ((b_{t_0}^{t_0 t_1})^{\circlearrowleft})^* \text{ dont la signature et la co-signature}$$

sont égales à t_0 .

Notons que

$$\forall d \in I(\text{exp}) \exists i, j \in \mathbb{N} \text{ tel que } d \in [I((a_{t_0 t_1}^{t_0})^{\circlearrowleft})]^i \cdot [I((b_{t_0}^{t_0 t_1})^{\circlearrowleft})]^j$$

- le premier étage de $d \in I((a_{t_0 t_1}^{t_0})^{\circlearrowleft})$
 - la co-signature de d est t_0
- $$\left. \vphantom{\begin{matrix} \bullet \\ \bullet \end{matrix}} \right\} \Rightarrow \text{le premier étage de } d \text{ est } a_{t_0, t_1}^{t_0}$$
- le deuxième étage de $d \in I((a_{t_0 t_1}^{t_0})^{\circlearrowleft})$
 - sa co-signature est $t_0 t_1$
- $$\left. \vphantom{\begin{matrix} \bullet \\ \bullet \end{matrix}} \right\} \Rightarrow \text{le deuxième étage de } d \text{ est } a_{t_0 t_1}^{t_0}$$

d'une manière générale :

- le $k^{\text{ième}}$ étage ($k \leq i$) de $d \in I((a_{t_0 t_1}^{t_0})^{\circlearrowleft})$
 - sa co-signature est $\underbrace{t_0 t_1 \cdot t_1 \dots t_1}_{k-1 \text{ fois}}$
- $$\left. \vphantom{\begin{matrix} \bullet \\ \bullet \end{matrix}} \right\} \Rightarrow \text{le } k^{\text{ième}} \text{ étage de } d \text{ est } a_{t_0 t_1}^{t_0} \otimes \underbrace{e_{t_1 \dots t_1}}_{k-1 \text{ fois}}$$

- le $i+1^{\text{ième}}$ étage de $d \in I((b_{t_0}^{t_0 t_1})^{\circlearrowleft})$
 - sa co-signature est $\underbrace{t_0 t_1 \dots t_1}_{i \text{ fois}}$
- $$\left. \vphantom{\begin{matrix} \bullet \\ \bullet \end{matrix}} \right\} \Rightarrow \text{le } i+1^{\text{ième}} \text{ étage de } d \text{ est } \underbrace{b_{t_0}^{t_0 t_1}}_{i \text{ fois}} \otimes \underbrace{e_{t_1 \dots t_1}}_{i-1 \text{ fois}}$$

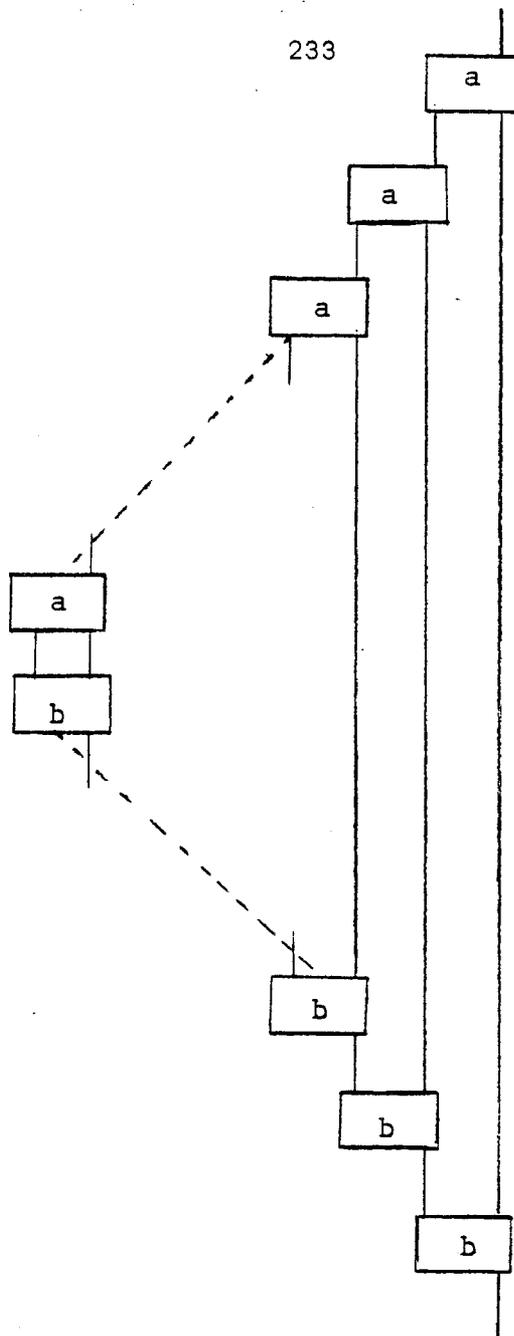
d'une manière générale :

$$\left. \begin{array}{l}
 \bullet \text{ le } i+l \text{ ième étage } (l \leq j) \in I((b_{t_0}^{t_0 t_1})^{\otimes l}) \\
 \bullet \text{ sa co-signature est } \underbrace{t_0 t_1 \dots t_1}_{i-l+1 \text{ fois}}
 \end{array} \right\} \Rightarrow \text{ le } i+l \text{ ième étage de } d \text{ est }$$

$$b_{t_0}^{t_0 t_1} \otimes \underbrace{e_{t_1 \dots t_1}}_{i-l \text{ fois}}$$

$$\left. \begin{array}{l}
 \bullet \text{ le dernier étage } (i+j \text{ ième}) \text{ est } b_{t_0}^{t_0 t_1} \otimes \underbrace{e_{t_1 \dots t_1}}_{i-j \text{ fois}} \\
 \bullet \text{ sa signature est } t_0
 \end{array} \right\} \Rightarrow i = j$$

On conclut de cette étude que les éléments de $I(\exp)$ sont des dags de la forme suivante :



VI.4.1.3. - Quelques propriétés des expressions simples de dags

Les opérateurs qui interviennent dans les expressions simples de dags expriment la composition des dags dans deux directions :

- verticale
 - le produit de composition contrôlé par les types : •
 - le produit de composition contrôlé par les types et itéré : X^*

- horizontale {
 - la mise asynchrone en parallèle : \circledast
 - la mise asynchrone en parallèle itérée : \circledast^2

Dans chacune des deux directions, on observe des propriétés analogues à celles des expressions rationnelles de mots :

VI.4.1.3.1. - Propriété

$$\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex}$$

- $(\text{exp1}^* \cdot \text{exp2}^*)^* = (\text{exp1} + \text{exp2})^*$
- $(\text{exp1}^{\circledast} \circledast \text{exp2}^{\circledast})^{\circledast} = (\text{exp1} + \text{exp2})^{\circledast}$
- $\text{exp1}^{\circledast} = \text{exp}_1 \circledast \text{exp}_1^{\circledast} = \text{exp}_1^{\circledast} \circledast \text{exp}_1$

L'introduction des deux types d'opérateurs, fait apparaître de nouvelles propriétés.

VI.4.1.3.2. - Propriété

$$\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex}$$

- $(\text{exp1}^{\circledast} + \text{exp2}^{\circledast})^* = (\text{exp1}^{\circledast} \cdot \text{exp2}^{\circledast})^* = (\text{exp1}^{\circledast} \circledast \text{exp2}^{\circledast})^* = ((\text{exp1} + \text{exp2})^{\circledast})^*$

Preuve :

$$\text{a) } (\text{exp1}^{\circledast} \cdot \text{exp2}^{\circledast})^* \underset{1}{\subset} (\text{exp1}^{\circledast} + \text{exp2}^{\circledast})^* \underset{2}{\subset} (\text{exp1}^{\circledast} \circledast \text{exp2}^{\circledast})^* \underset{3}{\subset} ((\text{exp1} + \text{exp2})^{\circledast})^*$$

- Les inclusions 1 et 3 sont évidentes

- l'inclusion 2 est vraie car $\text{exp1}^{\circledast} \subset (\text{exp1}^{\circledast} \circledast \text{exp2}^{\circledast})$ et $\text{exp2}^{\circledast} \subset (\text{exp1}^{\circledast} \circledast \text{exp2}^{\circledast})$

$$\text{b) } ((\text{exp1} + \text{exp2})^{\circledast})^* \subset (\text{exp1}^{\circledast} \cdot \text{exp2}^{\circledast})^*$$

Preuve :

Nous allons montrer que $(\text{exp1} + \text{exp2})^{\circledast} \subset (\text{exp1}^{\circledast} \cdot \text{exp2}^{\circledast})$

• $\forall d \in (\text{exp1} + \text{exp2})^{\circledast}$ d est une juxtaposition d'éléments de l'ensemble $\{\text{ep} / \text{p} \in T^*\} \cup I(\text{exp1}) \cup I(\text{exp2})$.

• Tout élément de $I(\text{exp}_2)$ peut s'écrire sous la forme d'un élément de $\{\text{ep} / \text{p} \in T^*\}$ composé avec lui-même, de même que tout élément de $I(\text{exp1})$ peut s'écrire sous la forme de lui-même composé avec un élément de $\{\text{ep} / \text{p} \in T^*\}$, et tout élément de $\{\text{ep} / \text{p} \in T^*\}$ peut être recomposé à lui-même.

\Rightarrow d peut être considéré comme la composition de deux éléments, l'un appartenant à $I(\text{exp}_1^{\circledast})$, l'autre à $I(\text{exp}_2^{\circledast})$.

□

Remarque :

Malgré ces quelques propriétés, les expressions simples de dags ne peuvent être considérées comme les solutions d'un système d'équations comme dans le cas des expressions rationnelles de mots.

VI.5. - JUSTIFICATION DE LA DEFINITION DE RATIONALITE SUR LES DAGS

Notre volonté d'aboutir au résultat $\text{REC} = \text{REG} = \text{RAT}$ a contrebalancé notre désir de poser les définitions les plus naturelles et les plus simples de rationalité. L'ambition de cette partie est de vous persuader que les choix pris constituent le meilleur compromis entre ces deux intérêts. Deux points essentiels méritent une justification :

- 1 - L'introduction des types dans la définition d'une expression rationnelle.
- 2 - Le choix de l'interprétation de l'opérateur \circledast de mise en parallèle.

VI.5.1. - L'introduction des types

Le recours à des lettres bitypées, et par conséquent l'introduction lors de l'interprétation d'une expression rationnelle d'une fonction de détypage s'appuient sur des considérations faites sur les forêts régulières d'arbres.

En effet, il est connu qu'on ne peut exprimer rationnellement les forêts régulières d'arbres avec un unique opérateur d'itération, mais que ces opérateurs doivent être typés par des "étiquettes" dans le cas de boucles entrelacées.

Nous retraçons dans la suite, comment peuvent être abordés ces problèmes de types dans :

- a) Les mots et les expressions rationnelles de mots.
- b) Les arbres et les expressions rationnelles d'arbres.
- c) Les dags et les expressions rationnelles de dags.

Notons aussi que la présence dans les expressions rationnelles de dags d'un rationnel I, et d'un rationnel F qui filtrent les dags retenus en imposant une contrainte sur leur signature et co-signature, n'est que le pendant du contrôle haut et contrôle bas des automates d'états finis.

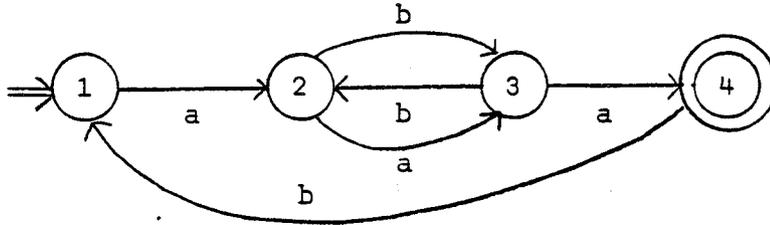
a) Dans les mots

Un langage régulier (ou reconnaissable) des mots est un langage pour lequel la succession des lettres obéit à un nombre fini de règles. La façon dont doivent se succéder les lettres d'un mot du langage peut être traduite en dimension un, par une expression rendant compte

- . de l'enchaînement des lettres : l'opérateur de concaténation (\cdot)
- . du choix entre différentes lettres : l'opérateur d'union ($+$)
- . de la reprise de séquences de lettres : l'opérateur d'itération ($*$)

Exemple :

Soit l'automate de mots :



Le langage reconnu par cet automate est :

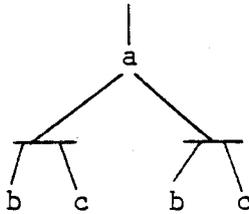
$$(a \cdot ((a+b) \cdot b)^* \cdot (a+b) \cdot ab)^* \cdot a((a+b) \cdot b)^* \cdot (a+b) \cdot a$$

L'opérateur d'itération est unique car on peut isoler parfaitement en dimension 1 par un jeu de parenthèses de facteur du mot susceptible d'être itéré.

b) Dans les arbres

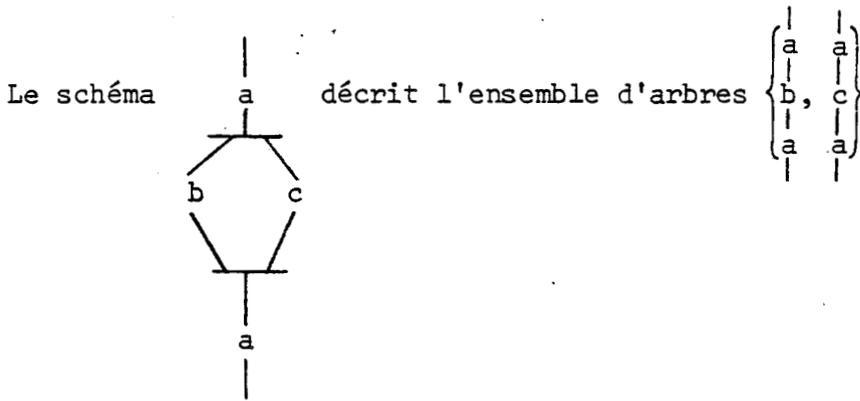
Notation :

Dans les schémas qui suivent le symbole $\overline{\dots}$ aura la signification d'un choix entre diverses possibilités ainsi le schéma



décrit l'ensemble d'arbres : $\left\{ \begin{array}{c} | \\ a \\ / \quad \backslash \\ b \quad c \end{array} , \begin{array}{c} | \\ a \\ / \quad \backslash \\ b \quad b \end{array} , \begin{array}{c} | \\ a \\ / \quad \backslash \\ c \quad b \end{array} , \begin{array}{c} | \\ a \\ / \quad \backslash \\ c \quad c \end{array} \right\}$

où de façon symétrique :



Dans les forêts régulières (ou reconnaissables) d'arbres, le problème ne peut être résolu aussi facilement. En effet, le passage en dimension deux induit une difficulté de description supplémentaire : la localisation des sous-arbres à itérer.

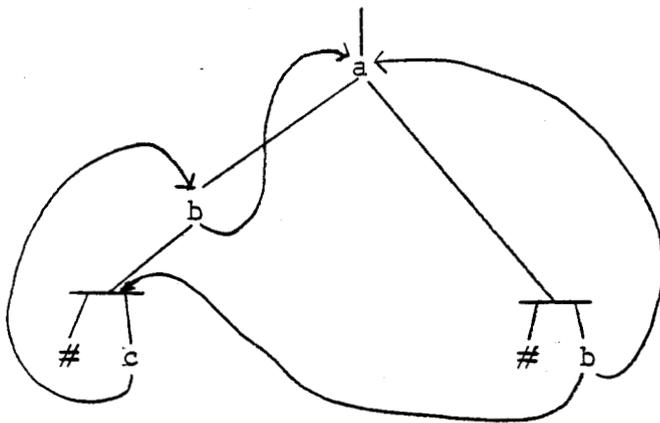
Exemple :

Soit l'automate d'arbres suivant $M = (\Sigma, Q, F, R)$

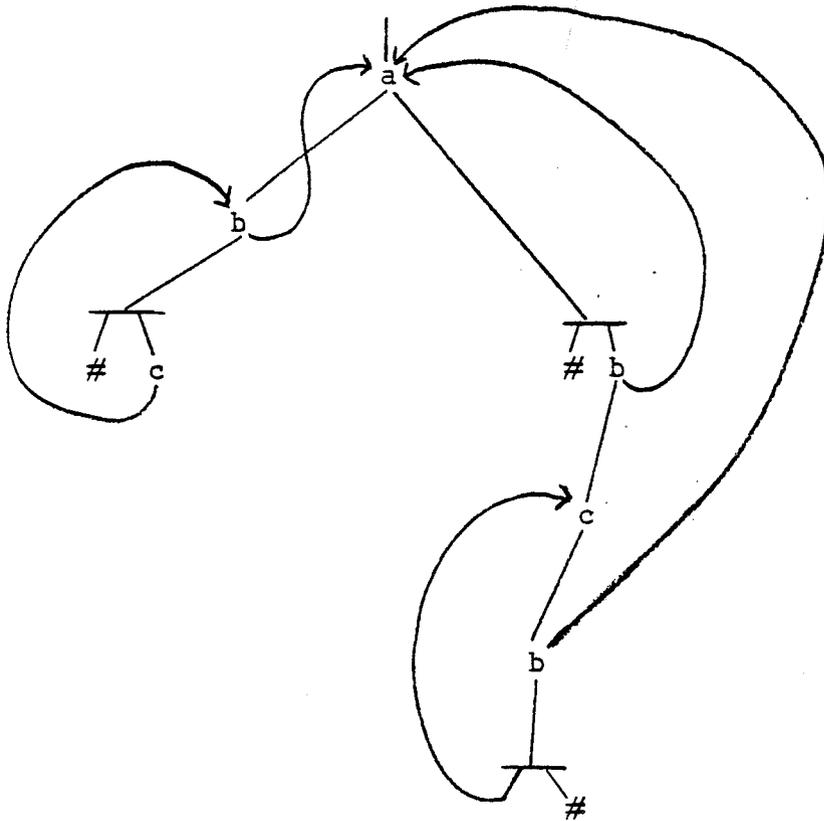
$\Sigma = \{a, b, c, \#\}$ $Q = \{q_0, q_1, q_2, q_3\}$ $F = \{q_0\}$

$R = \{q_0 \cdot a \vdash a \cdot (q_1, q_2) ; a_1 \cdot b \vdash b \cdot (q_3, q_0) ; q_3 \cdot c \vdash c \cdot q_1 ;$
 $q_2 \cdot b \vdash b \cdot (q_3, q_0) ; q_3 \cdot \# \vdash \#, q_2 \cdot \# \vdash \# \}.$

La forêt reconnue par cet automate peut être décrite en dimension deux par :



où encore, en développant ce schéma, de façon à ce que chaque recyclage se fasse en remontant sur un noeud père :

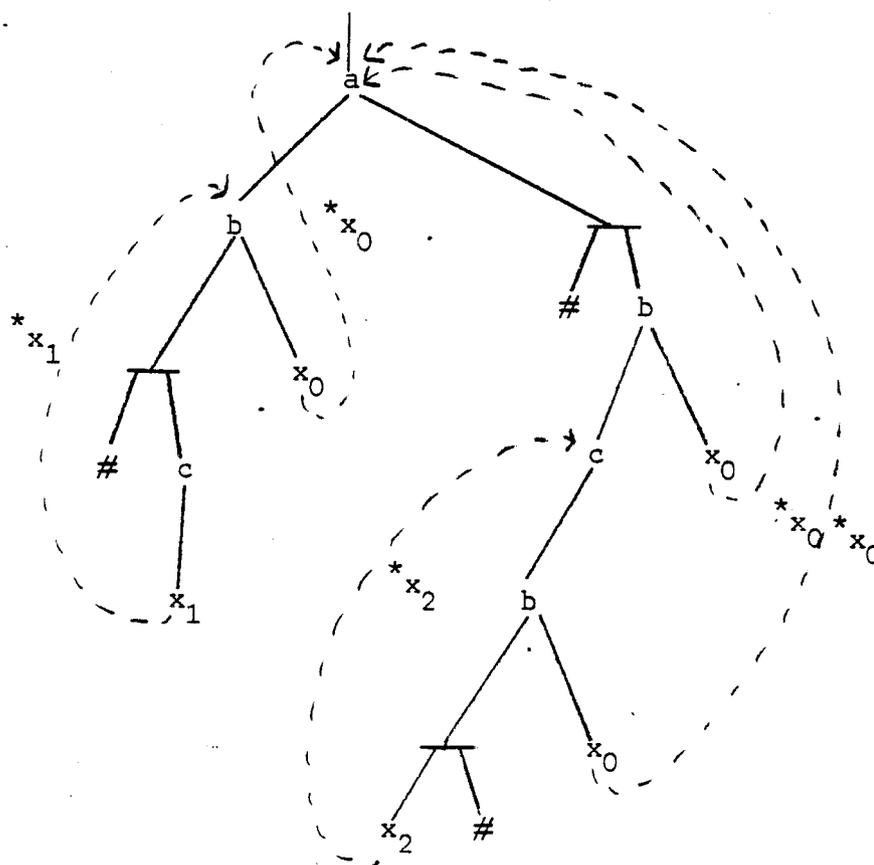


On remarque que trois sous-arbres font l'objet d'une itération, et que ces itérations s'amorcent à partir de différents endroits, d'où l'idée pour traduire ce schéma en notation linéaire de marquer certaines places de l'arbre, marquages qui vont de pair avec des opérateurs d'itérations relatifs à ces marquages.

Ainsi la forêt décrite précédemment pourrait s'écrire :

$$[a \cdot ([b \cdot (c \cdot x_1 + \#, x_0)]^{*x_1}, b \cdot ([c \cdot b(x_2 + \#, x_0)]^{*x_2}, x_0) + \#)]^{*x_0}$$

Ce qui correspond au marquage suivant :



L'opérateur $*_{x_0}$ (respectivement $*_{x_1}$, $*_{x_2}$) équivaut à itérer à partir des places marquées x_0 (respectivement x_1 , x_2).

Remarque 1 :

Ce marquage de certaines arités des lettres intervenant dans une expression rationnelle d'arbres est en quelque sorte un "typage" de ces lettres.

Remarque 2 :

Ces expressions rationnelles d'arbres sont les solutions calculables de systèmes d'équations réguliers.

c) Dans les dags

L'introduction de lettres bigraduées doit donc inévitablement conduire à la fois à un marquage des arités, mais aussi des co-arités de certaines lettres. Il en ressort que les expressions rationnelles de dags seront construites à partir des lettres d'un alphabet bitypé.

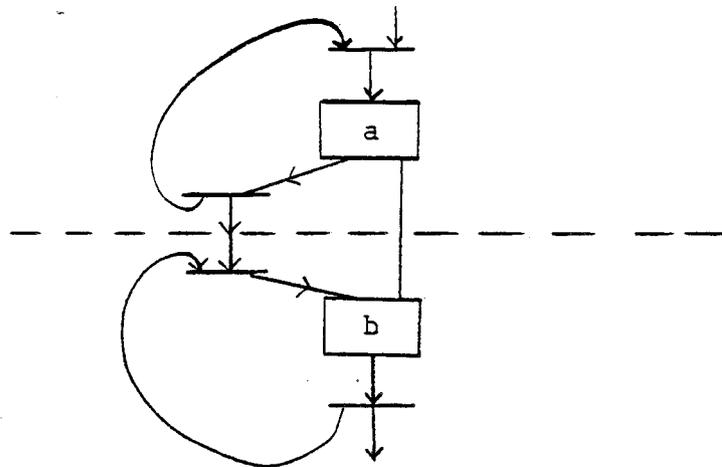
Exemple :

Soit $M = (\Sigma, Q, I, F, R)$ un automate de dag, où

- $\Sigma = \Sigma_2^1 \cup \Sigma_1^2$ et $\Sigma_2^1 = \{a\}$ et $\Sigma_1^2 = \{b\}$
- $Q = \{q_0, q_1, q_2\}$
- $I = \{q_0\}$ $F = \{q_2\}$
- $R = \left\{ \begin{array}{l} q_0 \cdot a \mid a \cdot (q_0 \otimes q_1) ; (q_2 \otimes q_1) \cdot b \mid b \cdot q_2 \\ q_0 \cdot a \mid a \cdot (q_2 \otimes q_1) \end{array} \right\}$

Le taillis reconnu par cet automate est celui donné à titre d'exemple au paragraphe V.4.1.2. c).

On peut associer à ce taillis une représentation finie en dimension deux.



Cette description ne fait plus réellement apparaître de sous-graphes pouvant être itérés mais exprime plutôt un certain nombre de contraintes quant aux assemblages des lettres. Ces contraintes peuvent être parfaitement traduites par des correspondances de types à l'intérieur d'un magmaïde typé.

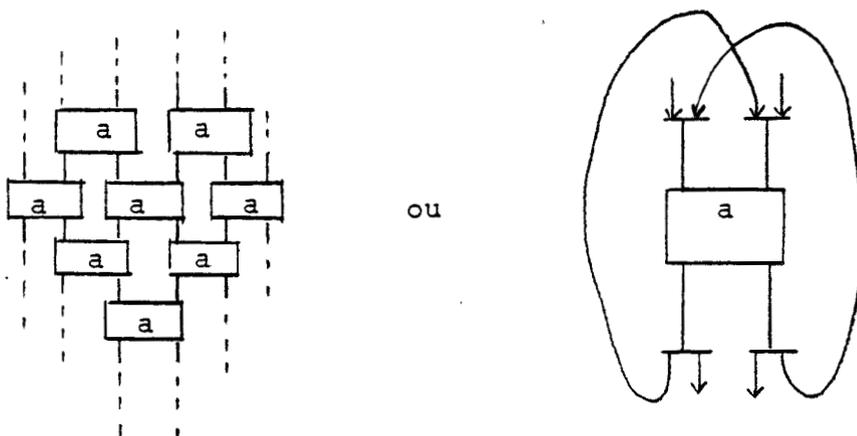
Ainsi le taillis précédent peut s'écrire :

$$(q_0, ((a_{q_0 q_1}^{q_0})^{\odot})^* \cdot ((b_{q_0}^{q_0 q_1})^{\odot})^*, q_0)$$

Signalons tout de même la perte de perception graphique du taillis régulier lorsqu'on le caractérise par une expression rationnelle.

Ainsi dans l'exemple suivant :

l'expression rationnelle : $((q_0 q_1)^*, ((a_{q_1 q_0}^{q_0 q_1})^{\odot})^*, (q_0 + q_1)^*)$ s'interprète comme le taillis des dags de la forme :



VI.5.2. - Le choix de l'opérateur de mise en parallèle

L'extension canonique de la structure de magmaïde aux parties amène à considérer les opérateurs produit tensoriel (\otimes) et produit tensoriel itéré (X^{\otimes}), dont les interprétations seraient les suivantes :

$\forall \text{exp1} \in \text{Ex} \quad \forall \text{exp2} \in \text{Ex}$

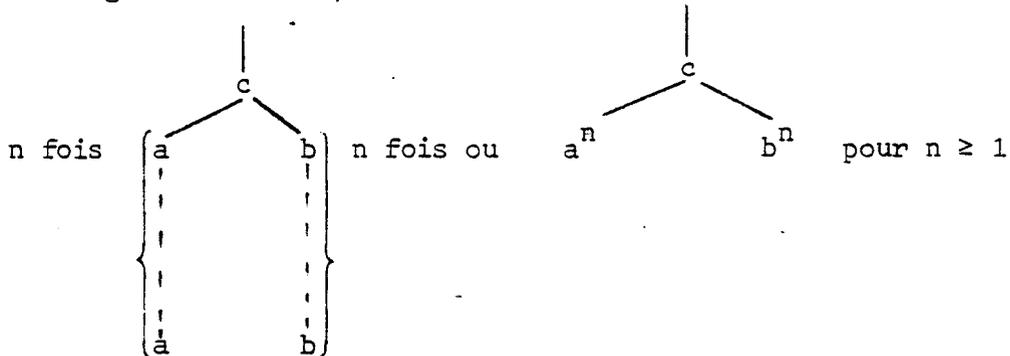
$$I(\text{exp1} \otimes \text{exp2}) = I(\text{exp1}) \otimes I(\text{exp2})$$

$\forall \text{exp} \in \text{Ex}$

$$I(\text{exp}^{\otimes}) = \bigcup_{i \in \mathbb{N}^*} I(\underbrace{\text{exp} \otimes \dots \otimes \text{exp}}_{i \text{ fois}}) \cup \{e_p / p \in T^*\}$$

Or de telles définitions vont à l'encontre du but fixé (REC = RAT = REG) puisque la clôture par de tels opérateurs des sous-ensembles finis de dags nous entraîne hors de la classe des taillis reconnaissables.

En effet, on aurait alors, pour $(\Sigma, T) = \left\{ \begin{matrix} t_1 & t_1 & t_1 \\ a_{t_1} & b_{t_1} & c_{t_1 t_1} \end{matrix} \right\}$, comme interprétation de l'expression rationnelle $(t_1, c_{t_1 t_1}^{t_1} \cdot (a_{t_1}^{t_1} \otimes b_{t_1}^{t_1})^*, t_1 t_1)$ le taillis composé de dags de la forme :



Et ce taillis n'est pas un taillis reconnaissable.

La raison fondamentale est que l'opérateur \otimes s'interprète comme une mise en parallèle correspondant à des mouvements de reconnaissance synchrone le long de deux branches indépendantes. Ce type de reconnaissance ne peut être assuré par un automate d'états finis qui ne considère que le contexte immédiat d'une lettre (les états qui sont composés avec elle).

L'interprétation de l'opérateur \otimes est définie, quant à elle, de sorte à simuler des mouvements de reconnaissances asynchrones en parallèles : on respecte le contexte (les types), la mise en parallèle (l'ordre horizontal) mais les mouvements sont indépendants. Ce comportement reproduit celui des automates d'états finis considérés usuellement dans les arbres et par extension ici dans les dags.

Remarque :

Une autre manière d'aborder cette question aurait été de remettre en cause notre notion de reconnaissabilité, et en particulier d'enrichir la gamme des mouvements de reconnaissance possibles. Pour ce faire, on pourrait imaginer l'existence de règles permettant de reconnaître un parallélisme synchrone local, ces règles seraient de la forme :

Exemple :

$$(t_1 \otimes t_1) \cdot (a \otimes b) \vdash (a \otimes b) \cdot (t_1 \otimes t_1)$$

Sous ces conditions, le taillis $\left\{ \begin{array}{c} c \\ / \quad \backslash \\ a^n \quad b^n \end{array} \right. / n \in \mathbb{N}^* \}$ devient reconnaissable

En introduisant la possibilité d'un contrôle de synchronisation local, on peut par intersection obtenir une synchronisation sur une largeur quelconque.

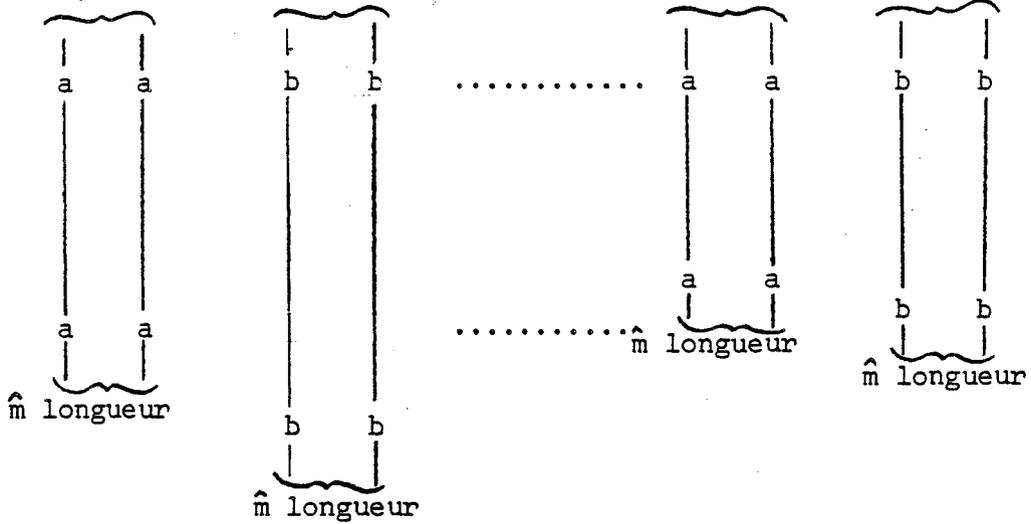
Exemple :

$$(\Sigma, T) = \left\{ \begin{array}{cc} t_0 & t_1 \\ a_{t_0} & b_{t_1} \end{array} \right\}, \text{ ex1 et ex2 deux expressions rationnelles avec}$$

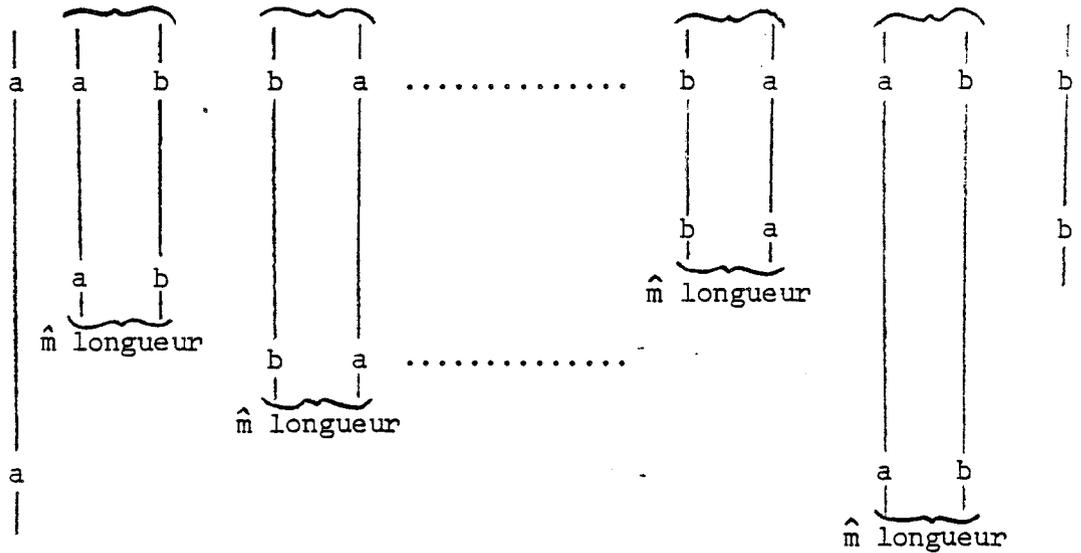
$$\text{ex1} = ((t_0^2 \cdot t_1^2)^*, [[(a_{t_0}^{t_0} \otimes a_{t_0}^{t_0}) + (b_{t_1}^{t_1} \otimes b_{t_1}^{t_1})]^\otimes]^*, (t_0^2 \cdot t_1^2)^*)$$

$$\text{ex2} = ((t_0^2 \cdot t_1^2)^*, [a_{t_0}^{t_0} \otimes ((a_{t_0}^{t_0} \otimes b_{t_1}^{t_1}) + (b_{t_1}^{t_1} \otimes a_{t_0}^{t_0}))^\otimes \otimes b_{t_1}^{t_1}]^*, (t_0^2 \cdot t_1^2)^*)$$

- les dags de l'interprétation de ex1 sont de la forme :



- les dags de l'interprétation de ex2 sont de la forme :



Par intersection de ces deux taillis on obtient une famille de dags composés alternativement dans le sens de la largeur de chaînes de "a" et de chaînes de "b" qui ont toutes la même longueur.

On a $T(M)$ égal au taillis décrit ci-dessus (cf. exemple du mur dans le premier chapitre).

On peut émettre la conjecture suivante : la classe des taillis reconnus par ce type d'automate d'états finis est close par intersection.

CHAPITRE VII

LE THEOREME RAT = REG

PLAN DU CHAPITRE VII

VII.1. - ENONCE DU THEOREME

VII.2. - DEMONSTRATION DU THEOREME

VII.2.1. - Enoncé du lemme

VII.2.2. - Démonstration du lemme par induction

VII.2.3. - Démonstration de l'inclusion de RAT dans I

VII.2.4. - Démonstration de l'inclusion de REC dans I

VII.3. - CONCLUSION

VII. - LE THEOREME RAT = REC ET SA DEMONSTRATION

VII.1. - ENONCE DU THEOREME

Quelque soit T taillis reconnaissable, il est également l'interprétation d'une expression rationnelle de dags.

Quelque soit r expression rationnelle, son interprétation est un taillis reconnaissable.

VII.2. - DEMONSTRATION DU THEOREME

Cette démonstration sera menée en deux temps :

1. Démonstration de $RAT \subset REC$
2. Démonstration de $REC \subset RAT$

La démonstration de $RAT \subset REC$, s'appuie sur la démonstration d'un lemme fort important, qui requiert une preuve par induction sur la construction des expressions simples de dags.

VII.2.1. - Enoncé du Lemme

$\forall exp \in Ex$ (expression simple de dag sur (Σ, T)).

$\exists M = \langle \Sigma, Q, I, F, R \rangle$ automate de dag planaire qui vérifie les trois propriétés suivantes.

Propriété 1 :

Q est un produit cartésien d'ensembles dans lequel figure une fois T et $\forall a \in \Sigma \quad \forall m, m' \in Q^* \quad m \cdot a \vdash a \cdot m' \in R \Rightarrow (\Pi_T(m), a, \Pi_T(m')) \in (\Sigma, T)$.

Cette propriété exprime la compatibilité des règles de R avec le typage des lettres de (Σ, T) .

Cette propriété 1 permet la construction du système de réécriture τ défini par :

- à toute règle de $R : m \cdot a \vdash a \cdot m'$, on associe la règle de réécriture $m \cdot a \xrightarrow{\tau} (\Pi_T(m), a, \Pi_T(m')) \cdot m'$ dans τ .
- aux pseudo-règles $m \cdot e \vdash e \cdot m$, on associe la règle de réécriture $m \cdot e \xrightarrow{\tau} e_{\Pi_T(m)} \cdot m$.
- A tout mouvement dans M , on peut donc associer une réécriture dans τ telle que

$$\forall \delta \in \hat{DP}(\Sigma) \quad \forall m, m' \in Q^*$$

$$m \cdot \delta \xrightarrow[M]{*} \delta \cdot m' \Rightarrow m \cdot \delta \xrightarrow{\tau} \tau(\delta) \cdot m'$$

L'élément de $\hat{DP}(\Sigma, T)$ résultant de cette réécriture dans τ , sera appelé $\tau(\delta)$.

Remarque :

On rappelle que $\bar{\tau}$ désigne la fonction de détypage de $\hat{DP}(\Sigma, T)$ dans $\hat{DP}(\Sigma)$.

$\bar{\tau}(\tau(\delta)) = \delta$; τ peut donc être assimiler à un "typage" d'un élément δ de $\hat{DP}(\Sigma)$ relativement à un mouvement dans M .

Exemple :

Soit $M = \langle \Sigma, Q \times T, I, F, R \rangle$ où

- $\Sigma = \{a, b\}$
- $Q = \{i, f, m\}$ et $T = \{t_1, t_2\}$
- $I = (i, t_1)$ et $F = (f, t_1)$
- $R = \{(i, t_1) \cdot a \vdash a \cdot (m, t_2) ; (m, t_2) \cdot b \vdash b \cdot (f, t_1)\}$

Notations :

Soit L un langage rationnel de mots inclus dans X^* , on appellera :

$$FG(L) = \{u \in X^+ / \exists v \in X^+ \ uv \in L\}$$

ensemble des facteurs gauches propres de L

$$FD(L) = \{v \in X^+ / \exists u \in X^+ \ uv \in L\}$$

ensemble des facteurs droits propres de L.

$$FS(L) = \{w \in X^+ / \exists u, v \in X^+ \ uwv \in L\}$$

ensemble des facteurs stricts de L.

Propriété 3 :

$\forall \delta \in \overset{\vee}{DP}(\Sigma)$, δ non élément neutre.

$$1) \cdot \exists m_i \in FG(I) \text{ et } m_f \in FG(F) \text{ tels que } m_i \cdot \delta \stackrel{*}{M} \delta \cdot m_f \Rightarrow$$

$$\forall t' \in T^* \quad \tau(\delta) \otimes e_{t'} \in I(\text{exp}).$$

$$2) \cdot \exists m_i \in FD(I) \text{ et } m_f \in FD(F) \text{ tels que } m_i \cdot \delta \stackrel{*}{M} \delta \cdot m_f \Rightarrow$$

$$\forall t \in T^* \quad e_t \otimes \tau(\delta) \in I(\text{exp}).$$

$$3) \cdot \exists m_i \in FS(I) \text{ et } m_f \in FS(F) \text{ tels que } m_i \cdot \delta \stackrel{*}{M} \delta \cdot m_f \Rightarrow$$

$$\forall t, t' \in T^* \quad e_t \otimes \tau(\delta) \otimes e_{t'} \in I(\text{exp}).$$

Commentaire :

On s'impose cette propriété 3 pour rendre compte de certaines particularités de l'interprétation des expressions simples de dags, particularités que nous illustrerons par un exemple :

$$\text{Soit } \text{exp} = a_{t_1}^{t_1} \otimes b_{t_2}^{t_2} \otimes c_{t_3}^{t_3} \in \text{Ex} ; \text{ on a alors}$$

$$\begin{aligned} \forall t \in T^* & \quad a_{t_1}^{t_1} \otimes e_t \in I(\text{exp}) \\ \forall t \in T^* & \quad e_t \otimes c_{t_3}^{t_3} \in I(\text{exp}) \\ \forall t, t' \in T^* & \quad e_t \otimes b_{t_2}^{t_2} \otimes e_{t'} \in I(\text{exp}) \end{aligned}$$

VII.2.2. - Démonstration de lemme par induction

a) exp est réduit à une lettre de (Σ, T)

Soit $\text{exp} = a_{t_1 t_2 \dots t_n}^{t_1 t_2 \dots t_n}$; lettre de (Σ, T) .

On peut effectivement construire un automate qui vérifie les propriétés 1, 2 et 3 :

L'automate $M = \langle \Sigma, Q \times T, I, F, R \rangle$ où

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

$$I = (q_0, t_1) \otimes (q_0, t_2) \otimes \dots \otimes (q_0, t_n)$$

$$F = (q_1, t'_1) \otimes (q_1, t'_2) \otimes \dots \otimes (q_1, t'_m)$$

$$R = \{I \cdot a \mid a \cdot F\}$$

• M vérifie sans conteste la propriété 1

• $T(M) = a$; la seule manière de reconnaître a est : $I \cdot a \mid_M a \cdot F$

$$\bullet I \cdot a \mid_M a \cdot F \Rightarrow \tau(a) = a_{t'_1 \dots t'_m}^{t_1 \dots t_n} \in I(a_{t'_1 \dots t'_m}^{t_1 \dots t_n})$$

la propriété 2 est donc trivialement vérifiée.

• La vérification de la propriété 3 ne se pose pas, il est impossible de se placer dans des conditions telles que les prémisses des implications soient vraies.

b) exp est la somme de deux expressions simples de dags

Soit $\text{exp} = \text{exp1} + \text{exp2}$.

D'après notre hypothèse d'induction :

$\exists M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ qui vérifie les propriétés 1, 2, 3 pour exp1

$\exists M_2 = \langle \Sigma, Q_2, I_2, F_2, R_2 \rangle$ qui vérifie les propriétés 1, 2, 3 pour exp2

de plus, on peut s'assurer que $Q_1 \cap Q_2 = \emptyset$.

Soit l'automate $M = \langle \Sigma, Q_1 \cup Q_2, I_1 \cup I_2, F_1 \cup F_2, R_1 \cup R_2 \rangle$.

On peut énoncer directement, comme conséquence de la séparation des ensembles d'états Q_1 et Q_2 que M vérifie la propriété suivante :

$$\forall \delta \in \overset{\vee}{\text{DP}}(\Sigma) \quad \forall m_1, m_2 \in ((Q_1 \cup Q_2) \times T)^*$$

$$m_1 \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta \cdot m_2 \iff \begin{cases} m_1 \cdot \delta \stackrel{*}{\underset{M_1}{\mid}} \delta \cdot m_2 \\ \text{ou} \\ m_1 \cdot \delta \stackrel{*}{\underset{M_2}{\mid}} \delta \cdot m_2 \end{cases}$$

De cette propriété résulte la transmission des propriétés 1, 2, 3 à l'automate M .

c) exp est la mise en parallèle asynchrone de deux expressions simples de dags

Soit $\text{exp} = \text{exp1} \textcircled{\vee} \text{exp2}$.

D'après notre hypothèse d'induction :

$\exists M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ qui vérifie les propriétés 1, 2, 3 pour exp1

$\exists M_2 = \langle \Sigma, Q_2, I_2, F_2, R_2 \rangle$ qui vérifie les propriétés 1, 2, 3 pour exp2

de plus, on peut s'assurer que $Q_1 \cap Q_2 = \emptyset$.

Soit l'automate $M = \langle \Sigma, Q, I, F, R \rangle$ où

$$Q = Q_1 \cup Q_2 \cup (\{\Omega\} \times T); \quad \Omega \text{ étant un nouvel état}$$

$$I = (I_1 \otimes I_2) \cup (I_1 \otimes (\{\Omega\} \times T)^*) \cup ((\{\Omega\} \times T)^* \otimes I_2)$$

$$F = (F_1 \otimes F_2) \cup (F_1 \otimes (\{\Omega\} \times T)^*) \cup ((\{\Omega\} \times T)^* \otimes F_2)$$

$$R = R_1 \cup R_2$$

La propriété 1 est vraie pour les ensembles de règles R_1 et R_2 , elle est donc vraie pour $R_1 \cup R_2$; l'automate M vérifie donc la propriété 1

M vérifie également la propriété 2.

Preuve :

Le fait que $Q_1 \cap Q_2 = \emptyset$ et que Ω soit un nouvel état, confère à M la propriété suivante :

$$\forall \delta \in \text{DP}(\Sigma, T) \quad \forall m_i \in I \quad \forall m_f \in F$$

$$m_i \cdot \delta \stackrel{*}{\underset{M}{\mid}} \delta \cdot m_f \Rightarrow$$

On distingue trois types de reconnaissances :

1^{er} cas :

$$m_i = m_{i1} \otimes m_{i2}; \quad m_{i1} \in I_1 \text{ et } m_{i2} \in I_2$$

$$m_f = m_{f1} \otimes m_{f2}; \quad m_{f1} \in F_1 \text{ et } m_{f2} \in F_2$$

$$\delta = \delta_1 \otimes \delta_2$$

$$\text{et } m_{i1} \cdot \delta_1 \stackrel{*}{\underset{M_1}{\mid}} \delta_1 \cdot m_{f1} \quad \text{et } m_{i2} \cdot \delta_2 \stackrel{*}{\underset{M_2}{\mid}} \delta_2 \cdot m_{f2}$$

⇒ comme M_1 et M_2 vérifie la propriété 2

$$\tau(\delta_1) \in I(\text{exp1}) \text{ et } \tau(\delta_2) \in I(\text{exp2})$$

$$\Rightarrow \tau(\delta) = \tau(\delta_1) \otimes \tau(\delta_2) \in I(\text{exp1} \oplus \text{exp2})$$

□

2^{ème} cas :

$$m_i = m_{i_1} \otimes w ; m_{i_1} \in I_1 \text{ et } w \in (\{\Omega\} \times T)^*$$

$$m_f = m_{f_1} \otimes w ; m_{f_1} \in F_1$$

$$\delta = \delta_1 \otimes e_{\ell(w)} ; \text{ où } \ell(w) \text{ désigne le nombre de lettre de } w$$

$$\text{et } m_{i_1} \cdot \delta_1 \stackrel{*}{M_1} \delta_1 \cdot m_{f_1}$$

⇒ comme M_1 vérifie la propriété 2

$$\tau(\delta_1) \in I(\text{exp1})$$

$$\Rightarrow \tau(\delta) = \tau(\delta_1) \otimes e_{\Pi_T(w)} \in I(\text{exp1} \oplus \text{exp2})$$

□

3^{ème} cas :

$$m_i = w \otimes m_{i_2} ; w \in (\{\Omega\} \times T)^* \text{ et } m_{i_2} \in I_2$$

$$m_f = w \otimes m_{f_2} ; m_{f_2} \in F_2$$

$$\delta = e_{\ell(w)} \otimes \delta_2$$

$$\text{et } m_{i_2} \cdot \delta_2 \stackrel{*}{M_2} \delta_2 \cdot m_{f_2}$$

⇒ comme M_2 vérifie la propriété 2

$$\tau(\delta_2) \in I(\text{exp2})$$

$$\Rightarrow \tau(\delta) = e_{\Pi_T(w)} \otimes \tau(\delta_2) \in I(\text{exp1} \oplus \text{exp2})$$

□

Ceci conclut la preuve que M vérifie bien le premier point de la propriété

Point deux de la propriété 2 :

$$\bullet \forall \delta' \in I(\text{exp1} \otimes \text{exp2})$$

$$\delta' \in I(\text{exp1}) \otimes I(\text{exp2}) \text{ ou } \delta' \in I(\text{exp1}) \otimes \{e_p / p \in T^*\} \text{ ou } \delta' \in \{e_p / p \in T^*\} \otimes I(\text{exp2})$$

envisageons tour à tour les trois éventualités.

1^{er} cas :

$$\delta' \in I(\text{exp1}) \otimes I(\text{exp2}) \text{ d'où } \delta' = \delta'_1 \otimes \delta'_2 ; \delta'_1 \in I(\text{exp1}) \text{ et } \delta'_2 \in I(\text{exp2})$$

$$- \delta'_1 \in I(\text{exp1}) \Rightarrow \exists \delta_1 \in T(M_1) \text{ tel que } \tau(\delta_1) = \delta'_1$$

car M_1 vérifie la propriété 2

$$- \delta'_2 \in I(\text{exp2}) \Rightarrow \exists \delta_2 \in T(M_2) \text{ tel que } \tau(\delta_2) = \delta'_2$$

car M_2 vérifie la propriété 2.

$$\text{Il en résulte que } \delta_1 \otimes \delta_2 \in T(M) \text{ et que } \tau(\delta_1 \otimes \delta_2) = \delta'_1 \otimes \delta'_2 = \delta'$$

□

2^{ème} cas :

$$\delta' \in I(\text{exp1}) \otimes \{e_p / p \in T^*\} \text{ d'où } \delta' = \delta'_1 \otimes e_{t_1 t_2 \dots t_n} \text{ avec } \delta'_1 \in I(\text{exp1})$$

$$- \delta'_1 \in I(\text{exp1}) \Rightarrow \exists \delta_1 \in T(M_1) \text{ tel que } \tau(\delta_1) = \delta'_1$$

car M_1 vérifie la propriété 2.

$$- \delta_1 \in T(M_1) \Rightarrow \exists m_i \in I \text{ et } m_f \in F \text{ tels que } m_i \cdot \delta_1 \stackrel{*}{|}_{M_1} \delta_1 \cdot m_f$$

Le mouvement suivant est alors valide dans M

$$\begin{aligned} & (m_i \otimes (\Omega, t_1) \otimes (\Omega, t_2) \dots \otimes (\Omega, t_n)) \cdot (\delta_1 \otimes e_n) \Big|_M^* (\delta_1 \otimes e_n) \cdot \\ & (m_f \otimes (\Omega, t_1) \otimes \dots \otimes (\Omega, t_n)) \end{aligned}$$

Ce qui implique que :

$$\delta_1 \otimes e_n \in T(M) \text{ et } \tau(\delta_1 \otimes e_n) = \delta'_1 \otimes e_{t_1 \dots t_n} = \delta'$$

□

3^{ème} cas :

$$\delta' \in \{ep / p \in T^*\} \otimes I(\exp 2)$$

Ce cas se traite de façon tout à fait symétrique au précédent.

Ceci achève la preuve que M vérifie la propriété 2.

□

M vérifie également la propriété 3.

Preuve :

Nous axerons notre démonstration uniquement sur le point 3 de cette propriété, les deux autres points se démontrant de façon similaire.

Soit $\delta \in \overset{\vee}{DP}(\Sigma)$, $m_i \in FS(I)$, et $m_f \in FS(F)$ tels que

$$m_i \cdot \delta \Big|_M^* \delta \cdot m_f$$

Nous allons examiner les différents modèles de facteurs stricts de I :

1) Si $m_i \in FS(I_1)$ obligatoirement $m_f \in FS(F_1)$

Les règles utilisées pour reconnaître δ appartiennent donc à R_1 .

$$\left. \begin{array}{l} \bullet m_i \cdot \delta \Big|_{M_1}^* \delta \cdot m_f \\ M \text{ vérifie la propriété 3} \end{array} \right\} \Rightarrow \forall t, t' \in T^* \quad e_t \otimes \tau(\delta) \otimes e_{t'} \in I(\text{exp1})$$

$$\bullet I(\text{exp1}) \subset I(\text{exp1} \otimes \text{exp2})$$

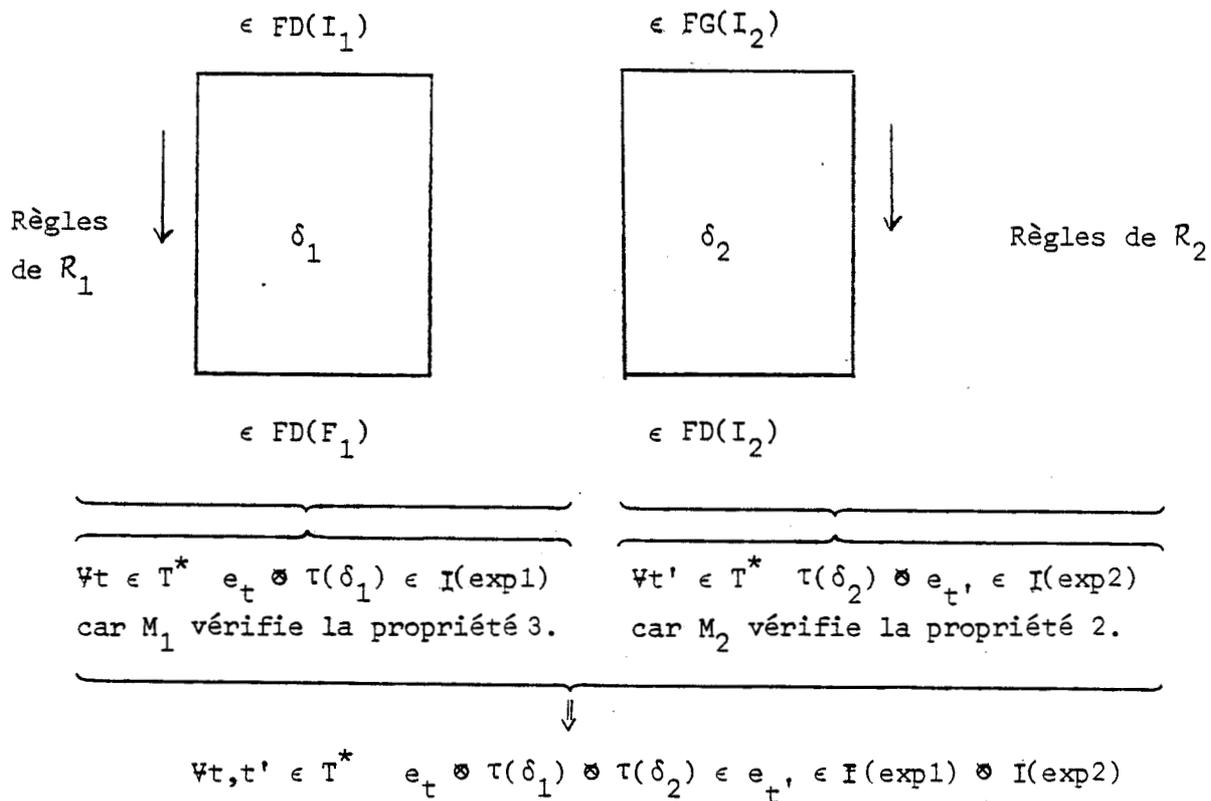
$$\text{d'où } \forall t, t' \in T^* \quad e_t \otimes \tau(\delta) \otimes e_{t'} \in I(\text{exp})$$

□

2) Si $m_i \in \text{FS}(I_2)$.. cas analogue au précédent.

3) Si $m_i \in \text{FD}(I_1) \otimes \text{FG}(I_2)$ obligatoirement $m_f \in \text{FD}(F_1) \otimes \text{FG}(F_2)$

$m_i \cdot \delta \Big|_M^* \delta \cdot m_f$ peut se décomposer suivant le schéma suivant :



□

4) Si $m_i \in (\{\Omega\} \times T)^* \otimes FG(I_2)$, obligatoirement $m_f \in (\{\Omega\} \times T)^* \otimes FG(F_2)$

d'où : $\delta = e_n \otimes \delta_2$; $m_i = w \otimes m_{i_2}$; $m_f = w \otimes m_{f_2}$ avec

$w \in (\{\Omega\} \times T)^*$; $m_{i_2} \in FG(I_2)$; $m_{f_2} \in FG(F_2)$

et $m_{i_2} \cdot \delta_2 \left| \frac{*}{M_2} \right. \delta_2 \cdot m_{f_2}$

$\left. \begin{array}{l} \cdot m_{i_2} \cdot \delta_2 \left| \frac{*}{M_2} \right. \delta_2 \cdot m_{f_2} \\ M_2 \text{ vérifie la propriété 3} \end{array} \right\} \Rightarrow \forall t \in T^* \quad \tau(\delta_2) \otimes e_t \in I(\text{exp2})$

$\cdot \forall \delta' \in I(\text{exp2}) \quad \forall t' \in T^* \quad e_{t'} \otimes \delta' \in I(\text{exp1} \otimes \text{exp2})$

$\cdot \tau(\delta) = e_{\pi_T(w)} \otimes \tau(\delta_2)$

On en conclut que

$\forall t, t' \in T^* \quad e_{t'} \otimes \tau(\delta) \otimes e_t \in I(\text{exp1} \textcircled{\sim} \text{exp2})$

□

5) Si $m_i \in FD(I_1) \otimes (\{\Omega\} \times T)^*$... cas analogue au précédent.

Nous avons étudié tous les cas possibles, M vérifie donc le point 3 de la propriété 3.

□

d) exp est la mise en parallèle asynchrone itéré d'une expression simple de dag

Soit $\text{exp} = (\text{exp1}) \textcircled{\sim}$

D'après notre hypothèse d'induction :

$-M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ qui vérifie les propriétés 1, 2, 3 pour exp1.

Soit $L = \{bg, m, hd, bdg\}$, ensemble de nouveaux symboles.

Soit $L = bg \otimes m^* \otimes hd + bgd$.

Soit Ω , un nouvel état.

Soit l'automate $M = \langle \Sigma, Q, I, F, R \rangle$ où

$$Q = (Q_1 \times L) \cup (\{\Omega\} \times T)$$

$$I = (I' + \{\Omega\} \times T)^* \text{ dans lequel } I' = \{m \in (Q_1 \times L)^* / \Pi_L(m) \in L \text{ et } \Pi_{Q_1}(m) \in I_1\}$$

$$F = (F' + \{\Omega\} \times T)^* \text{ dans lequel } F' = \{m \in (Q_1 \times L)^* / \Pi_L(m) \in L \text{ et } \Pi_{Q_1}(m) \in F_1\}$$

$$R = \{m \cdot a \mid a \cdot m' / \Pi_{Q_1}(m) \cdot a \mid a \cdot \Pi_{Q_1}(m') \in R_1$$

et $\Pi_L(m)$ et $\Pi_L(m') \in m^* \otimes bd$ ou

$\Pi_L(m)$ et $\Pi_L(m') \in bg \otimes m^*$ ou

$\Pi_L(m)$ et $\Pi_L(m') \in m^*$ ou

$\Pi_L(m)$ et $\Pi_L(m') \in L\}$.

On a alors a :

M vérifié la propriété 1, en effet la construction des règles de R conserve les projections sur T .

M vérifie la propriété 2.

Preuve du premier point de la propriété 2 :

Cette preuve est basée sur les deux remarques suivantes :

• par M , à partir d'un mot de I' , on ne peut reconnaître qu'un dag appartenant à $T(M_1)$. En effet, tout mot de I' est un mot de I marqué à droite par bd et à gauche par bg (ou bgd si le mot est de longueur 1). Le transport de ces marques par l'intermédiaire des règles de R assure l'indépendance entre les reconnaissances menées à partir du mot de I' , et celles issues de son contexte.

• Dans aucune des règles de R n'apparaît d'état de la forme (Ω, t) ; sous un état appartenant à $\{\Omega\} \times T$, on ne reconnaît que des éléments neutres.

$$\forall \delta \in \overset{\vee}{DP}(\Sigma) \quad \forall m_i \in I \quad \forall m_f \in F$$

$$m_i \cdot \delta \Big|_M \delta \cdot m_f \Rightarrow \tau(\delta) \in (I(\text{exp}_1) + \{e_p / p \in T^*\})^* \Rightarrow \tau(\delta) \in I(\text{exp}_1^{\circledast})$$

□

Preuve du deuxième point de la propriété 2 :

$$\forall \delta' \in I(\text{exp}_1^{\circledast}) :$$

δ' s'écrit comme un produit tensoriel de dags appartenant alternativement à $\{e_p / p \in T^*\}$ et à $I(\text{exp}_1)$.

$$\delta' = e_{t_0} \otimes \delta'_0 \otimes e_{t_1} \otimes \delta'_1 \dots \otimes e_{t_i} \otimes \delta'_i \otimes \dots \otimes \delta'_k \otimes e_{t_{k+1}}$$

avec pour tout i , $t_i \in T^*$ et $\delta'_i \in I(\text{exp}_1)$.

- M_1 vérifie la propriété 2 } \Rightarrow pour tout i $\exists \delta_i \in T(M_1)$ tel que $\tau(\delta_i) = \delta'_i$.
- pour tout i , $\delta'_i \in I(\text{exp}_1)$ }
- $\delta_i \in T(M_1) \Rightarrow \delta_i \in T(M)$; et de plus δ_i est reconnu exactement de la même façon dans M et M_1 .
- $\forall t \in T^* \quad \exists e_n \in T(M)$ tel que $\tau(e_n) = e_t$.

On conclut des trois points précédents que

$$\exists \delta = e_{n_0} \otimes \delta_0 \otimes e_{n_1} \otimes \delta_1 \dots \otimes e_{n_i} \otimes \delta_i \dots \otimes \delta_k \otimes e_{n_{k+1}} \in T(M) \text{ tel que } \tau(\delta) = \delta'.$$

M vérifie la propriété 3.

Preuve :

Cette preuve se ramenant à une analyse de cas, nous n'examinerons que certains cas qui nous semblent suffisamment représentatifs des autres pour nous dispenser de leur démonstration.

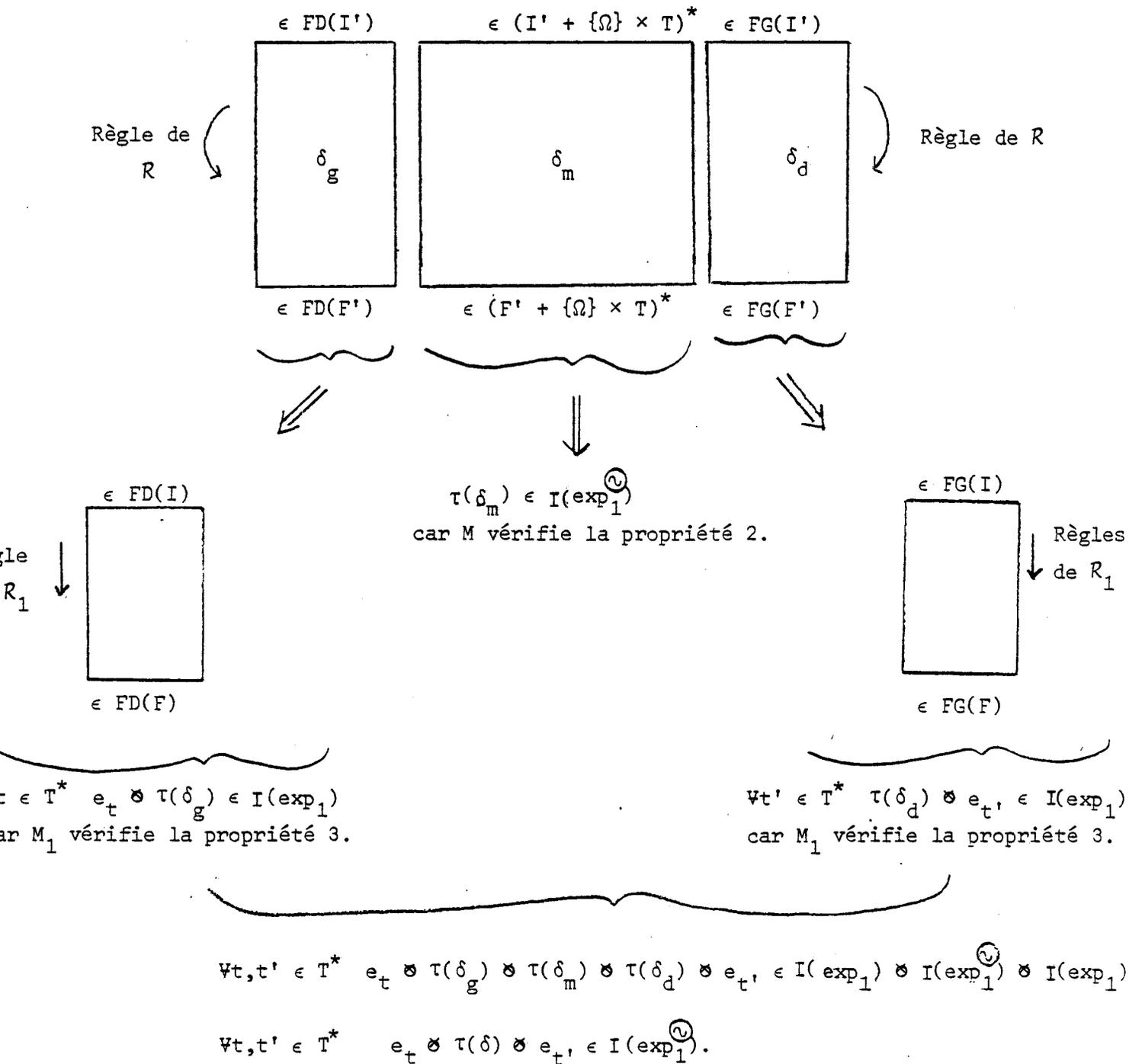
Notre intérêt se fixera donc sur le point 3 de la propriété 3, et sur quelques conformations particulières des facteurs stricts de I .

Soit $\delta \in \tilde{DP}(\Sigma)$, $m_i \in FS(I)$, $m_f \in FS(F)$ tels que :

$$m_i \cdot \delta \Big|_M^* \delta \cdot m_f.$$

1) Si $m_i \in FD(I') \otimes (I' + \{\Omega\} \times T)^* \otimes FG(I')$, alors obligatoirement $m_f \in FD(F') \otimes (F' + \{\Omega\} \times T)^* \otimes FG(F')$.

$m_i \cdot \delta \Big|_M^* \delta \cdot m_f$ peut se décomposer suivant le schéma suivant :



2) Si $m_i \in (I' + \{\Omega\} \times T)^*$, obligatoirement $m_f \in (F' + \{\Omega\} \times T)^*$

$$\left. \begin{array}{l} \cdot m_i \cdot \delta \left| \frac{\star}{M} \right. \delta \cdot m_f \\ M \text{ vérifie la propriété 2} \end{array} \right\} \Rightarrow \tau(\delta) \in I(\exp_1^{\circledast})$$

d'où

$$\forall t, t' \in T^* \quad e_t \otimes \tau(\delta) \otimes e_{t'} \in J(\exp_1^{\circledast}).$$

□

e) exp est le produit de composition de deux expressions simples de dags

$$\text{Soit } \exp = \exp_1 \cdot \exp_2.$$

D'après notre hypothèse d'induction.

$\exists M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ qui vérifie les propriétés 1, 2, 3 pour \exp_1

$\exists M_2 = \langle \Sigma, Q_2, I_2, F_2, R_2 \rangle$ qui vérifie les propriétés 1, 2 et 3 pour \exp_2

avec $\underline{Q_1 \cap Q_2 = \emptyset}$.

Soit M'_1 et M'_2 les automates construits à partir de M_1 et M_2 par les constructions définies en V.2.4.

$$\cdot M'_1 = \langle \Sigma, K_1 \times Q_1 \times K_1, I'_1, F'_1, R'_1 \rangle \text{ où}$$

$$I'_1 = \Delta_{k_{1_0} k_{1_f}} \cap \Pi_{Q_1}^{-1}(I_1)$$

$$F'_1 = \lambda_1^*$$

$$R'_1 = \{m \cdot a \vdash a \cdot m' / \exists k_{1_i}, k_{1_j} \in K_1 \text{ tels que } m \text{ et } m' \in \Delta_{k_{1_i} k_{1_j}}\}$$

$$\text{et } \Pi_{Q_1}(m) \cdot a \vdash a \cdot \Pi_{Q_1}(m') \in R_1\}$$

$$\bullet M'_2 = (\Sigma, V_2 \times Q_2 \times V_2, I'_2, F'_2, R'_2) \text{ où}$$

$$I'_2 = \mu_2^*$$

$$F'_2 = \Delta v_{2_0} v_{2_f} \cap \Pi_{Q_2}^{-1}(F_2)$$

$$R'_2 = \{m \cdot a \mid a \cdot m' / \exists v_{2_i}, v_{2_j} \in V_2 \text{ tels que } m \text{ et } m' \in \Delta v_{2_i} v_{2_j}$$

$$\text{et } \Pi_{Q_2}(m) \cdot a \mid a \cdot \Pi_{Q_2}(m') \in R_2\}.$$

Nous allons montrer, dans un premier temps, que l'automate M'_1 (resp. M'_2) vérifie les propriétés 1, 2, 3 pour exp_1 (resp. exp_2).

M'_1 et M'_2 vérifient la propriété 1 pour respectivement exp_1 , et exp_2 ;
chaque règle de R'_1 (ou R'_2) étant bâtie à partir d'une règle de R_1 (ou R_2).

M'_1 et M'_2 vérifie la propriété 2, en effet :

Point 1 de la propriété 2 :

$$\forall \delta \in \hat{D}\hat{P}(\Sigma), \forall m_i \in I'_1 \quad \forall m_f \in F'_1$$

$$m_i \cdot \delta \mid_{M'_1}^* \delta \cdot m_f \Rightarrow \left| \begin{array}{l} \Pi_{Q_1}(m_i) \cdot \delta \mid_{M_1}^* \delta \cdot \Pi_{Q_1}(m_f) \\ \text{et } \Pi_{Q_1}(m_i) \in I_1 \\ \text{et } \Pi_{Q_1}(m_f) \in F_1 \end{array} \right.$$

la projection sur T des règles employées lors de la reconnaissance dans M'_1 et dans M_1 sont les mêmes.

Comme M_1 vérifie la proposition 2, on a donc $\tau(\delta) \in I(\text{exp}_1)$.

□

Un raisonnement tout-à-fait comparable permettrait de démontrer que M'_2 vérifie lui-aussi le premier point de la propriété 2.

Point 2 de la propriété 2 :

- $\forall \delta' \in I(\text{exp}_1) \quad \exists \delta \in T(M_1)$ tel que $\tau(\delta) = \delta'$ (propriété de M_1).

En plus de l'égalité de $T(M_1)$ et $T(M'_1)$, à toute reconnaissance dans M_1 , on peut associer une reconnaissance dans M'_1 qui utilise des règles dont les projections sur Q_1 sont les règles utilisées pour la reconnaissance dans M_1 , c'est-à-dire qu'il y a conservation de $\tau(\delta)$, d'où

- $\forall \delta' \in I(\text{exp}_1) \quad \exists \delta \in T(M'_1)$ tel que $\tau(\delta) = \delta'$.

□

- Idem pour M'_2 .

L'argumentation précédente suffit pour justifier que M'_1 et M'_2 vérifient également la propriété 3.

□

Soit la substitution fini S de

$$K_1 \times Q_1 \times K_1 \rightarrow (V_2 \times Q_2 \times V_2) \cup (K_1 \times O_1 \times K_1)$$

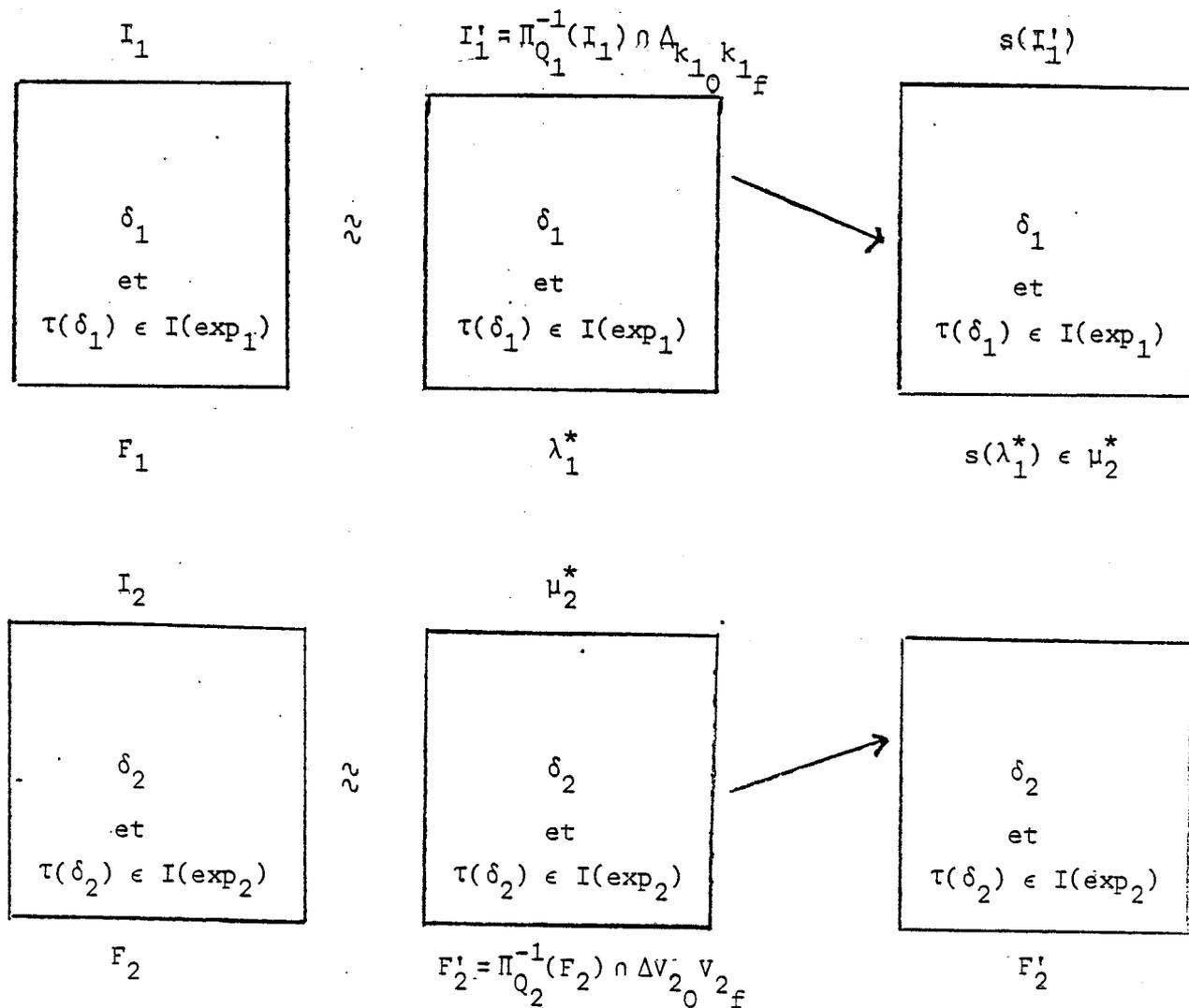
telle que • $\forall w \in \lambda_1 \quad s(w) = \{w\} \cup \{u \in \mu_2 / \Pi_T(u) = \Pi_T(w)\}$

- $\forall w \notin \lambda_1 \quad s(w) = \{w\}$.

Idée de la construction de l'automate M qui vérifie le lemme pour $\text{exp} = \text{exp}_1 \cdot \text{exp}_2$

Pour reconnaître les éléments de $I(\text{exp}_1) \cdot I(\text{exp}_2)$, il faut que consécutivement à la reconnaissance d'un élément de $I(\text{exp}_1)$, on puisse entamer la reconnaissance d'un élément de $I(\text{exp}_2)$. Le passage d'un élément de $I(\text{exp}_1)$ à un élément de $I(\text{exp}_2)$ se fait en cours de reconnaissance, la possibilité d'un contrôle bas sur l'élément de $I(\text{exp}_1)$, et d'un contrôle haut sur l'élément de $I(\text{exp}_2)$ est exclue, on a donc recours à la construction exposée en V.2.4. Ce passage sera donc caractérisé par la transformation d'un état appartenant à λ_1 en un état appartenant à μ_2 , ceci en préservant le type : c'est le rôle de la substitution s.

Schéma :



L'automate M vérifiant les propriétés 1, 2, 3 pour $\text{exp} = \text{exp}_1 \text{ exp}_2$ sera donc

$M = \langle \Sigma, Q, I, F, R \rangle$ dans lequel

$$Q = (K_1 \times Q_1 \times K_1) \cup (V_2 \times Q_2 \times V_2)$$

$$I = s(\Pi_{Q_1}^{-1}(I_1) \cap \Delta_{k_{10} k_{1f}})$$

$$F = \Pi_{Q_2}^{-1}(F_2) \cap \Delta_{v_{20} v_{2f}}$$

$$R = R_1' \cup R_2' \text{ où } R_1' = \{m \cdot a \mid a \cdot s(m) \text{ t\u00e9l que } a \cdot m \mid a \cdot m' \in R_1'\}.$$

Il reste à montrer que l'automate ainsi construit vérifie bien les propriétés 1, 2, 3 pour $\text{exp} = \text{exp}_1 \cdot \text{exp}_2$.

La propriété 1 est manifestement vérifiée, les projections sur Q_1 ou Q_2 des règles de R sont inchangées par rapport aux règles de R'_1 et R'_2 .

M satisfait aussi à la propriété 2.

Preuve du premier point :

$$\forall \delta \in \hat{D}\mathcal{P}(\Sigma) \quad \forall m_i \in I \quad \forall m_f \in F$$

$$m_i \cdot \delta \stackrel{*}{M} \delta \cdot m_f \Rightarrow \left\{ \begin{array}{l} \delta = \delta_1 \cdot \delta_2 \text{ et} \\ m_i \cdot \delta_1 \stackrel{*}{R'_1} \delta_1 \cdot m ; m \in \mu_2^* \\ m \cdot \delta_2 \stackrel{*}{R'_2} \delta_2 \cdot m_f \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{l} s^{-1}(m_i) \cdot \delta_1 \stackrel{*}{R'_1} \delta_1 \cdot s^{-1}(m) ; s^{-1}(m_i) \in \Delta k_{1_0} k_{1_f} \cap \Pi_{Q_1}^{-1}(I) \\ \text{et } s^{-1}(m) \in \lambda_1^* \\ m \cdot \delta_2 \stackrel{*}{R'_2} \delta_2 \cdot m_f ; m \in \mu_2^* \text{ et } m_f \in \Delta v_{2_0} v_{2_f} \cap \Pi_{Q_2}^{-1}(F_2) \end{array} \right.$$

\Rightarrow comme M'_1 et M'_2 vérifient la propriété pour exp_1 et exp_2 ; $\tau(\delta_1) \in I(\text{exp}_1)$; $\tau(\delta_2) \in I(\text{exp}_2)$.

Signature de $\tau(\delta_1) = \Pi_T(m) = \text{co-signature de } \tau(\delta_2)$.

$$\Rightarrow \tau(\delta_1) \cdot \tau(\delta_2) = \tau(\delta_1 \cdot \delta_2) \in I(\text{exp}_1) \cdot I(\text{exp}_2) = I(\text{exp}_1 \cdot \text{exp}_2).$$

□

Preuve du deuxième point :

$\forall \delta' \in I(\text{exp}_1 \cdot \text{exp}_2) : \delta' = \delta'_1 \cdot \delta'_2$ avec co-signature de $\delta'_2 =$ signature de δ_1 , $\delta'_1 \in I(\text{exp}_1)$ et $\delta'_2 \in I(\text{exp}_2)$.

- $\delta'_1 \in I(\exp_1) \Rightarrow \left\{ \begin{array}{l} \text{de part la propriété 2 vérifiée par } M'_1 \\ \exists \delta_1 \in T(M'_1) \text{ tel que } \tau(\delta_1) = \delta'_1 \end{array} \right.$
- $\delta'_2 \in I(\exp_2) \Rightarrow \left\{ \begin{array}{l} \text{de part la propriété 2 vérifiée par } M'_2 \\ \exists \delta_2 \in T(M'_2) \text{ tel que } \tau(\delta_2) = \delta'_2 \end{array} \right.$
- $\delta_1 \in T(M'_1) \Rightarrow \exists m_{i_1} \in I'_1 \text{ et } m_{f_1} \in \lambda_1^* \text{ tels que } m_{i_1} \cdot \delta_1 \stackrel{*}{\mid}_{M'_1} \delta_1 \cdot m_{f_1}$
- $\delta_2 \in T(M'_2) \Rightarrow \exists m_{i_2} \in \mu_2^* \text{ et } m_{f_2} \in F'_2 \text{ tels que } m_{i_2} \cdot \delta_2 \stackrel{*}{\mid}_{M'_2} \delta_2 \cdot m_{f_2}$
- $m_{i_2} \in s(m_{f_1})$ car co-signature de $\delta'_2 =$ signature de δ'_1 .

On en déduit que :

$$\exists m'_{i_1} \in s(m_{i_1}) \text{ tel que } m'_{i_1} \cdot \delta_1 \stackrel{*}{\mid}_M \delta_1 \cdot m_{i_2}$$

en rapprochant le résultat précédent de $m_{i_2} \cdot \delta_2 \stackrel{*}{\mid}_{M'_2} \delta_2 \cdot m_{f_2}$, on obtient :

$$m'_{i_1} \cdot (\delta_1 \cdot \delta_2) \stackrel{*}{\mid}_M (\delta_1 \cdot \delta_2) \cdot m_{f_2}$$

d'où $\delta_1 \cdot \delta_2 \in T(M)$ et $\tau(\delta_1 \cdot \delta_2) = \delta'$.

□

M satisfait la propriété 3.

Preuve :

Nous nous attacherons toujours à démontrer qu'un seul point de cette propriété, les autres preuves découlant de celle que nous donnons.

Preuve du point 3 :

Soient $\delta \in \overset{\vee}{DP}(\Sigma)$, $m_i \in FS(I)$ et $m_f \in FS(F)$ tels que

$$m_i \cdot \delta \stackrel{*}{\mid}_M \delta \cdot m_f$$

Toute reconnaissance dans M peut être divisée en deux mouvements, un premier dans lequel n'interviennent que des règles de R_1' , et un second n'utilisant que des règles de R_2' .

$$m_i \cdot \delta \vdash \delta \cdot m_f \Rightarrow \left\{ \begin{array}{l} \delta = \delta_1 \cdot \delta_2 \\ m_i \cdot \delta_1 \vdash_{R_1'}^* \delta_1 \cdot m ; m \in \mu_2^* \\ m \cdot \delta_2 \vdash_{R_2'}^* \delta_2 \cdot m_f \end{array} \right.$$

$$\Rightarrow \left\{ \begin{array}{l} s^{-1}(m_i) \cdot \delta_1 \vdash_{R_1'}^* \delta_1 \cdot s^{-1}(m) \\ s^{-1}(m_i) \in \text{FS}(I_1') \text{ et } s^{-1}(m) \in \lambda_1^* \\ m \cdot \delta_2 \vdash_{R_2'}^* \delta_2 \cdot m_f \\ m \in \mu_2^* \text{ et } m_f \in \text{FS}(F_2') \end{array} \right.$$

\Rightarrow Nous sommes placés dans les conditions d'application de la propriété pour M_1' et M_2'

$$\forall t, t' \in T^* \quad e_t \otimes \tau(\delta_1) \otimes e_{t'} \in I(\text{exp}_1)$$

et $e_t \otimes \tau(\delta_2) \otimes e_{t'} \in I(\text{exp}_2)$

$\Rightarrow \forall t, t' \in T^* \quad e_t \otimes \tau(\delta) \otimes e_{t'} \in I(\text{exp}).$

□

f) exp est le produit de composition itéré d'une expression simple de dag

Soit $\text{exp}' = \text{exp}^*$

Alors d'après notre hypothèse d'induction :

$\exists M = \langle \Sigma, Q, I, F, R \rangle$ automate de dags planaires qui vérifie les propriétés 1, 2, 3 pour exp .

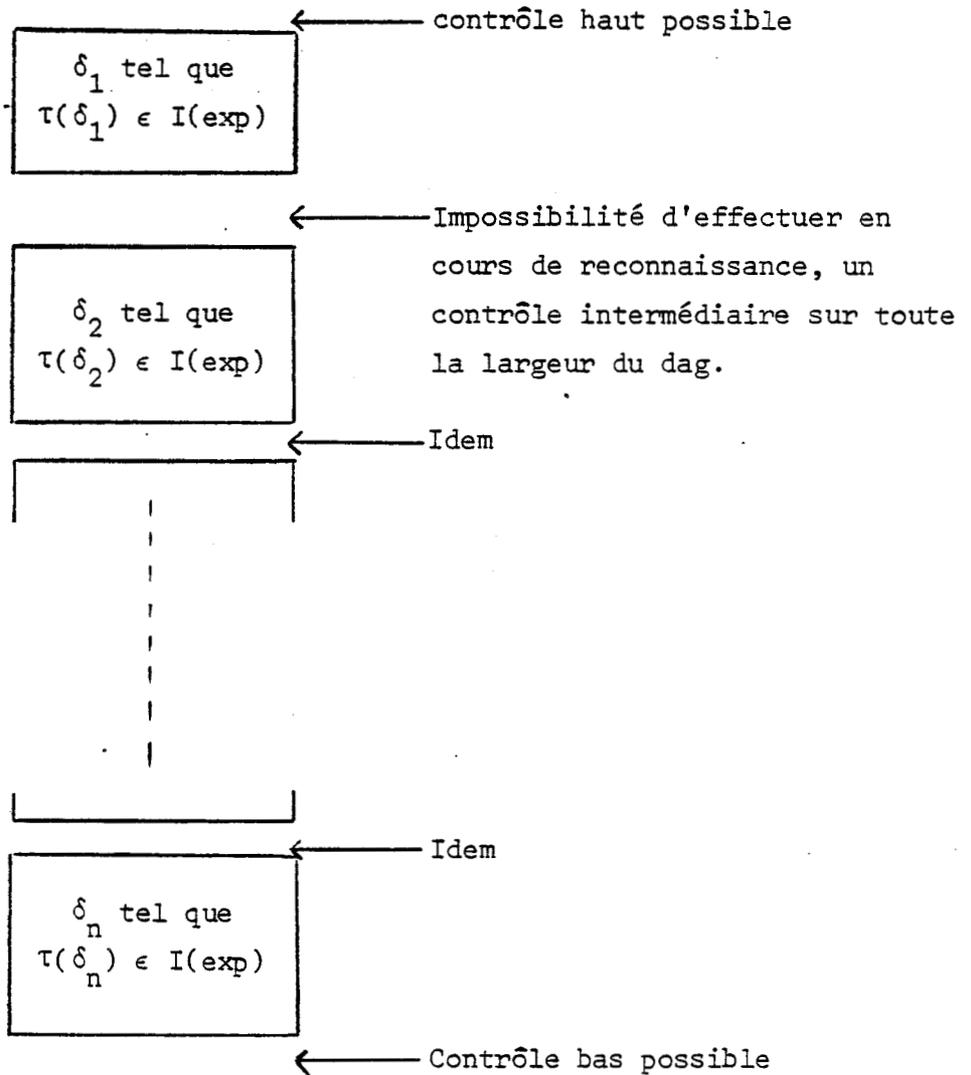
C'est à partir de cet automate, que nous allons construire l'automate qui reconnaît $I(\text{exp}^*)$.

Analyse du problème :

Tout dag, élément de $I(\text{exp}^*)$ est une succession d'éléments de $I(\text{exp})$ composés les uns aux autres.

Une cascade de reconnaissance d'éléments de $I(\text{exp})$ pose certains problèmes que nous mettrons en évidence sur un schéma :

Forme des dags
à reconnaître :



Il faut donc parvenir à réduire le contrôle au niveau du passage d'un dag de $I(\text{exp})$ au suivant, à un simple contrôle individuel sur les états assurant cette transition. Les contraintes qu'exprimaient les contrôles haut et bas devront être intégrés dans les mouvements de reconnaissance des dags.

Une construction apparentée à la construction exposée en V.2.4. permet de répondre à ces exigences, mais uniquement sur des dags connexes. Cette première construction doit être considéré comme une ébauche très incomplète de l'automate recherché.

Première étape de la construction :

Les notations retenues ici sont celles adoptées au paragraphe V.2.4.

Soit l'automate $M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ où

$$Q_1 = V \times K \times Q \times K \times V$$

$$I_1 = \{(v_0, k_0, q, k, v) / (v_0, q, v) \in \mu \text{ et } k \in K_1 ; v \in V_1\}.$$

$$\{(v, k, q, k', v') / (v, q, v') \in \mu \text{ et } k, k' \in K_1 ; v, v' \in V_1\}^*.$$

$$\{(v, k, q, k_f, v_f) / (v, q, v_f) \in \mu \text{ et } k \in K_1 ; v \in V_1\}$$

$$\cup \{(v_0, k_0, q, k_f, v_f) / (v_0, q, v_f) \in \mu\}.$$

$$F_1 = \Pi_{K \times Q \times K}^{-1}(\lambda^*) = \{(v_i, k_i, q, k_j, v_j) / (k_i, q, k_j) \in \lambda\}^*$$

$$R_1 = \{m \cdot a \vdash a \cdot m' / \Pi_Q(m) \cdot a \vdash a \cdot \Pi_Q(m') \in R \text{ et}$$

$$\exists k_i, k_j \in K \text{ tels que } \Pi_{K \times Q \times K}(m) \text{ et } \Pi_{K \times Q \times K}(m') \in \Delta k_i k_j$$

$$\exists v_i, v_j \in V \text{ tels que } \Pi_{V \times Q \times V}(m) \text{ et } \Pi_{V \times Q \times V}(m') \in \Delta v_i v_j\}$$

Cet automate possède les propriétés suivantes :

• $\forall \delta \in \hat{DP}(\Sigma)$; δ connexe.

$$1) \left. \begin{array}{l} m_i \in FG(I_1), m_f \in F_1 \\ \text{tels que} \\ m_i \cdot \delta \Big|_{M_1}^* \delta \cdot m_f \end{array} \right| \Rightarrow \left\{ \begin{array}{l} \Pi_Q(m_i) \in FG(I) \\ \Pi_Q(m_f) \in FG(F) \\ \text{et } \Pi_Q(m_i) \cdot \delta \Big|_M^* \delta \cdot \Pi_Q(m_f) \end{array} \right.$$

$$2) \left. \begin{array}{l} m_i \in FD(I_1), m_f \in F_1 \\ \text{tel que} \\ m_i \cdot \delta \Big|_{M_1}^* \delta \cdot m_f \end{array} \right| \Rightarrow \left\{ \begin{array}{l} \Pi_Q(m_i) \in FD(I) \\ \Pi_Q(m_f) \in FG(F) \\ \text{et } \Pi_Q(m_i) \cdot \delta \Big|_M^* \delta \cdot \Pi_Q(m_f) \end{array} \right.$$

$$3) \left. \begin{array}{l} m_i \in FS(I_1), m_f \in F_1 \\ \text{tels que} \\ m_i \cdot \delta \Big|_{M_1}^* \delta \cdot m_f \end{array} \right| \Rightarrow \left\{ \begin{array}{l} \Pi_Q(m_i) \in FS(I) \\ \Pi_Q(m_f) \in FS(F) \\ \text{et } \Pi_Q(m_i) \cdot \delta \Big|_M^* \delta \cdot \Pi_Q(m_f) \end{array} \right.$$

Preuve :

Ces propriétés résultent directement de la propriété énoncée et démontrée en V.2.4, en effet :

L'automate M_1 vérifie la propriété V.2.4. pour ses projections sur $K \times Q \times K$ et sur $V \times Q \times V$.

On a donc :

$$\left. \begin{array}{l} \delta \text{ connexe} \\ m_i \cdot \delta \Big|_{M_1}^* \delta \cdot m_f \end{array} \right| \Rightarrow \left\{ \begin{array}{l} \exists k_i, k_j \in K \text{ tels que } \Pi_{K \times Q \times K}(m) \text{ et} \\ \Pi_{K \times Q \times K}(m') \in \Delta k_i k_j \\ \exists v_i, v_j \in V \text{ tels que } \Pi_{V \times Q \times V}(m) \text{ et} \\ \Pi_{V \times Q \times V}(m') \in \Delta v_i v_j \end{array} \right.$$

On en déduit immédiatement la propriété 1 précédente.

$$\left. \begin{array}{l} m_i \in FG(I_1), m_f \in F_1 \\ \text{tel que} \\ m_i \cdot \delta \Big|_{M_1}^* \delta \cdot m_f \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \exists y_j \in Y_1 \text{ tel que } \Pi_{Y \times D \times Y}(m_i) \in \Delta y_j \cap \mu^* \\ \exists k_j \in K_1 \text{ tel que } \Pi_{K \times Q \times K}(m_f) \in \Delta k_j \cap \lambda^* \end{array} \right.$$

Les transitions de l'automate A_I permettent de passer de v_0 à v_j en lisant $\Pi_Q(m_i)$; m_i peut donc être prolongé en un mot de I , d'où $\Pi_Q(m_i) \in FG(I)$.

Les transitions de l'automate A_F permettent de passer de k_0 à k_j en lisant $\Pi_Q(m_f)$, m_f peut donc être prolongé en un mot de F , d'où $\Pi_Q(m_f) \in FG(F)$.

Les preuves des propriétés 2 et 3 s'appuient sur des raisonnements calqués sur le précédent.

Deuxième étape de la construction :

Pour reconnaître effectivement plusieurs éléments de $I(\text{exp})$ à la suite les uns des autres, il nous faut compléter les règles de façon à ce qu'en chaque endroit où un état final pouvait apparaître en cours de reconnaissance, on puisse lui substituer un état initial transportant le même type pour démarrer la reconnaissance de l'élément suivant de $I(\text{exp})$.

Un état final est caractérisé par le fait qu'il comporte une transition de l'automate qui reconnaît F , alors qu'un état initial comporte une transition de l'automate qui reconnaît I .

La substitution rationnelle s_1 que nous allons définir, formalise la transformation d'état final en état initial en préservant le type transporté.

Soit donc $s_1 : Q_1 \rightarrow Q_1^*$ telle que

$$\forall w \in Q_1$$

- $\Pi_{K \times Q \times K}(w) \in \lambda \Rightarrow s_1(w) = \{w\}$
- $w = (k_0, v_0, q, v, k)$ tels que $(k_0, q, k) \in \lambda$ et $v \in V_1, k \in K_1$
 $\Rightarrow s_1(w) = \{(k_0, v_0, q', v', k') \in Q_1 / (v_0, q', v') \in \mu ;$
 $v' \in V_1, k' \in K_1 \text{ et } \Pi_T(q) = \Pi_T(q')\} \cup \{w\}$
- $w = (k, v, q, v_f, k_f)$ tels que $(k, q, k_f) \in \lambda$ et $v \in V_1, k \in K_1$
 $\Rightarrow s_1(w) = \{(k', v', q', v_f, k_f) \in Q_1 / (v', q', v_f) \in \mu$
 $v' \in V_1, k' \in K_1 \text{ et } \Pi_T(q) = \Pi_T(q')\} \cup \{w\}$

• $w = (k_0, v_0, q, v_f, k_f)$ tels que $(k_0, q, k_f) \in \lambda$
 $\Rightarrow s_1(w) = \{(k_0, v_0, q', v_f, k_f) \in Q_1 / (v_0, q', v_f) \in \mu ;$
 et $\Pi_T(q) = \Pi_T(q')\} \cup \{w\}$.

• w ne rentre dans aucun des cas précités

$\Rightarrow s(w) = \{(k, v, q, v', k') \in Q_1 / (v, q, v') \in \mu ;$
 $v, v' \in V_1 ; k, k' \in K_1 ; \text{ et } \Pi_T(w) = \Pi_T(q)\} \cup \{w\}$.

L'automate M_2 suivant nous rapproche un peu plus du but recherché.

$M_2 = \langle \Sigma, Q_2, I_2, F_2, R_2 \rangle$ où

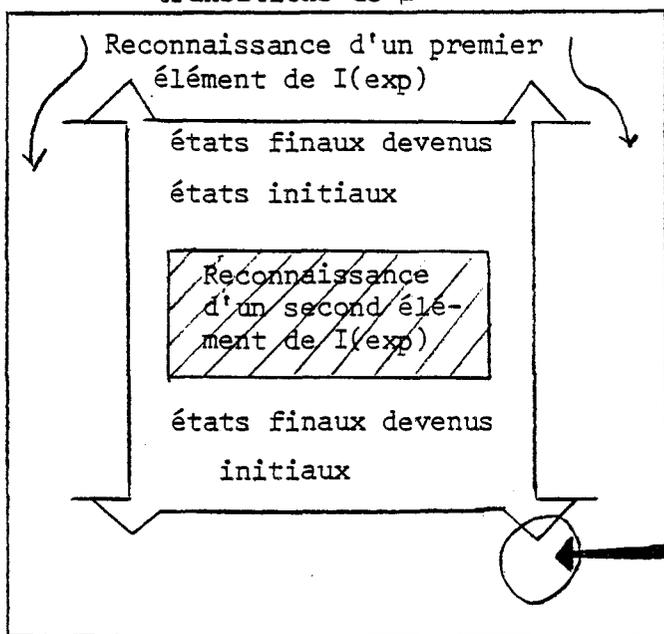
$Q_2 = Q_1 ; I_2 = I_1 ; F_2 = F_1$

$R_2 = \{m \cdot A \mid a \cdot s_1(m') / m \cdot a \mid a \cdot m' \in R_1\}$.

Cet automate reconnaît bien tous les dags δ , tels que $\tau(\delta)$ appartient à $I(\text{exp}^*)$, mais pourrait en reconnaître d'autres le cas échéant. Dans notre construction, aucun contrôle ne permet de différencier les états servant à la reconnaissance d'éléments normalement différents de $I(\text{exp})$; rien ne s'oppose donc à ce que ces états interviennent conjointement dans une même règle; ceci peut conduire à la reconnaissance d'un dag n'appartenant pas à $I(\text{exp}^*)$.

transitions de μ

Exemple :



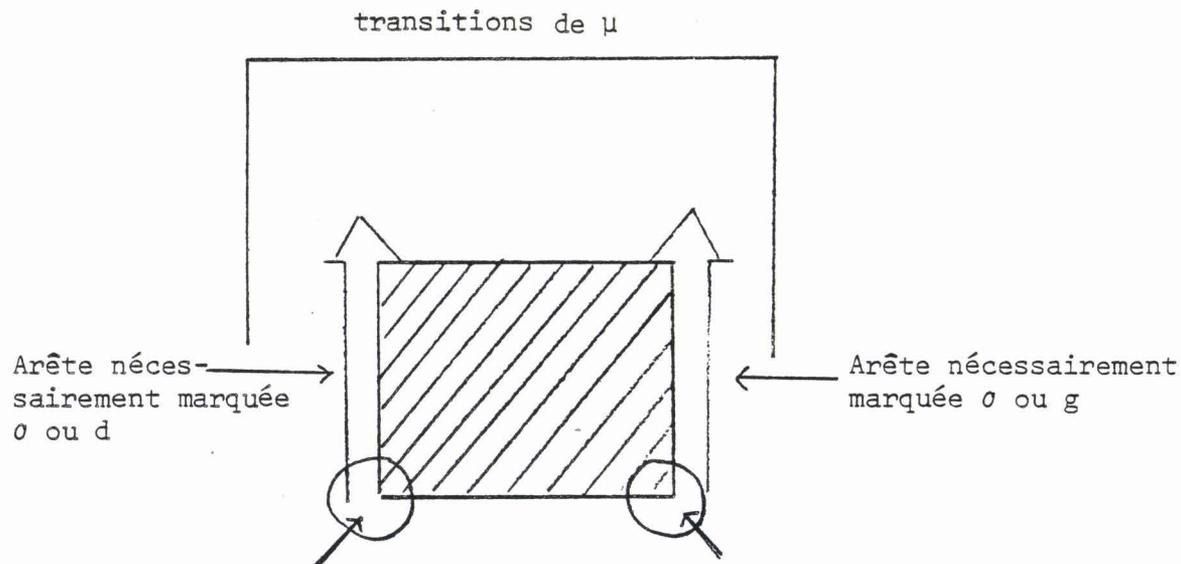
transitions de λ

Interférence entre des états qui participent à la reconnaissance du premier et ceux qui auraient du entâmer celle d'une troisième.

D'où la nécessité d'introduire dans nos états une nouvelle composante capable de "séparer" les reconnaissances d'éléments différents de $I(\text{exp})$. Le travail de cette composante sera donc, à la vue de l'exemple précédent, d'imposer des choix quant aux statuts des arêtes parcourues : arête droite, arête gauche, arête interne, extrémité d'arête. La reconnaissance sera bloquée dès que les choix considérés seront en désaccord avec la structure locale du dag :

Nous allons donc parallèlement à la reconnaissance des éléments de $I(\text{exp})$, promener les marques d , g , c , σ suivant les règles établies lors de la présentation de l'automate DG (cf. V.1.). Un nouveau marquage arbitraire sera généré à chaque transition : fin d'élément de $I(\text{exp})$ et début du suivant. La pertinence de ce choix sera éprouvée par la suite de la reconnaissance (blocage ou non).

Sur l'exemple précédent, le blocage se serait produit de la façon suivante :



un état marqué σ ou d
ne peut être placé qu'à
l'extrême droite d'une
partie gauche de règle.

un état marqué σ ou g
ne peut être placé qu'à
l'extrémité gauche d'une
partie gauche de règle.

L'automate M' qui satisfasse aux propriétés 1, 2, 3 pour \exp^* peut donc être élaboré comme suit :

L'automate M' :

Préliminaires :

Soit s la substitution rationnelle de $(Q_1 \times H)^*$ dans $(Q_1 \times H)^*$ définie par $\forall w \in Q_1 \times H$ si $\Pi_H(w) \neq \emptyset$ ou $\Pi_{K \times Q \times K}(w) \neq \emptyset$ & λ alors

$$\begin{array}{l} | \\ s(w) = w \\ \text{sinon} \\ | \\ s(w) = s_1(\Pi_{Q_1}(w)) \times H \\ \text{fsi} \end{array}$$

Définition de l'automate M' :

$$M' = \langle \Sigma, Q', I', F', R' \rangle$$

$$Q' = Q_1 \times H$$

$$I' = \Pi_{Q_1}^{-1}(I_1)$$

$$F' = \Pi_{K \times Q \times K}^{-1}(\lambda^*) \cap \Pi_H^{-1}(o^*)$$

$$R' = \{m \cdot a \mid a \cdot s(m') / m \cdot a \mid a \cdot m' \in R'_1\} \text{ et}$$

$$\begin{aligned} R'_1 = \{m \cdot a \mid a \cdot m' / & \cdot \Pi_Q(m) \cdot a \mid a \cdot \Pi_Q(m') \in R \\ & \cdot \Pi_H(m) \cdot a \mid a \cdot \Pi_H(m') \in P \\ & \cdot \exists k_i, k_j \text{ tels que } \Pi_{K \times Q \times K}(m) \text{ et } \Pi_{K \times Q \times K}(m') \in \Delta k_i, k_j \\ & \cdot \exists v_i, v_j \text{ tels que } \Pi_{V \times Q \times V}(m) \text{ et } \Pi_{V \times Q \times V}(m') \in \Delta v_i, v_j \} \end{aligned}$$

M' vérifie la propriété 1 du lemme, puisque les projections sur Q des règles de R' , sont des règles de R qui par hypothèse vérifient la propriété 1.

La démonstration du point 1 de la propriété 2 pour M' nécessite la preuve de la propriété liminaire suivante :

$$\begin{aligned} & \forall \delta \in \overset{\sim}{DP}(\Sigma) \quad \forall m_i \in I' \quad \forall m_f \in F' \\ & m_i \cdot \delta \stackrel{*}{\underset{M'}{\mid}} \delta \cdot m_f \Rightarrow \Pi_H(m_i) \in H^* \otimes (g \otimes c^* \otimes d + o) \otimes H^* \end{aligned}$$

Preuve :

Si δ appartient au taillis reconnu par M' , δ appartient également au taillis reconnu par l'automate DG' suivant :

$$DG' = \langle \Sigma, H, H^*, o^*, P' \rangle \text{ où}$$

P' est l'ensemble ci-après de règles :

$$\begin{aligned} P' = & \{(g \otimes c^*) \cdot a \mid a \cdot (H^* \otimes g \otimes c^*) / a \in \Sigma\} \cup \\ & \{c^* \cdot a \mid a \cdot (c^* + c^* \otimes d \otimes H^* \otimes g \otimes c^*)\} \cup \\ & \{(c^* \otimes d) \cdot a \mid a \cdot (c^* \otimes d \otimes H^*)\} \cup \\ & \{(o + g \otimes c^* \otimes d) \cdot a \mid a \cdot H^*\} \end{aligned}$$

et de plus $\Pi_H(m_i) \cdot \delta \stackrel{*}{\underset{DG'}{\mid}} \delta \cdot \Pi_H(m_f)$.

D'autre part nous pouvons établir que :

$$\begin{aligned} & \forall m_i \in c^* \otimes d \otimes H^* \quad \forall \delta \in \overset{\sim}{DP}(\Sigma) \\ & m_i \cdot \delta \stackrel{*}{\underset{DG'}{\mid}} \delta \cdot m_f \Rightarrow m_f \in c^* \otimes d \otimes H^* \end{aligned}$$

On observe, en effet que l'application d'une règle de P' , conserve l'appartenance du mot d'états obtenu au langage $c^* \otimes d \otimes H^*$.

i) La règle appartient à $\{(g \otimes c^*) \cdot a \vdash a \cdot (H^* \otimes g \otimes c^*) / a \in \Sigma\}$

$$m_i \in c^* \otimes d \otimes \underbrace{H^*}_{\downarrow}$$

$$m_f \in c^* \otimes d \otimes H^*$$

ii) La règle appartient à $\{c^* \cdot a \vdash a \cdot (c^* + c^* \otimes d \otimes H^* \otimes g \otimes c^*)\}$

$$m_i \in \underbrace{c^*}_{\swarrow} \otimes d \otimes \underbrace{H^*}_{\searrow}$$

$$m_f \in (c^* + c^* \otimes d \otimes H^* \otimes g \otimes c^*) \otimes d \otimes H^*$$

$$= c^* \otimes d \otimes H^* + c^* \otimes d \otimes H^* \otimes g \otimes c^* \otimes d \otimes H^* \otimes c^* \otimes d \otimes H^*$$

$$m_f \in c^* \otimes d \otimes H^*$$

iii) La règle appartient à $\{(c^* \otimes d) \cdot a \vdash a \cdot (c^* \otimes d \otimes H^*) / a \in \Sigma\}$

$$m_i \in \underbrace{c^* \otimes d}_{\swarrow} \otimes \underbrace{H^*}_{\searrow}$$

$$m_f \in \underbrace{c^* \otimes d \otimes H^*}_{\swarrow} \otimes H$$

$$m_f \in c^* \otimes d \otimes H^*$$

iv) La règle appartient à $\{(o + g \otimes c^* \otimes d) \cdot a \vdash a \cdot H^* / a \in \Sigma\}$

$$m_i \in c^* \otimes d \otimes \underbrace{H^*}_{\downarrow}$$

$$m_f \in c^* \otimes d \otimes H^*$$

De façon, tout-à-fait symétrique, on peut établir que

$$\forall m_i \in H^* \otimes g \otimes c^* \quad \forall \delta \in \overset{\vee}{DP}(\Sigma)$$

$$m_i \cdot \delta \Big|_{DG^*}^* \delta \cdot m_f \Rightarrow m_f \in H^* \otimes g \otimes c^*$$

Les constatations précédentes vont nous permettre de démontrer par l'absurde que $\Pi_H(m_i) \in H^* \otimes g \otimes c^* \otimes d \otimes H^*$.

Supposons que $\Pi_H(m_i) \notin H^* \otimes g \otimes c^* \otimes d \otimes H^* \Rightarrow \Pi_H(m_i) \in (c + d)^* \otimes (c + g)$
 d'où $\Pi_H(m_i) \in H^* \otimes g \otimes c^*$ ou $\Pi_H(m_i) \in c^* \otimes d \otimes H^*$ ou $\Pi_H(m_i) \in c^*$; d'après les comportements de l'automate mis en évidence précédemment le mot final d'états ne peut pas être constitué uniquement de \emptyset .

□

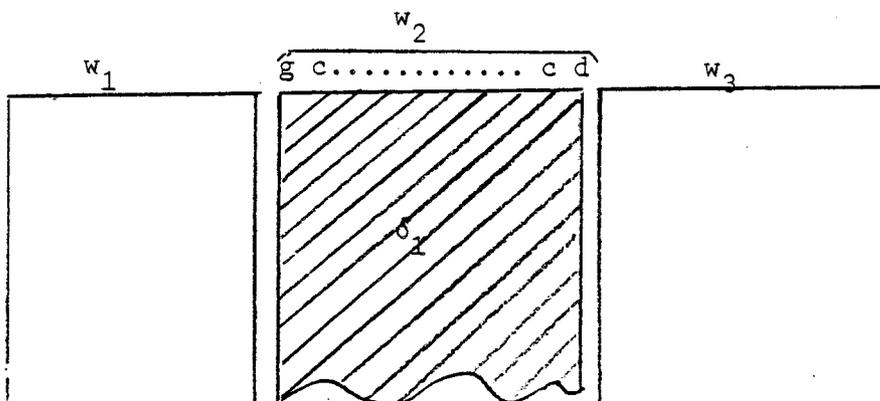
Preuve du point 1 de la propriété 2 :

$$\forall \delta \in \overset{\vee}{DP}(\Sigma) \quad \forall m_i \in I' \quad \forall m_f \in F'$$

$$m_i \cdot \delta \Big|_{M'}^* \delta \cdot m_f \Rightarrow \Pi_H(m_i) \in H^* \otimes g \otimes c^* \otimes d \otimes H^* \text{ (cf. propriété liminaire)}$$

Nous allons examiner le sous-dag reconnu à partir du sous-mot w_2 de m_i dont la projection sur H appartient à $g \otimes c^* \otimes d$, et ceci tant que les règles appliquées n'utilise pas en partie gauche des états qui auraient subis, par l'application de la substitution s , la transformation d'état final en état initial.

Schéma :



$s(w)$: mot constitué d'états terminaux transformés en initiaux.

Si on a $w_2 \cdot \delta_1 \left| \frac{\star}{M'} \right. \delta_1 \cdot s(w)$, on a également $w_2 \cdot \delta_1 \left| \frac{\star}{M'} \right. \delta_1 \cdot w$

On en déduit donc :

$$\left. \begin{array}{l} \Pi_H(w_2) \cdot \delta_1 \left| \frac{\star}{DG} \right. \delta_1 \cdot \Pi_H(w) \\ \Pi_H(w_2) \in g \otimes c^* \otimes d \\ \Pi_H(w) \in o^* \end{array} \right\} \Rightarrow \delta_1 \text{ connexe} \\ \text{(cf. propriété V.2.3.)}$$

$$\delta_1 \text{ connexe} \Rightarrow \exists v_i, v_j \in V \text{ tel que } \Pi_{V \times Q \times V}(w_2) \in \Delta v_i, v_j \\ \text{(cf. propriété V.2.4.)}$$

$$\delta_1 \text{ connexe} \Rightarrow \exists k_i, k_j \in V \text{ tel que } \Pi_{K \times Q \times K}(w_2) \in \Delta k_i, k_j \\ \text{(cf. propriété V.2.4.)}$$

$$\left. \begin{array}{l} \Pi_{V \times Q \times V}(w_2) \in \Delta v_i, v_j \\ \Pi_{V \times Q \times V}(w_2) \in \mu \end{array} \right\} \Rightarrow \Pi_Q(w_2) \in \text{FG}(I) \text{ ou } \text{FD}(I) \text{ ou } \text{FS}(I)$$

$$\left. \begin{array}{l} \Pi_{K \times Q \times K}(w) \in \Delta k_i, k_j \\ \Pi_{K \times Q \times K}(w) \in \lambda^* \end{array} \right\} \Rightarrow \Pi_Q(w) \in \text{FG}(F) \text{ ou } \text{FD}(F) \text{ ou } \text{FS}(I)$$

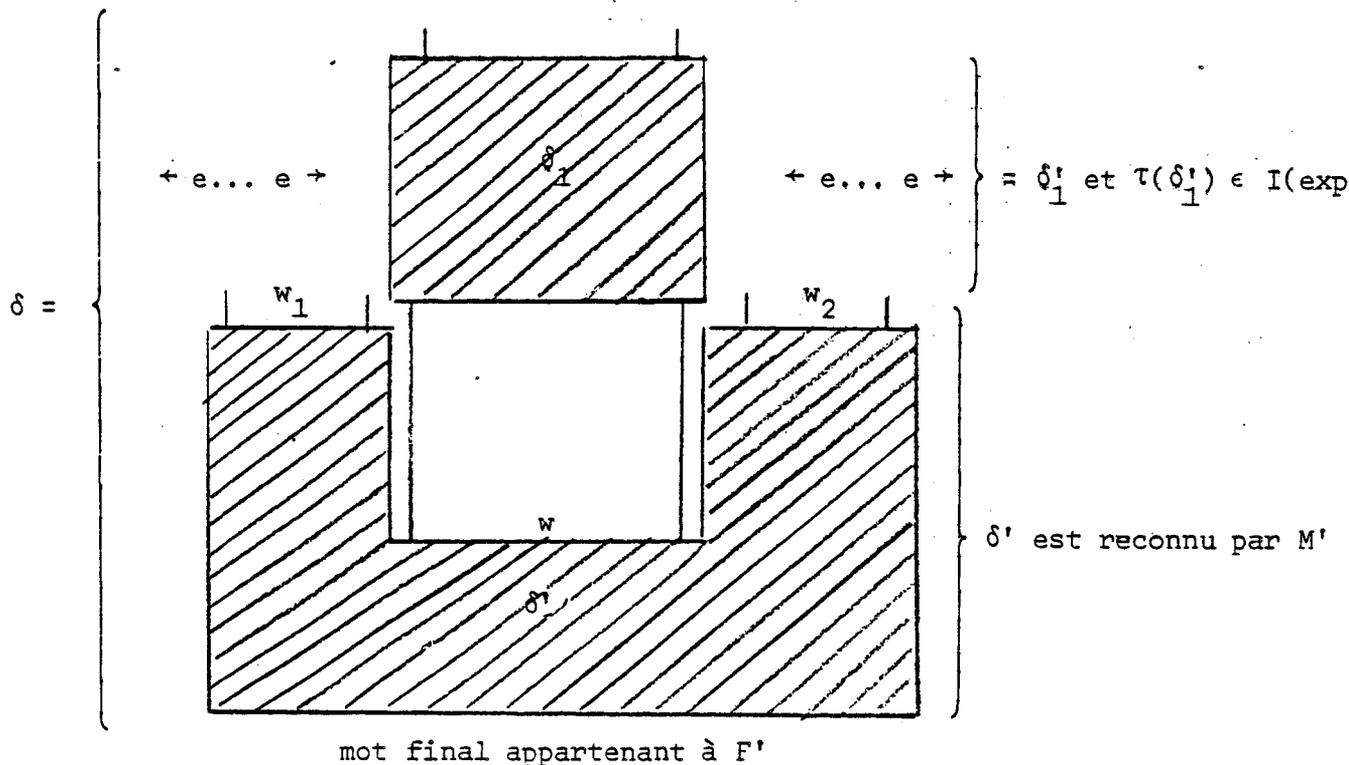
On a donc, conformément à la propriété 3 vérifiée par M.

$$\text{Soit } \forall t \in T^* \quad \tau(\delta_1) \otimes e_t \in I(\text{exp})$$

$$\text{Soit } \forall t \in T^* \quad e_t \otimes \tau(\delta_1) \in I(\text{exp})$$

$$\text{Soit } \forall t, t' \in T^* \quad e_t \otimes \tau(\delta_1) \otimes e_{t'} \in I(\text{exp}).$$

Nous venons de faire la preuve que



En poursuivant jusqu'à l'extrême cette décomposition des dags reconnus par M' en un dag dont le "typage" induit par sa reconnaissance dans M' est un élément de $I(\text{exp})$ et un autre élément du taillis reconnu par M' , on obtient

$$\delta = \delta'_1 \cdot \delta'_2 \dots \delta'_n \text{ avec } \forall i \in \{1, \dots, n\} \quad \tau(\delta'_i) \in I(\text{exp})$$

d'où $\tau(\delta) = \tau(\delta'_1) \cdot \tau(\delta'_2) \dots \tau(\delta'_n) \in I(\text{exp}^*)$.

□

Preuve du point 2 de la propriété 2 :

$$\forall \delta' \in I(\text{exp}^*) \quad \exists \delta'_1, \delta'_2, \dots, \delta'_n \in I(\text{exp}) \text{ tels que}$$

$$\delta' = \delta'_1 \cdot \delta'_2 \cdot \dots \cdot \delta'_n.$$

$\forall \delta'_i \quad \exists \delta_i \in T(M)$ tel que $\tau(\delta_i) = \delta'_i$ (cf. propriété 2 de M) or si $\delta_i \in T(M)$ alors $\delta_i \in T(M')$ et son marquage $\tau(\delta_i)$ est identique d'où $\forall \delta'_i \quad \exists \delta_i \in T(M')$ tel que $\tau(\delta_i) = \delta'_i$.

D'après les règles de M' , on a aussi $\delta_1 \cdot \delta_2 \dots \delta_n \in T(M')$ et $\tau(\delta_1 \cdot \delta_2 \dots \delta_n) = \tau(\delta_1) \cdot \tau(\delta_2) \dots \tau(\delta_n) = \delta'_1 \cdot \delta'_2 \dots \delta'_n = \delta'$.

□

Preuve de la propriété 3

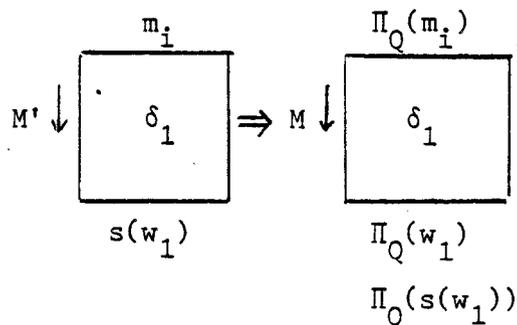
Cette preuve ne comporte pas de difficultés majeures et s'explique parfaitement sur un schéma :

Point 1 :

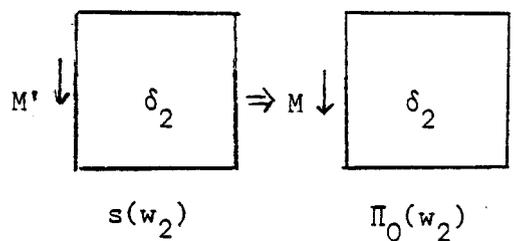
Soit $\delta \in \hat{DP}(\Sigma)$, $m_i \in FG(I')$ et $m_f \in FG(F')$ tels que

$$m_i \cdot \delta \stackrel{*}{\underset{M'}{\parallel}} \delta \cdot m_f.$$

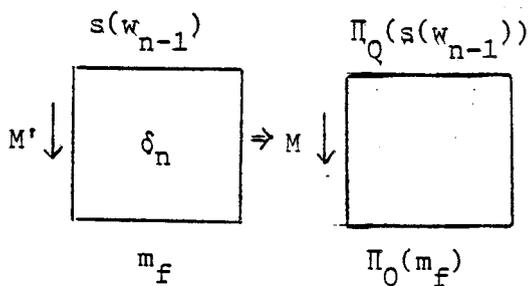
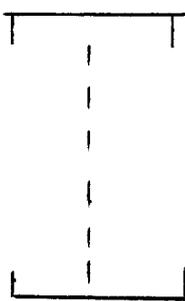
La reconnaissance de δ , par l'automate M' peut être décomposée la façon suivante :



et $\Pi_Q(m_i) \in FG(I)$ et $\Pi_Q(w_1) \in FG(F)$
(car v_0, k_0 sont conservés à chaque transition).



et $\Pi_Q(s(w_1)) \in FG(I)$ et $\Pi_Q(w_2) \in FG(F)$
(car v_0, k_0 sont conservés à chaque transition).



et $\Pi_Q(s(w_{n-1})) \in FG(I)$ et $\Pi_Q(m_f) \in FG(F)$
(car v_0, k_0 sont conservés à chaque transition).

On peut donc appliquer à chaque δ_i , la propriété 3 point 1 vérifiée par M :

$$\forall t \in T^* \quad \tau(\delta_i) \otimes e_t \in I(\text{exp})$$

$$\Rightarrow \forall t \in T^* \quad (\tau(\delta_1) \otimes e_t) \cdot (\tau(\delta_2) \otimes e_t) \dots \cdot (\tau(\delta_n) \otimes e_t) \in I(\text{exp}^*)$$

$$\Rightarrow \forall t \in T^* \quad ((\tau(\delta_1) \cdot \tau(\delta_2) \dots \cdot \tau(\delta_n)) \otimes e_t) \in I(\text{exp}^*)$$

$$\Rightarrow \forall t \in T^* \quad (\tau(\delta) \otimes e_t) \in I(\text{exp}^*)$$

□

Les points 2 et 3 n'appellent pas de commentaires particuliers et se démontrent de façon rigoureusement similaire.

□

Ceci clos la démonstration du lemme VI.3.1.

□

VI.3.2. - Démonstration de l'inclusion de RAT dans REC

Soit une expression rationnelle de dags (I, exp, F) où

- I est un langage rationnel de T^*
- exp est une expression simple de dags sur (Σ, T)
- F est un langage rationnel de T^*

Nous allons prouver que l'interprétation de cette expression rationnelle, ou encore les dags de $\hat{D}\mathcal{P}(\Sigma)$ obtenus par détypage des éléments de $I(\text{exp})$ dont la co-signature est un mot de I, et la signature un mot de F, est un taillis reconnaissable.

Le lemme VI.3.1. nous assure l'existence d'un automate d'états finis de dags : $M_1 = \langle \Sigma, Q_1, I_1, F_1, R_1 \rangle$ tel que :

$$\forall \delta \in T(M_1) \quad \tau(\delta) \in I(\text{exp})$$

et

$$\forall \delta' \in I(\text{exp}) \quad \exists \delta \in T(M_1) \text{ tel que } \tau(\delta) = \delta'.$$

L'automate $M' = \langle \Sigma, Q', I', F', R' \rangle$ où

$$Q' = Q_1$$

$$I' = I_1 \cap \Pi_T^{-1}(I)$$

$$F' = F_1 \cap \Pi_T^{-1}(F)$$

$$R' = R_1$$

reconnait de la même manière, les mêmes éléments de $\overset{\vee}{DP}(\Sigma)$ à condition toutefois que la projection sur T du contrôle haut soit un mot de I , et que la projection sur T du contrôle bas soit un mot de F .

Les propriétés mises en évidence par le lemme VI.3.1, nous permettent d'affirmer que :

- $\forall \delta \in T(M')$ $\tau(\delta) \in I(\text{exp})$; la co-signature de $\tau(\delta) \in I$ et la signature de $\tau(\delta) \in F$, c'est-à-dire $\bar{\tau}(\tau(\delta)) = \delta \in (I, \text{exp}, F)$.

ou encore :

$$T(M') \subset (I, \text{exp}, F)$$

- $\forall \delta \in (I, \text{exp}, F)$ $\exists \delta' \in I(\text{exp})$ de co-signature appartenant à I et de signature appartenant à F tel que $\bar{\tau}(\delta') = \delta$.

Ce qui nous autorise à conclure que $\delta \in T(M')$ ou encore

$$(I, \text{exp}, F) \subset T(M')$$

L'automate M' reconnaît exactement (I, exp, F) , cet ensemble de dags est donc un taillis reconnaissable.

□

VI.2.4. - Démonstration de l'inclusion de REC dans RAT

La similitude des rôles assumés par les types dans un magmoïde typé et les états dans un automate d'états finis déjà notée lors du paragraphe V.3.2, est la base de cette démonstration.

Soit $M = (\Sigma, Q, I, F, R)$ automate d'états finis considérons l'alphabet bitypé (Σ, Q) tel que :

$$(q_1 q_2 \dots q_n, a, q'_1 q'_2 \dots q'_m) \in (\Sigma, Q) \text{ ssi}$$

$$(q_1 \times q_2 \dots \times q_n) \cdot a \vdash a \cdot (q'_1 \times q'_2 \times \dots \times q'_m) \in R$$

Il est alors évident que $(I, ((\Sigma, Q)^{\circledast})^*, F)$ est égal au taillis reconnu par M . En effet :

$$((\Sigma, Q)^{\circledast})^* = \overset{\vee}{\text{DP}}(\Sigma, Q)$$

et le détypage des éléments de $\overset{\vee}{\text{DP}}(\Sigma, Q)$ dont la signature appartient à F et la co-signature à I coïncide bien avec les éléments reconnus par M , les types constituant en quelque sorte la trace, ou la mémorisation de la reconnaissance dans M

Exemple :

Le taillis reconnu par $M = (\Sigma, Q, I, F, R)$ où

$$\Sigma = \Sigma_2^2 \cup \Sigma_1^1 \text{ et } \Sigma_2^2 = \{a\} \text{ et } \Sigma_1^1 = \{b, c\}$$

$$Q = \{q_0, q_1, q_2, q_3\}$$

$$I = q_0 \otimes q_0$$

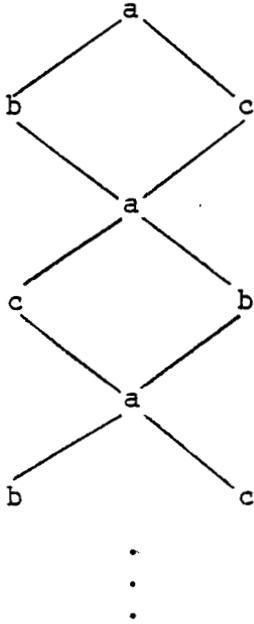
$$F = q_0 \otimes q_0$$

$$R = \{(q_0 \otimes q_0) \cdot a \vdash a \cdot (q_1 \otimes q_2) ; (q_3 \otimes q_3) \cdot a \vdash a \cdot (q_2 \otimes q_1) ; \\ q_1 \cdot b \vdash b \cdot q_0 ; q_1 \cdot b \vdash b \cdot q_3 ; q_2 \cdot c \vdash c \cdot q_0 ; q_2 \cdot c \vdash c \cdot q_3\}$$

peut être décrit par l'expression rationnelle :

$$\text{exp}_0 = (q_0 \otimes q_0, ((a_{q_1 q_2}^{q_0 q_0} + a_{q_2 q_1}^{q_3 q_3} + b_{q_0}^{q_1} + b_{q_3}^{q_1} + c_{q_0}^{q_2} + c_{q_3}^{q_2})^*, q_0 \otimes q_0)$$

ce qui représente encore les dags de la forme :



; dags où b et c change alternativement de place dans la descendance immédiate de a.

Remarque :

Les expressions rationnelles suivantes décrivent le même sous-ensemble de dags de $\hat{DP}(\Sigma)$, et réclame un nombre moins important de types distincts.

$$\text{exp}_1 = (q_0 \otimes q_0, (a_{q_2 q_2}^{q_0 q_0} \cdot (b_{q_1}^{q_2} \otimes c_{q_1}^{q_2}) + a_{q_2 q_2}^{q_1 q_1} \cdot (c_{q_0}^{q_2} \otimes b_{q_0}^{q_2}))^*, q_0 \otimes q_0)$$

qui ne nécessite pas trois types distincts.

$$\text{exp}_2 = (q_0 \otimes q_0, (a_{q_1 q_1}^{q_0 q_0} \cdot (b_{q_0}^{q_1} \otimes c_{q_0}^{q_1}) \cdot a_{q_1 q_1}^{q_0 q_0} \cdot (c_{q_0}^{q_1} \otimes b_{q_0}^{q_1}))^*, q_0 \otimes q_0)$$

qui ne nécessite que deux types distincts.

Alors que l'exp₀ ne nous renseigne guère sur "l'allure" des dags qu'elle représentent, l'exp₁ et l'exp₂ évoque de façon plus précise la "forme" des éléments dont elles sont la description synthétique. On peut donc penser que la réduction du nombre de types enrichit la description que constitue l'expression rationnelle. Nos études ne nous ont pas permis de mettre au point un procédé effectif de réduction du nombre de types nécessaires à l'élaboration d'une expression rationnelle, et l'existence d'un tel algorithme nous paraît très incertaine.

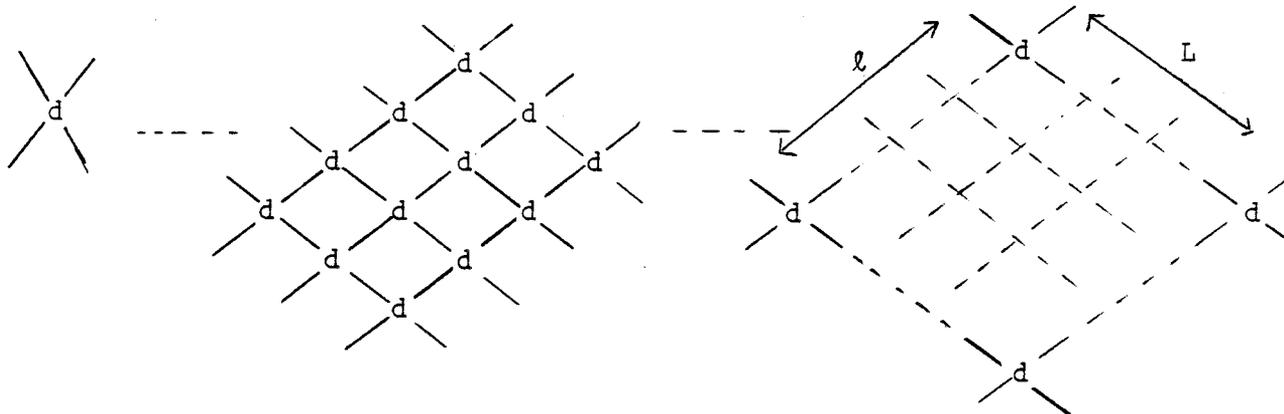
VII.3. - CONCLUSION

Cette longue et fastidieuse démonstration atteste, pour le moins, de la distance qui sépare la caractérisation d'un taillis reconnaissable par l'existence d'un automate qui reconnaît ces éléments, de celle qui consiste à exhiber une expression rationnelle qui décrive ces éléments.

On peut, par exemple, établir que l'ensemble des dags possédant un motif connexe donné est rationnel, attendu que cette spécification se traduit de façon simple sous forme d'expression rationnelle : $(T^*, \hat{D}\hat{P}(\Sigma, T) \cdot \text{motif} \cdot \hat{D}\hat{P}(\Sigma, T))$. La recherche de l'automate de dags planaires capable de reconnaître ce type de dags, et uniquement ceux-là, réclamerait un effort considérable de combinatoire. Inversement, pour d'autres taillis reconnaissables, il sera plus évident de découvrir un automate d'états finis qui le reconnaisse que de produire une expression rationnelle non réductible directement à un automate d'états finis.

Exemple :

Soit $\Sigma = \Sigma_2^2 = \{d\}$, et l'ensemble de dags suivants :



est reconnu par l'automate $M = \langle \Sigma, Q, I, F, R \rangle$ où

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

$$I = q_0^* \otimes q_1^*$$

$$F = q_1^* \otimes q_0^*$$

$$R = \{(q_0 \otimes q_1 \cdot d \mid d \cdot (q_1 \otimes q_0))\}.$$

L'élément fondamental de cet automate est la règle de R , et on la retrouvera d'une façon ou d'une autre dans une expression rationnelle.

Ces deux manières de caractériser les taillis reconnaissables sont donc complémentaires, et nous offre deux méthodes d'approche très différentes à l'égard des problèmes de rationalité dans les dags.

Le second apport de cette démonstration est de nous avoir obligé à décortiquer les mécanismes de fonctionnement de l'opération $*$ sur les expressions rationnelles, ce qui nous permet de dégager le rôle capital joué par les parties connexes d'un dag composé.

Mettre à l'étoile une expression simple de dag équivaut à :

- assembler, en respectant les concordances de types, les "blocs" connexes qui sont représentés par l'expression simple,
- en respectant uniquement la place des blocs droit et gauche qui devront obligatoirement se situer sur les bords droit et gauche des dags construits.

Exemple :

Soit l'expression simple de dag suivante : $\text{exp} = (a_t^t \otimes b_t^t \otimes c_t^t \otimes d_t^t)^*$
 les blocs connexes sont ici : $a_t^t, b_t^t, c_t^t, d_t^t$
 le bloc droit est : a_t^t
 le bloc gauche est : d_t^t

Tous les dags composés de a, b, c ou d pourvu que les occurrences de la lettre a se trouvent sur le bord droit et les occurrences de la lettre d se trou-

vent sur le bord gauche, appartiennent à l'interprétation de cette expression simple de dags.

Exemple :

a	b	b	d	∈ I(exp), on peut en effet considéré ce dag
c	b	c	b	

comme :

a	b	c	d	→ ∈ I(a ⊗ b ⊗ c ⊗ d)
	b	c		→ ∈ I(a ⊗ b ⊗ c ⊗ d)
c				→ ∈ I(a ⊗ b ⊗ c ⊗ d)
			b	→ ∈ I(a ⊗ b ⊗ c ⊗ d)

Par contre $\begin{array}{|} \hline a \\ \hline \end{array}$ n'appartient pas à I(exp), une occurrence de la lettre a étant placée sur un bord droit.

BIBLIOGRAPHIE

- [1] A. ARNOLD, M. DAUCHET
"Théorie des magmoïdes (1)".
 RAIRO Informatique Théorique, 1979, pp. 135-154.
- [2] A. ARNOLD, M. DAUCHET
"Théorie des magmoïdes (2)".
 RAIRO Informatique Théorique, 1979, pp. 135-154.
- [3] M. NIVAT
"Compilation et interprétation".
 Actes de la Dixième Ecole de Printemps, Barèges, 1982, LITP.
- [4] B. COURCELLE, P. FRANCHI-ZANNETTI
"On the expressive power of attribute grammars".
 Proc. of 21 St. IEEE Symp. of Foundations of Computer Science, Act. 1980,
 pp. 39-59.
- [5] B. LORHO
"Semantic attributes processing in the systel delta".
 Methods of Algorithmic Languages Implementation
- [6] M.C. GAUDEL
"Correctness proof of programming language translations".
 IFIP TC2 Working Conference, Garmish Parten Kirschen, 1977, Rapport
 GRECO Programmation N° 013-81.
- [7] D. KAHN
"Editeurs dirigés par la syntaxe".
 Ecole INRIA.
- [8] J.A. GODEN, J.W. THATCHER, E.G. WAGNER
*"An initial algebra approach to the specification, correctness and imple-
 mentation of abstract data types current trends in programming methodology"*.
 Rapport Technique YEH (Editeur), Volume IV, Prentice-Hall, 1978.

- [9] C. PAIR, M.C. GAUDEL
"Les Structures de données et leur représentation en mémoire".
INRIA, 1979, ISBN 2-7261-0229-8.
- [10] A.V. AHO, J.D. ULMAN
"Principles of compiler design".
ISBN 0-201-00022-9.
- [11] G. COMYN, M. DAUCHET
"Metric approximations in order domains".
Algebraic Methods in Semantics, Editeurs J.C. REYNOLD et M. NIVAT. Cambridge University Press, 1975.
- [12] A. ARNOLD, M. DAUCHET
"Morphismes et bimorphismes d'arbres"
Theoretical Computer Science, 1982, pp. 33-93.
- [13] J. ENGELFRIET
"A hierarchy of tree transducteurs".
Proc. 3^{ème} Colloque sur les arbres en algèbre et programmation,
Université de Lille 1, 1978, pp. 103-106.
- [14] W.C. ROUNDS
"Context-free grammars on tree".
Proc. 1st ACM Conference of Theory of Comput, 1969, pp. 143-148.
- [15] J.W. THATCHER
"Tree-automata : an informal survey".
Currents in the Theory of Computing, Prentice-Hall, Englewood Cliffs,
1973, pp. 142-178.
- [16] D. JANSSENS, G. ROZENBERG
"A survey of NLC grammars".
Lecture in Notes in Computer Sciences, Volume 159, pp. 114-128.

- [17] RAOULT
 "Sur le traitement opératoire des graphes fonctionnels".
 Thèse d'Etat, 1983, Université de Paris-Sud, Centre d'Orsay
- [18] V. CLAUS, H. EHRIG, G. ROZENBERG
 "Graph grammars and their applications to computer sciences and biology".
 Lecture in Notes in Computer Sciences, Volume 73, 1979.
- [19] H. EHRIG, M. PFENDER, H.J. SCHNEIDER
 "Graph grammars and their applications to computer science".
 Lecture in Notes in Computer Sciences, Volume 153, 1983, ISBN 3-540-12310-5,
 ISBN 0-387-123210-5 (Anglais).
- [20] B. COURCELLE
 "Equivalence and transformation of regular system : application to
 recursive program schemes".
 Rapport Technique n° 8430, Université de Bordeaux, 1984.
- [21] B. COURCELLE
 "Graphs expressions and graphs rewriting".
 A paraître, CAAP 1986.
- [22] S. EILENBERG, J.B. WRIGHT
 "Automata and tree grammars".
 Information and Control 11, 1967, pp. 217-231.
- [23] H. EHRIG, M. PFENDER, H.J. SCHNEIDER
 "Graph grammars : an algebraic approach".
 Proc. 14th Ann. IEEE Symposium on Switching and Automata Theory, IOAWA CITY,
 1973, pp. 167-180.
- [24] W. PATERSON
 "Linear unification".
 JCSS, Volume 16, 1978, pp. 158-167.

- [25] FAGES
"Unification. Notes sur l'unification des termes du premier ordre fini ou infini".
Rapport LITP
- [26] MARTELLI, MOUTANARI
"An efficient unification algorithm".
Transactions on programming languages and syst., 1982, pp. 256-282.
- [27] ARBIB, MANES
Graph Grammars, 1973.
- [28] D.F. STANAT
"A homomorphism theorem for weighted context-free grammars".
Journal of Computer and Systems Sciences, Volume 6.
- [29] J.E. HOPCROFT, J. ULLMAN
"Introduction to automata theory, languages and computation".
Editeur Addison-Wesley.
- [30] BEERSTEL
"Transductions and context-free languages".
1979.
- [31] W.S. BRAININ
"Tree generating regular systems".
Information and Control 14, 1969, pp. 217-231.
- [32] S. EILENBERG
"Automata, languages and machines".
New-York, 1974.

- [33] J. ENGELFRIET
"Tree automata and tree grammars".
DAIMI Report FN-10, University of Aarhus, Denmark, 1975.
- [34] T.S.E. MAIBAUM
"A generalised approach to formal languages".
Journal Computer System Sciences 8, 1974, pp. 402-432.
- [35] W.C. ROUNDS
"Context-free grammars on trees".
1st ACM Proc. on Theory of Computing, 1969, pp. 257-287.
- [36] W.C. ROUNDS
"Mappings and grammars on trees".
Math. Systems Theory 4, 1970, pp. 257-287.
- [37] D. PERRIN
"Le déterminisme en informatique".
Actes du Colloque : Informatique et connaissances, CIRSH, 1983.
- [38] M. NIVAT
"Transductions des langages de Chowsky".
Annales de l'Institut Fourier de Grenoble, 18, 1968, pp. 339-345.
- [39] G. HUET
"Confluent reductions : abstract properties an applications to term rewriting systems".
JACM, Volume 27, n° 4, 1980, pp. 797-821.
- [40] D. CORBEL
"Schéma de câblage et schéma de contrôle. Application à la simulation et à la gestion des processus industriels".
Thèse de 3^{ème} Cycle, Lille 1, 1979.

- [42] N. CARIDROIT
"Automates d'états finis de dags planaires".
Mémoire de DEA, 1985, Université de Lille 1.
- [43] M. NIVAT
"Langages algébriques sur le magma libre et sémantique des schémas de programme".
Automata, Languages and programming, 1st Colloquium, Editeur M. NIVAT,
Springer-Verlag, 1973, pp. 293-307.
- [44] A. ARNOLD
"Systèmes d'équations dans le magmaïde. Ensembles rationnels et algébriques d'arbres".
Thèse d'Etat, Lille, 1977.
- [45] G. COUSINEAU
"Langages d'interprétation des schémas récursifs".
RAIRO - Informatique Computer Science, 1975, Volume 1.
- [46] M. DAUCHET
"Transductions de forêts. Bimorphismes de magmaïdes".
Thèse d'Etat, Lille, 1977.

