N° d'ordre : 866

UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR DE 3ème CYCLE

(Traitement de l'Information)

par

Bernard LEGUY

REDUCTIONS, TRANSFORMATIONS ET CLASSIFICATION DES GRAMMAIRES ALGEBRIQUES D'ARBRES

Thèse soutenue le jeudi 4 décembre 1980 devant la Commission d'Examen

Membres du Jury	MM. G. JACOB	Président
	M. DAUCHET	Rapporteur
	A. ARNOLD	Examinateur
	M. NIVAT	Examinateur

P R O F E S S E U R S 1ERE C L A S S E

M. BACCHUS Pierre	Mathématiques
M. BEAUFILS Jean Pierre (dét.)	Chimie
M. BIAYS Pierre	G.A.S.
M. BILLARD Jean	Physique
M. BONNOT Ernest	Biologie
M. BOUGHON Pierre	Mathématiques
M. BOURIQUET Robert	Biologie
M. CELET Paul	Sciences de la Terre
M. COEURE Gérard	Mathématiques
M. CONSTANT Eugène	I.E.E.A.
M. CORDONNIER Vincent	I.E.E.A.
M. DEBOURSE Jean Pierre	S.E.S.
M. DELATTRE Charles	Sciences de la Terre
M. DURCHON Maurice	Biologie
M. ESCAIG Bertrand	Physique
M. FAURE Robert	Mathématiques
M. FOURET René	Physique
M. GABILLARD Robert	I.E.E.A.
M. GRANELLE Jean Jacques	S.E.S.
M. GRUSON Laurent	Mathématiques
M. GUILLAUME Jean	Biologie
M. HECTOR Joseph	Mathématiques
M. HEUBEL Joseph	Chimie
M. LABLACHE COMBIER Alain	Chimie
M. LACOSTE Louis	Biologie
M. LANSRAUX Guy	Physique
M. LAVEINE Jean Pierre	Sciences de la Terre
M. LEBRUN André	C.U.E.E.P.
M. LEHMANN Daniel	Mathématiques
Mme LENOBLE Jacqueline	Physique
M. LHOMME Jean	Chimie
M. LOMBARD Jacques	S.E.S.
M. LOUCHEUX Claude	Chimie
M. LUCQUIN Michel	Chimie
M. MAILLET Pierre	S.E.S.
M. MONTREUIL Jean	Biologie
M. PAQUET Jacques	Sciences de la Terre
M. PARREAU Michel	Mathématiques
M. PROUVOST Jean	Sciences de la Terre

.../...

Professeurs 1ère classe (suite)

M. SALMER Georges	I.E.E.A.
Mme SCHWARTZ Marie Hélène	Mathématiques
M. SEGUIER Guy	I.E.E.A.
M. STANKIEWICZ François	Sciences Economiques
M TILLIEU Jacques	Physique
M. TRIDOT Gabriel	Chimie
M. VIDAL Pierre	I.E.E.A.
M. VIVIER Emile	Biologie
M. WERTHEIMER Raymond	Physique
M. ZEYTOUNIAN Radyadour	Mathématiques

PROFESSEURS 2ème CLASSE

M. AL FAKIR Sabah	Mathématiques
M. ANTOINE Philippe	Mathématiques
M. BART André	Biologie
Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mathématiques
M. BELLET Jean	Physique
M. BKOUCHE Rudolphe	Mathématiques
M. BOBE Bernard	S.E.S.
M. BODART Marcel	Biologie
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean Pierre	Chimie
M. BOSQ Denis	Mathématiques
M. BREZINSKI Claude	I.E.E.A.
M. BRUYELLE Pierre (Chargé d"enseignement)	Géographie
M. CAPURON Alfred	Biologie
M. CARREZ Christian	I.E.E.A.
M. CHAMLEY Hervé	E.U.D.I.L.
M. CHAPOTON Alain	C.U.E.E.P.
M. COQUERY Jean Marie	Biologie
Mme CORSIN Paule	Sciences de la Terre
M. CORTOIS Jean	Physique
M. COUTURIER Daniel	Chimie
Mle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	E.U.D.I.L.
M. DEGAUQUE Pierre	I.E.E.A.
M. DELORME Pierre	Biologie
M. DEMUNTER Paul	C.U.E.E.P.
M. DE PARIS Jean-Claude	Mathématiques

.../...

Professeurs 2ème classe (suite)

M. PERROT Pierre	Chimie
M. PERTUZON Emile	Biologie
M. PONSOLLE Louis	Chimie
M. PORCHET Maurice	Biologie
M. POVY Lucien	E.U.D.I.L
M. RACZY Ladislas	I.E.E.A.
M. RICHARD Alain	Biologie
M. RIETSCH François	E.U.D.I.L.
M. ROGALSKI Marc	M.P.A.
M. ROUSSEAU Jean-Paul	Biologie
M. ROY Jean-Claude	Biologie
M. SALAMA Pierre	S.E.S.
Mme SCHWARZBACH Yvette (CCP)	M.P.A.
M. SCHAMPS Joël	Physique
M. SIMON Michel	S.E.S.
M. SLIWA Henri	Chimie
M. SOMME Jean	G.A.S.
Mle SPIK Geneviève	Biologie
M. STERBOUL François	E.U.D.I.L.
M. TAILLIEZ Roger	Institut Agricole
M. TOULOTTE Jean-Marc	I.E.E.A.
M. VANDORPE Bernard	E.U.D.I.L.
M. WALLART Francis	Chimie
M. WATERLOT Michel	Sciences de la Terre
Mme ZINN JUSTIN Nicole	M.P.A.

CHARGES DE COURS

M. TOP Géard	S.E.S.
M. ADAM Michel	S.E.S.

CHARGES DE CONFERENCES

M. DUVEAU Jacques	S.E.S.
M. HOFLACK Jacoues	I.P.A.
M. LATOUCHE Serge	S.E.S.
M. MALAUSSENA DE PERNO Jean-Louis	S.E.S.
M. OPIGEZ Philippe	S.E.S.

M. DEVRAINNE Pierre	Chimie
M. DHAINAUT André	Biologie
M. DORMARD Serge	S.E.S.
M. DOUKHAN Jean Claude	E.U.D.I.L.
M. DUBOIS Henri	Physique
M. DUBRULLE Alain	Physique
M. DUEE Gérard	Sciences de la Terre
M. DYMENT Arthur	Mathématiques
M. FLAMME Jean Marie	E.U.D.I.L.
M. FONTAINE Hubert	Physique
M. GERVAIS Michel	S.E.S.
M. GOBLOT Rémi	Mathématiques
M. GOSSELIN Gavriel	S.E.S.
M. GOUDMAND Pierre	Chimie
M. GREVET Patrice	S.E.S.
M. GUILBAULT Pierre	Biologie
M. HANGAN Théodor	Mathématiques
M. HERMAN Maurice	Physique
M. JACOB Gérard	I.E.E.A.
M. JACOB Pierre	Mathématiques
M. JOURNEL Gérard	E.U.D.I.L.
M. KREMBEL Jean	Biologie
M. LAURENT François	I.E.E.A.
Mlle LEGRAND Denise	Mathématiques
Mlle LEGRAND Solange	Mathématiques (Calais)
Mme LEHMANN Josiane	Mathématiques
M. LEMAIRE Jean	Physique
M. LENTACKER Firmin	G.A.S.
M. LEVASSEUR Michel	I.P.A.
M. LHENAFF René	G.A.S.
M. LOCQUENEUX Robert	Physique
M. LOSFELD Joseph	I.E.E.A.
M. LOUAGE Francis	E.U.D.I.L.
M. MACKE Bruno	Physique
M. MAIZIERES Christian	I.E.E.A.
Mlle MARQUET Simone	Mathématiques
M. MESSELYN Jean	Physique
M. MIGEON Michel	E.U.D.I.L.
M. MIGNOT Fulbert	Mathématiques
M. MONTEL Marc	Physique
Mme NGUYEN VAN CHI Régine	G.A.S.
M. PARSY Fernand	Mathématiques
Mlle PAUPARDIN Colette	Biologie

Je remercie vivement Gérard JACOB de me faire l'honneur et l'amitié de présider le jury.

André ARNOLD m'a initié à cette spécialité difficile, je le remercie de me l'avoir fait apprécier. Je le remercie aussi pour les nombreux conseils, orientations et indications qu'il m'a toujours donnés volontiers malgré l'éloignement.

Max DAUCHET a eu l'amabilité de s'intéresser à mes travaux en cours et de me conseiller efficacement tant sur des problèmes de fond que de rédaction. Je ne saurais trop le remercier de son aide précieuse.

Maurice NIVAT par ses travaux personnels a profondément influencé tous les chercheurs dans ce domaine et particulièrement ceux de notre équipe. Je le remercie pour l'honneur qu'il me fait de s'intéresser à mes travaux et de participer au jury.

Bien des membres du laboratoire devraient aussi être remerciés pour avoir su chacun à leur manière m'encourager à cette recherche en informatique théorique à laquelle mon passé de technicien ne me prédisposait pas. Cette recherche m'a aussi conduit vers l'équipe d'informatique théorique dont le climat amical et joyeux, outre la compétence de ses membres, est un réconfort permanent.

Je tiens à remercier le secrétariat du laboratoire et particulièrement Madame Patricia CARON, Madame et Monsieur DEBOCK qui ont su assurer la réalisation matérielle de cette thèse avec la compétence et la gentillesse dont ils ne se départissent jamais.

A tous les miens.

Gwez kalonek va bro,
Fao, kistin ha derv,
Ganeoc'h e vin bepred
A gorf hag a spered

Youenn GWERNIG.

Vaillants arbres de mon pays,
Hêtres, chataîgners et chênes,
Je serai toujours avec vous
Corps et âme.

Table des Matières

* * *

RÉSUMÉ

Nous étudions les grammaires algébriques d'arbres, c'est à dire les schémas de programmes non déterministes, leurs transformations et leurs réductions.

Nous montrons qu'il est en général impossible de réduire la non strictitude et la non complétude dans les grammaires algébriques OI d'arbres et nous en déduisons une hiérarchie dans les forêts algébriques OI. Par contre nous donnons des constructions permettant de réduire la non strictitude et la non complétude dans les grammaires algébriques IO. Nous étudions aussi quelques autres réductions ou transformations de grammaires d'arbres et la non fermeture par homomorphisme linéaire inverse de la classe des forêts algébriques OI.

INTRODUCTION GENERALE

Le monde scientifique reconnaît dans l'arbre un modèle extrêmement important et l'a idéalisé en un objet mathématique. L'arbre est alors le schéma usuel des notions de classification, de décomposition ou de hiérarchie. Il représente la démarche qui relie le général au particulier, l'essentiel à la description détaillée. Un fait remarquable est que cette structure d'arbre, suffisamment puissante pour servir de charpente à des algorithmes très sophistiqués, soit suffisamment bien adaptée à nos structures mentales pour figurer dans des livres élémentaires de grammaire et de mathématiques destinés aux jeunes enfants.

Voici un exemple du genre d'arbres auxquels nous nous intéressons :

```
          +
        /   \
       1     si
          /   |   \
        et    +    0
       / \    |\
      <   <   *  5
     /\  /\   |\
     A B B C  3 X
```

Ceci peut être vu comme une représentation de l'expression Algol :
"1 + si A < B et B < C alors 3 * X + 5 sinon 0".
Les noeuds de cet arbre sont étiquetés par des symboles représentant des opérateurs ou des valeurs. Nous dirons que ce sont des symboles d'un alphabet gradué car chacun possède un degré qui dans l'exemple s'identifie au nombre d'opérandes (le degré de "+" est 2, degré de "si" est 3 et degré de "5" est 0). Généralement les arbres ne sont pas considérés isolément mais comme éléments d'ensembles d'arbres ou forêts définis souvent par une propriété commune : forêt des arbres de dérivation dans une grammaire, forêt des arbres AVL ...

Comme en théorie des langages, on décrira en termes finis des forêts infinies en utilisant des systèmes de génération par dérivation : les grammaires. Formellement la notion de grammaire d'arbres généralise de façon évidente celle de grammaire de mots et on obtient comme dans le cas des langages de mots, une première classification suivant la forme des parties gauches des règles :

- les grammaires régulières dont les règles sont de la forme $X \to t$ où X est un symbole de degré 0 et t un arbre. Elles ont été les premières étudiées [Br] [Ro1] [Mar] [St] en particulier pour l'étude des arbres de dérivation dans les grammaires de mots [P] [Th].

- les grammaires algébriques introduites par Fischer [Fi] dont les règles sont de la forme

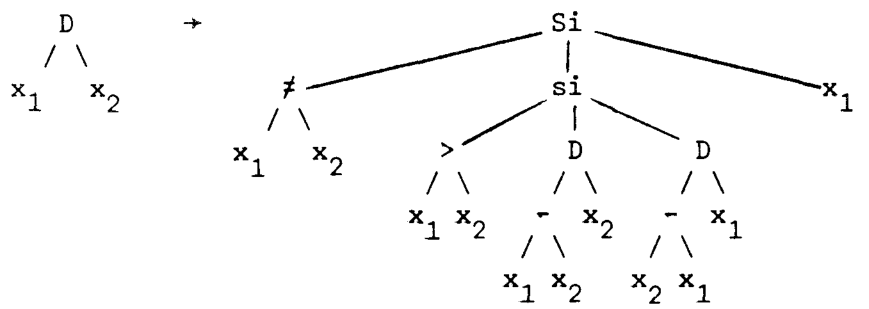
$$\begin{array}{c} X \\ / \mid \backslash \\ x_1\ x_2 \ldots x_n \end{array} \to t$$

où X est un symbole non terminal et t un arbre construit à l'aide de symboles terminaux ou non et de paramètres de formels ou variables de degré 0 apparaissant sous X en partie gauche.

- Les grammaires générales qui engendrent les forêts récursivement énumérables [AD-r].

Nous nous intéressons aux grammaires algébriques dont l'intérêt n'est plus à démontrer pour l'étude des schémas de programme [E], [BO2], [N], [CO3], [A-te], [Da]. Nous avons fait cette étude dans un cadre formel sûr et précis, celui de la théorie des magmoïdes introduite par Arnold et Dauchet [ADm1] [ADm2].

Afin d'éviter d'être trop abstraits, nous interpréterons les grammaires algébriques comme des schémas de programmes. Chaque symbole terminal de degré n pourra être vu comme un opérateur à n arguments, chaque symbole non terminal de degré n comme une procédure à n paramètres et la règle :



pourra être vue comme une représentation de la fonction :

$$D(x_1, x_2) = \text{si } x_1 \neq x_2 \text{ alors si } x_1 > x_2 \text{ alors } D(x_1 - x_2, x_2)$$
$$\text{sinon } D(x_2 - x_1, x_1)$$
$$\text{sinon } x_1.$$

Dériver dans une telle grammaire revient alors à expliciter formellement un calcul en réalisant des appels de fonctions par substitution du corps de la fonction.

D'une part pour les mêmes raisons qu'en théorie des langages et d'autre part pour améliorer l'efficacité des schémas de programmes, nous nous posons certains problèmes de réduction de grammaires. Ces problèmes ont été pour la plupart étudiés et résolus dans le cas des schémas de programmes déterministes par I. Guessarian [Gu]. Nous nous intéresserons au cas plus général des non déterministes.

Considérons les règles qui sont de la forme $\begin{array}{c} X \\ / \backslash \\ x_1 \ldots x_n \end{array} \to x_i$, c'est à dire où la fonction X peut être une simple projection. Ces règles non strictes sont la généralisation des règles effaçantes dans les grammaires de mots. Nous montrons au terme d'une preuve très technique que cette réduction est en général impossible : il n'existe pas en général de schéma de programme équivalent à un schéma de programme donné et où aucune fonction ne puisse se résumer à la projection d'un de ses paramètres.

Une autre règle peu efficace est une règle non complète
$$\begin{array}{cl} X & \to t \\ / \ \backslash & \\ x_1 \ldots x_n & \end{array}$$
où l'un des paramètres, x_i, n'apparait pas dans t.
On peut encore exprimer cela en disant que la fonction X n'utilise pas toujours tous ses paramètres. Nous montrons que la non complètude n'est pas toujours réductible et qu'il existe des forêts algébriques pour lesquelles il existe une grammaire stricte et une grammaire complète mais pas de grammaire à la fois complète et stricte.

Il est facile de voir qu'une règle non stricte est aussi non complète si le non terminal en partie gauche est de degré supérieur à 1. Par contre une règle non stricte s'appliquant à un symbole d'arité 1 est complète, cette non strictitude monadique qui est en quelque sorte la non strictitude pure est réductible et nous donnons une construction qui permet de la réduire. Il en ressort aussi que toute forêt engendrée par une grammaire complète peut être engendrée par une grammaire stricte (mais cette nouvelle grammaire n'est plus complète en général).

Nous étudions aussi les grammaires linéaires (sans duplication), dans lesquelles toute fonction utilise chaque paramètre au plus une fois et nous montrons que la non complètude et la non strictitude sont réductibles dans cette classe de grammaires. Alors en notant Reg, Lin, CS, C, S, Alg les classes de forêts algébriques engendrées par les grammaires respectivement régulières, linéaires, à règles complètes et strictes, à règles complètes, à règles strictes, algébriques quelconques, nous obtenons la hiérarchie :

$$\mathrm{Reg} \subsetneq \mathrm{Lin} \subsetneq \mathrm{CS} \subsetneq \mathrm{C} \subsetneq \mathrm{S} \subsetneq \mathrm{Alg}.$$

Etant donnée une grammaire G on l'habitude de distinguer deux modes de dérivation particuliers. D'une part les dérivations ascendantes ou IO pour "Innermost-Outermost" indiquant que dans un arbre écrit en notation fonctionnelle parenthésée, on commence par dériver ce qui est à l'intérieur des parenthèses avant de dériver l'extérieur, ce qui correspond à la notion d'appel par valeur, les paramètres étant évalués d'abord. On note $F_{IO}(G)$ la forêt engendrée en IO par G. A l'inverse on définit les dérivations descendantes ou encore OI correspondant à l'appel par nom et on note $F_{OI}(G)$ la forêt engendrée en OI par G. Il est bien connu que, en notant F(G) la forêt engendrée par G sans contraintes sur les dérivations, on a $F_{IO}(G) \subset F_{OI}(G) = F(G)$. Ainsi pour la grammaire G dont les règles sont :

$P \to Q(N(0))$

$Q(x_1) \to *(x_1, x_1)$

$N(x_2) \to s(N(x_1))$ ou $N(x_1) \to 0$

on a $F_{IO}(G) = \left\{ \begin{array}{c} * \\ / \quad \backslash \\ n\left\{\begin{array}{cc} s & s \\ | & | \\ s & s \\ \vdots & \vdots \\ s & s \end{array}\right\}n \\ \begin{array}{cc} | & | \\ 0 & 0 \end{array} \end{array} \text{ avec } n \in \mathbb{N} \right\} \subset F_{IO}(G) = F(G) = \left\{ \begin{array}{c} * \\ / \quad \backslash \\ n\left\{\begin{array}{cc} s & s \\ | & | \\ s & s \\ \vdots & \vdots \\ s & s \end{array}\right\}p \\ \begin{array}{cc} | & | \\ 0 & 0 \end{array} \end{array} \text{ avec } n, p \in \mathbb{N} \right\}.$

Nous montrons comment pour toute grammaire G, on peut construire une grammaire à la fois complète et stricte G' telle que $F_{IO}(G) = F_{IO}(G')$.

La différence des résultats obtenus en IO et en OI peut surprendre, aussi allons nous essayer d'expliquer les raisons de l'irréductibilité en OI. Les raisons de la réductibilité en IO seront alors assez claires. Dans une grammaire non complète, non stricte et non deterministe, on peut trouver un non terminal X à deux arguments (ou plus) auquel s'appliquent deux règles $X(x_1, x_2) \to x_1$ et $X(x_1, x_2) \to x_2$. On peut dire que X est une fonction de choix non déterministe. A l'aide d'une telle fonction il est aisé de constituer par dérivations un arbre portant comme sous arbres tous les arbres d'un ensemble fini mais non borné. Appelons prototype un tel arbre.

Par exemple, en partant de

```
  Y
 / \
0   0
```

, les règles

```
   Y     →     Y          et    Y     →    Z
  / \         / \              / \         |
x1   x2      X      s        x1   x2       *
            / \     |                     / \
          x1   s    x2                  x1   x2
               |
               x2
```

peuvent conduire par dérivation descendantes au prototype

```
                      *
                     / \
                    X   s
                   / \   \
                  X   s   s   n
                 / \   \   \
                    s   s   n
                X    \   \   .
               / \    s n-1 s
              0   s    \  .  \
                   \    .  s  0
                    0    s  \
                          \  0
                           0
```

et ce prototype peut à son tour se dériver en l'un quelconque des arbres de la forme

```
          *
         / \
        s   s
       /     \
   p  s       s   n
     /         \
    .           .
   s             s
  /               \
 0                 0
```

avec $p \le n$.

Si de plus ce prototype qui est surmonté du non terminal Z peut être dupliqué un nombre arbitrairement grand de fois, donc plus de n fois, par des règles du type

```
Z  →    Z       et  Z  →    b
|       |           |      / \
x1      b           x1   x1   x1
       / \
     x1   x1
```

, toute la puissance de choix mise dans le prototype pourra être effectivement utilisée. Réduire la non complétude et la non strictitude revient à réaliser ce choix non borné avant les duplications, nous montrons que c'est impossible dans le cas des dérivations OI. Par contre dans le cas des dérivations IO le prototype est nécessairement entièrement dérivé avant toute duplication, il n'y a alors plus de choix ultérieur ce qui explique que la non complétude et la non strictitude soient réductibles dans ce cas.

La preuve de la non réductibilité de la non strictitude et de la non complétude s'appuie sur la preuve de la necessité de construire de tels prototypes à choix non borné pour obtenir certaines forêts algébriques qui nous servent de contre exemples. A cette fin nous avons étudié les forêts balancées engendrées depuis le non terminal Z par les règles citées plus haut. Nous avons alors montré un lemme qui pour la largeur de ces arbres balancés est un peu ce que le lemme de l'étoile est à la longueur des mots. Ce lemme nous amène à définir pour certains noeuds autres que le sommet dans les arbres obtenus au cours des dérivations, des attributs que nous nommons cotes ou poids et qui sont invariants par dérivation. Cette démarche nous permet de montrer que l'image par homomorphisme linéaire inverse de la classe CS n'est pas incluse dans Alg, ce qui améliore un peu le résultat de Arnold et Dauchet [AD, ϕ-1].

Afin de ne pas être gênés par diverses contingences comme les non terminaux inutiles ou les paramètres inutiles, nous avons complété certaines réductions élémentaires qui avaient déjà été étudiées par divers auteurs. Nous définissons ainsi et étudions la OI -réduction qui est à peu près la réduction forte de Boudol [Bo] étudiée aussi partiellement dans [Co 3] et [AD-d]. Nous étudions la réduction de la non complétude et de la non strictitude dans les dérivations initiales définies par Arnold et Dauchet [AD, d]. Enfin nous définissons la IO-réduction qui est une propriété plus forte que la OI-réduction, Le chapitre I est consacré à introduire le cadre formel utilisé qui est celui de la théorie des Magmoïdes, et à donner quelques définitions et résultats de base sur les grammaires algébriques d'arbres.

Le chapitre II définit les OI et IO réductions et traite de la OI-réduction.

Le chapitre III traite des réductions conservant les forêts engendrées par dérivation IO.

Le chapitre IV expose d'abord la réduction de la non strictitude monadique et la réduction de la non strictitude et de la non complétude dans les dérivations initiales. Ensuite est faite l'étude des forêts balancées et les applications aux contre-exemples qui nous permettent d'obtenir nos résultats de non réduction et de non fermeture par homomorphisme linéaire inverse.

I - PRELIMINAIRES

Préliminaire informel

Dans la suite de ce travail il sera question d'arbres, de forêts et de grammaires algébriques définis formellement et certaines propriétés de ces objets formels seront montrées. L'étude de ces objets formels est si intéressante en elle-même qu'elle pourrait parfois nous faire oublier qu'ils n'ont pas été définis à seule fin de les étudier mais bien au contraire parce qu'ils s'avèrent être des modèles théoriques adéquats pour bon nombre de problèmes pratiques. Il serait parfaitement inutile de rappeler longuement tous les usages qui sont faits des arbres (organisation hiérarchisée, arbres de recherche ou de tri, arbres de dérivation, ...). On s'intéresse généralement à des ensembles d'arbres ou forêts (ensemble des arbres AVL, ensemble des arbres de dérivation d'un langage algébrique donné, ...) et un moyen classique de définir un ensemble est d'en donner un système fini de générateurs, c'est ce rôle que jouent les grammaires d'arbres.

Un exemple d'arbre tout à fait classique est celui que l'on associe à une expression de calcul, par exemple à l'expression (a+b) * c - d / e on a l'habitude d'associer l'arbre

```
          -
        /   \
      *       /
     / \     / \
    +   c   d   e
   / \
  a   b
```

dont les feuilles sont les paramètres de l'expression et les autres noeuds les opérateurs. On identifie couramment cet arbre à l'expression préfixée du calcul : -(*(+(a, b), c), / (d, e)), interprétée comme l'ordre ; soustraire du produit de la somme de a et de b par c le quotient de d par e. Ceci se généralise aisément à des fonctions quelconques et les programmes déterministes classiques ont alors pour modèle la classe particulière des grammaires déterministes.

Si actuellement les calculateurs obéissent essentiellement à des programmes déterministes, l'étude des problèmes conduit bien souvent à des solutions non déterministes qui seront rendues déterministes par des choix plus ou moins arbitraires afin de les rendre exploitables sur les calculateurs actuels.

Pour fixer les idées faisons l'étude non déterministe du problème du tri par partitionnement d'un ensemble d'objets E, le résultat de ce tri étant une liste de ces éléments ordonnés suivant un critère qui pourra être testé entre deux éléments à l'aide des opérateurs habituels : $<$, $\leq$, $\geq$, $>$.

On notera $\emptyset$ l'ensemble vide et Λ la liste vide et on disposera des opérateurs :

C : Concaténation de deux listes.

+ : Ajout d'un élément à un ensemble.

− : Retrait d'un élément d'un ensemble.

V : V(E) vrai ssi E est vide

U : U(E) vrai ssi E contient un seul élément.

? : Un opérateur de choix déterministe dans un ensemble, c'est à dire que ?(E) choisit toujours le même élément dans $E \neq \emptyset$.

La fonction TRI(E) devra donc donner comme résultat la liste triée composée des mêmes éléments que E. Il est clair que si E est vide le résultat doit être Λ et si E contient un seul élément la liste contiendra ce seul élément et on l'identifiera à cet élément que l'on pourra extraire par le choix déterministe ?(E). Si E contient plusieurs éléments on pourra seulement essayer de partitionner E en deux parties E_1 et E_2 telles qu'aucun élément de E_1 ne soit supérieur à un élément de E_2. On aura alors

$TRI(E) = C(TRI(E_1), TRI(E_2))$ et le tri pourra être effectif si à tout moment on peut trouver un partitionnement effectif (en deux parties non vides).

Le partitionnement consistera à retirer un à un les éléments de E en utilisant un tirage ? et à les mettre dans E_1 ou E_2 selon un critère qui pourra être la comparaison à une valeur de référence qui devra être fixe d'un bout à l'autre du partionnement. Il est bien clair que l'efficacité maximum sera atteinte si la valeur partage E en deux sous-ensembles égaux. Cette valeur dépend donc de E, on la désignera par K(E), ce pourra être une valeur centrale ou au pis aller celle d'un élément quelconque de E.

La fonction de partitionnement opérera alors sur les trois sous-ensembles E'_1, E', E'_2 dont la réunion est E et où E'_1 et E'_2 représentent E_1 et E_2 en cours de construction, le critère de comparaison apparaissant en quatrième paramètre. Le résultat de PARTITION (E'_1, E', E'_2, K(E)) où aucun élément de E'_1 n'est supérieur à un élément de E'_2 et où la réunion de E'_1, E' et E'_2 est E, doit être la liste triée TRI(E).

On peut alors écrire le programme non déterministe :

```
TRI(E) = Si V(E) alors Λ
      ou Si U(E) alors ?(E)
      ou Si non V(E) alors PARTITION (Ø, E, Ø, K(E)).
PARTITION (inf, reste, sup, critère) =
         Si V(reste) alors C(TRI(inf), TRI(sup))
      ou Si non V(reste) alors
              REDUIRE(inf, reste, sup, critère, ?(reste)).
```

REDUIRE (inf, reste, sup, critère, élém) =

Si élém ≤ critère alors

PARTITION(inf + élem, reste - élem, sup, critère)

ou Si élem ≥ critère alors

PARTITION (inf, reste - élem, sup + élem, critère).

Une expression peut être une solution associée à TRI(E) dans ce système d'équations, par exemple : e_1 = si U(E) alors ?(E)

ou encore

e_2 = Si non V(E) alors

Si non V(E) alors

Si ?(E) ≤ K(E) alors

Si V(E - ?(E)) alors

C(Si U(∅ + ?(E)) alors ?(∅ + ?(E)),

Si V(∅) alors Λ).

Une telle expression appliquée à un ensemble E ou bien donnera la liste triée ou bien échouera, étant entendu que toute expression conditionnelle "si test alors expression" échoue quand le test est faux. Avec cette interprétation les deux expressions précédentes trient des ensembles de 1 élément, pour tout ensemble E le système a des solutions qui trient E et toute solution appliquée à E qui n'échoue pas donne comme résultat la liste triée.

On pourra parvenir à une forme plus habituelle du programme si on sait le rendre déterministe. Cela pourra se faite pour TRI et PARTITION en utilisant le traditionnel opérateur si-alors-sinon :

```
TRI(E) = Si V(E) alors Λ
        Sinon si U(E) alors ?(E)
              sinon PARTITION (∅, E, ∅, K(E)).
PARTITION (inf, reste, sup, critère) =
     Si V (reste) alors C(TRI(inf), TRI(sup))
     Sinon REDUIRE (inf, reste, sup, critère, ?(reste)).
```

Par contre REDUIRE ne peut être rendu aussi facilement déterministe, ainsi si l'on écrivait

```
REDUIRE (inf, reste, sup, critère, élem) =
    Si élem ≤ critère alors
         PARTITION (inf + élem, reste - élem, sup, critère)
    Sinon PARTITION (inf, reste - élem, sup + élem, critère)
```

Le programme ainsi défini serait incapable de trier un ensemble contenant plusieurs éléments égaux du point de vue de la relation d'ordre.

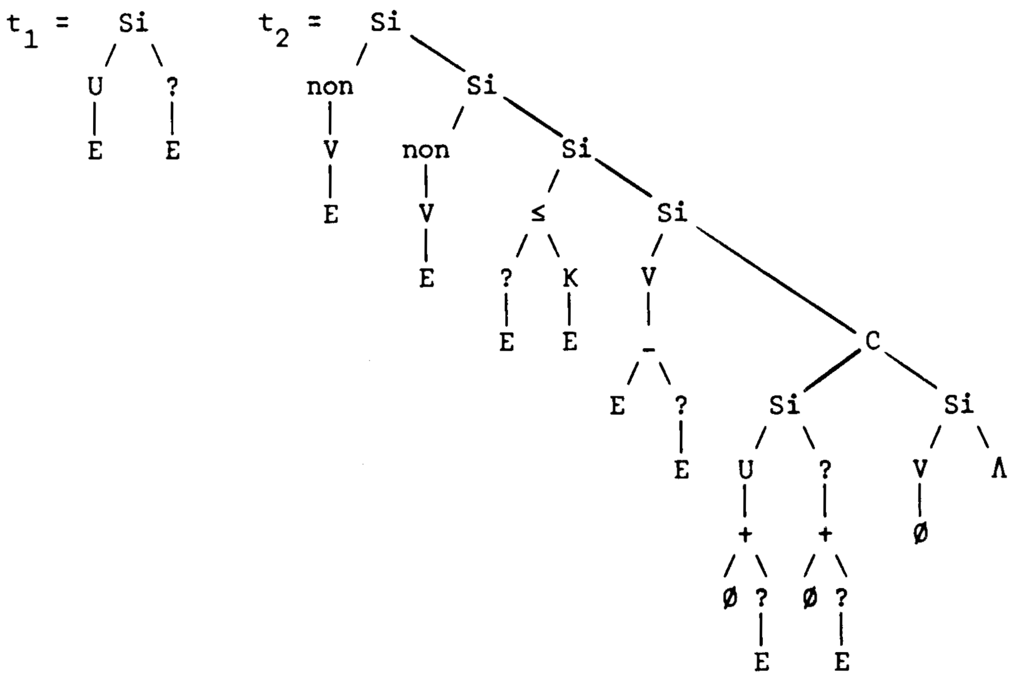
Une solution assurant l'effectivité de la partition pourrait être :

```
REDUIRE (inf, reste, sup, critère, élem) =
  Si élem < critère alors
        PARTITION (inf + élem, reste - élem, sup, critère)
  Sinon si élem > critére alors
        PARTITION (inf, reste - élem, sup + élem, critère)
        sinon si V(inf) alors
                 PARTITION (inf + élem, reste - élem, sup, critère)
            sinon PARTITION (inf, reste - élem, sup + élem, critère).
```

D'autres solutions plus élégantes et efficaces, s'appuyant sur des propriétés de la représentation de E pourront être trouvées dans [Wi, Kn]

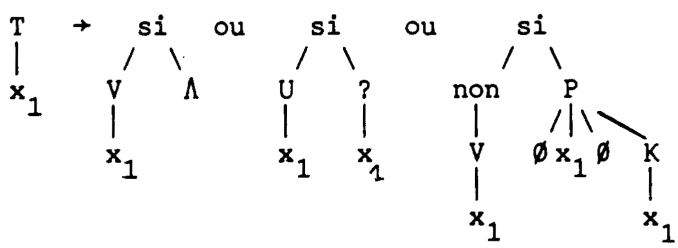
Ce genre d'analyse des problèmes est abondamment décrit dans [Do] et [Ru].

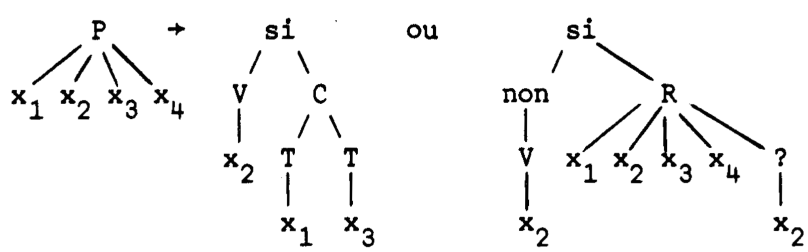
De même qu'il est commode de représenter une expression par un arbre, ainsi t_1 et t_2 représentent e_1 et e_2 :

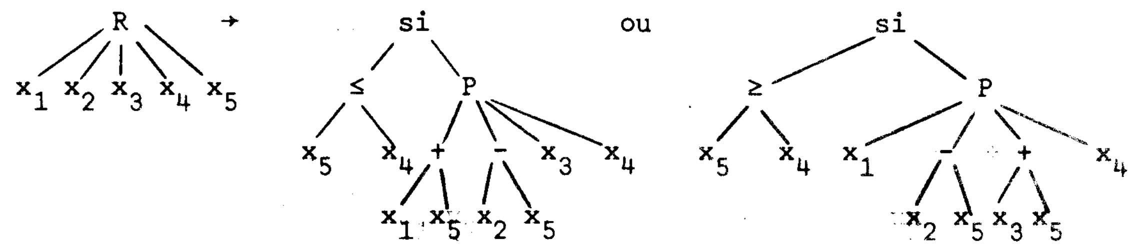


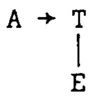
On associera au système d'équations un système de génération de tous les arbres représentant ses solutions :

en posant les règles de réécriture

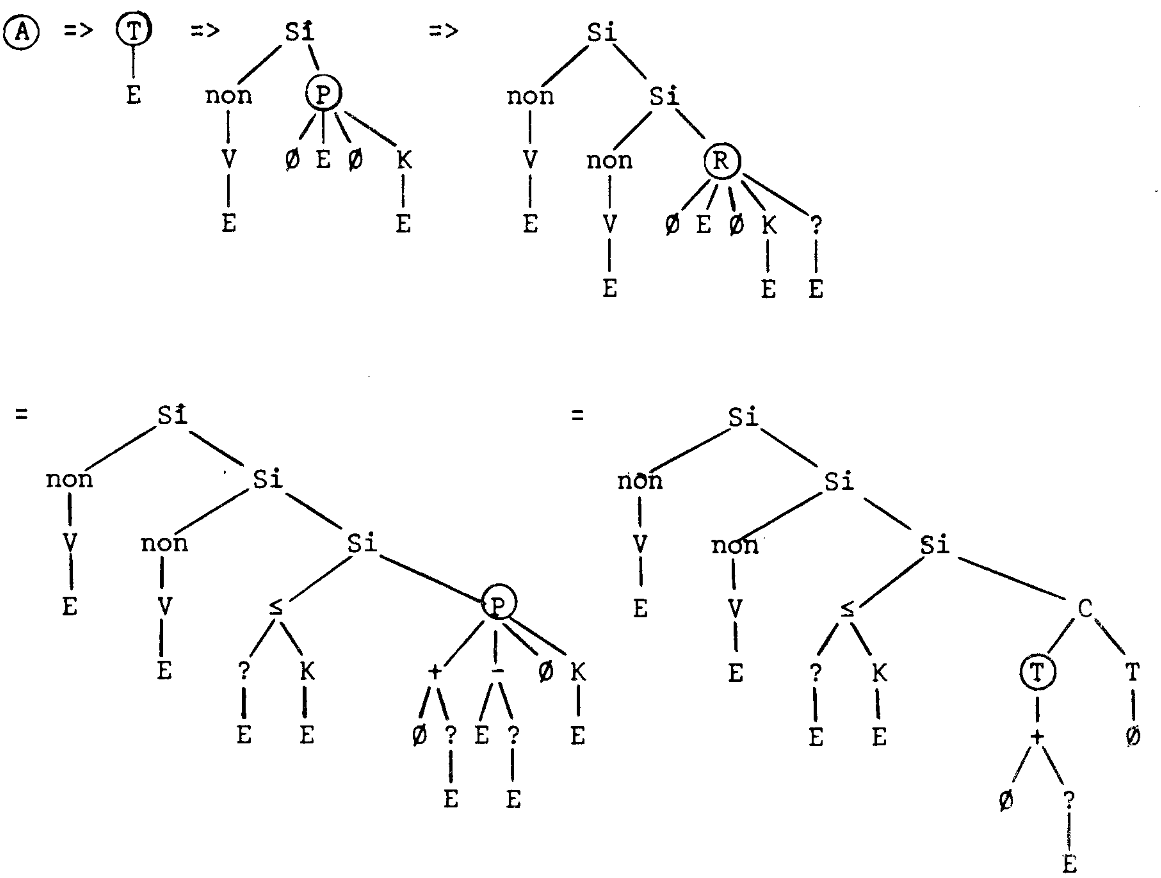




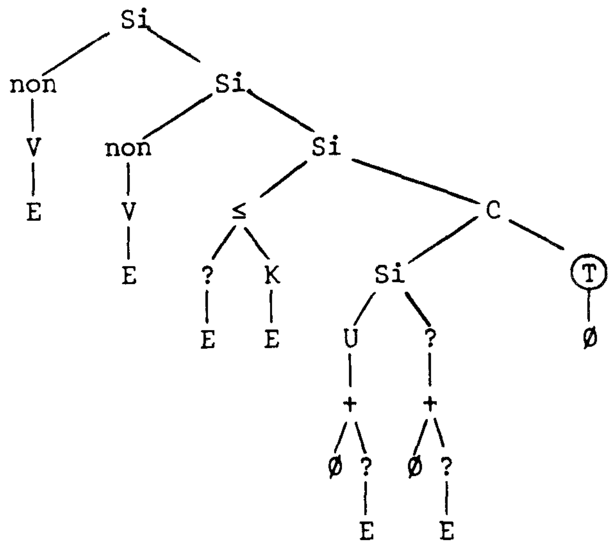




On obtiendra chaque représentation d'une solution en partant de l'axiome A et en appliquant les règles de réécriture aux non terminaux A, T, P et R comme on remplace en Algol 60 chaque appel de procédure par une copie du corps dans lequel les paramètres formels (notés ici x_1, x_2, x_3 ...) sont remplacés par les paramètres effectifs. Un exemple d'une telle suite de transformations ou dérivations est :



=>



=> t_2

Dans ces règles de réécriture, on ne trouve jamais deux symboles non terminaux situés l'un au dessous d'un autre, c'est à dire apparaissant dans un paramètre d'un autre. Cette propriété remarquable fait que l'ordre dans lequel les règles sont appliquées est sans influence sur le résultat d'une dérivation. Le remplacement d'un non terminal par un arbre terminal dérivé de lui, fixe le calcul qui sera fait et donc le résultat. En ce sens ce remplacement peut être identifié au calcul qui en résultera et de même qu'en Algol 60 la question se pose de savoir si un paramètre est évalué avant l'appel d'une fonction (appel par valeur) ou chaque fois que l'évaluation de la fonction fera usage de ce paramètre (appel par nom), de même au cours des dérivations dans un système quelconque, la question se posera de savoir si les remplacements des paramètres devront être faits avant celui de la fonction c'est à dire de bas en haut (dérivations ascendantes ou IO) ou si le remplacement de la fonction doit être fait avant l'évaluation des paramètres, c'est à dire de haut en bas (dérivations descendantes ou OI). Il est alors clair que l'étude des propriétés de ces deux types de dérivations apportera des informations sur les propriétés de l'appel par nom et de l'appel par valeur.

Examinons l'exemple simple suivant :

R = C(H)

C(x) = x * x

H = 0 ou 1 ou 2 ou 3.

On pourra interpréter H comme une fonction aléatoire choisissant un entier dans un intervalle donné et * comme la multiplication. Il est clair que si le paramètre de C est appelé par valeur, C est la fonction carrée, s'il est appelé par nom C fait seulement le produit de deux évaluations du paramètre. En mettant le système d'équations sous forme de système de réécriture :

```
R → C    C  → *         et H → 0 ou 1 ou 2 ou 3
    |    |   / \
    H    x  x   x
```

on obtient, si on commence par réécrire les symboles du bas,

```
R => C => {C ou C ou C ou C} => {  *    ou    *  ou    *   ou    *  }
     |     |    |    |    |       / \        / \      / \       / \
     H     0    1    2    3      0   0      1   1    2   2     3   3
```

c'est à dire le même résultat que par l'appel par valeur.

Si on commence par réécrire les symboles du haut

```
R => C =>   *    => {  *    ou    *    ou ... ou    *    ou    *    ou ... ou    * }
     |     / \        / \        / \               / \        / \               / \
     H    H   H      0   0      0   1             0   3      1   0             3   3
```

c'est à dire le même résultat que par l'appel par nom.

La définition de PARTITION est peu efficace en ce sens que tous ses paramètres ne sont pas toujours utilisés. Ainsi on a

```
      P         →    Si        où x4 n'est pas utilisé.
   / / \ \          /  \
 x1 x2 x3 x4       V    C
                   |   / \
                  x2  T   T
                      |   |
                     x1   x3
```

On dira que cette règle est incomplète et nous étudirons sous quelles conditions de telle règles peuvent être évitées. Il en sera de même pour les fonctions de projection du type $Z(x_1, x_2, x_3) \rightarrow x_2$, correspondant à des règles dites non strictes ou effaçantes par analogie avec la théorie des langages.

A - LE CADRE FORMEL

1) DEFINITIONS PRELIMINAIRES

On appelle **alphabet gradué** un ensemble de symboles à chacun desquels est associé un entier naturel appelé son **degré** ou son **arité**. On notera d la fonction qui associe son degré à un symbole. De cette façon, dans un arbre représentant une expression, on identifiera souvent les symboles aux opérateurs et constantes ; le degré de chaque symbole indiquant le nombre d'opérandes sur lesquels travaille l'opérateur. Un problème peut se poser, par exemple dans l'expression e = -(A+(B-C)) représentée par

```
  -
  |
  +
 / \
A   -
   / \
  B   C
```

les deux utilisations monadique et dyadique du signe moins obligeraient à lui octroyer deux degrés. On pourra toujours contourner ce problème ent utilisant deux symboles distincts ou en disant que les symboles de l'expression sont des étiquettes et que les symboles de l'alphabet gradué sont des couples <étiquette, degré>.
Ainsi à l'expression e on fera correspondre l'arbre

```
        <-, 1>
          |
        <+, 2>
       /      \
<A, 0>         <-, 2>
              /      \
        <B, 0>        <C, 0>
```

La plupart des alphabets gradués considérés sont finis. Un exemple simple d'un tel alphabet est

$\Sigma = \{\#, a, \bar{a}, b, \beta\}$ avec $d(\#) = 0$, $d(a) = d(\bar{a}) = 1$, $d(b) = 2$ et $d(\beta) = 4$.

Il sera alors commode de noter $\Sigma_i = \{X \in \Sigma \mid d(x) = i\}$ pour tout i entier.

Ceci nous permettra d'introduire tout alphabet Σ en énumérant les parties Σ_i de Σ non vides : $\Sigma_o = \{\#\}$, $\Sigma_1 = \{a, \bar{a}\}$, $\Sigma_2 = \{b\}$, $\Sigma_4 = \{\beta\}$.

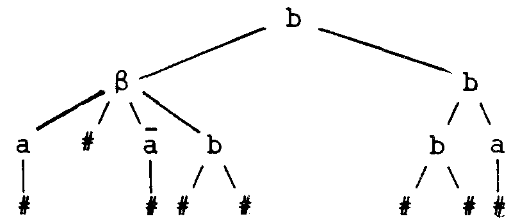
L'**ensemble des arbres** sur Σ noté T_Σ est le plus petit ensemble contenant Σ_o et défini par :

$T_\Sigma = \{\sigma(t_1, \ldots, t_n) \mid \sigma \in \Sigma_n \text{ et } t_1, \ldots, t_n \in T_\Sigma\}$.

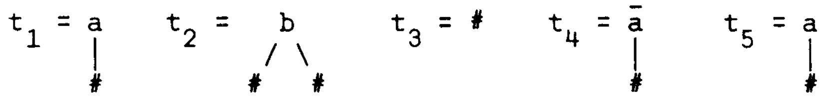
On appelle **Forêt** toute partie de T_Σ.

Ainsi étant donné l'alphabet Ω défini par $\Omega_o = \{A, B, C, D, E\}$ et $\Omega_2 = \{+, -, *, \div\}$ où A, B, C, D et E sont 5 constantes entières et $+$, $-$, $*$, $\div$ les opérateurs entiers usuels, T_Ω sera l'ensemble des expressions préfixées utilisant ces éléments. Par exemple A, +(A, -(B, C)), *(A, +(A, B) sont des arbres de T_Ω. On pourra définir une forêt de T_Ω distincte de T_Ω par une propriété vraie sur une partie de T_Ω seulement. Par exemple la forêt $F = \{t \in T_\Omega \mid \text{tout symbole de } \Omega_o \text{ est utilisé une fois et une seule dans } t\}$.
Alors $\tau = +(A, *(+(D, E), -(B, C))) \in F$. Une autre forêt F_x abondamment étudiée est définie par $F_x = \{t \in F \mid \text{le résultat de } t \text{ est } x\}$, en appelant résultat de t la valeur arithmétique calculée.
Un exemple d'arbre de T_Σ est $t = b(\beta(a(\#),\#,\bar{a}(\#),b(\#,\#)),b(b(\#,\#),a(\#)))$.

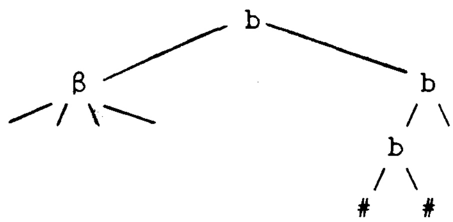
Il sera souvent commode de dessiner les arbres afin d'en rendre la structure plus visible. On procédera alors d'une manière conforme à la coutume occidentale qui est de lire et écrire de gauche à droite et de bas en haut. Ainsi tout arbre de la forme $\sigma(t_1, \ldots, t_n)$, σ étant un symbole d'arité n, sera dessiné en écrivant le symbole σ et sous ce symbole, les dessins de $t_1, \ldots, t_n$ dans l'ordre de gauche à droite, reliés à σ par des segments, étant entendu que tout arbre réduit à un symbole d'arité 0 a pour dessin ce symbole. Ainsi t se dessine :



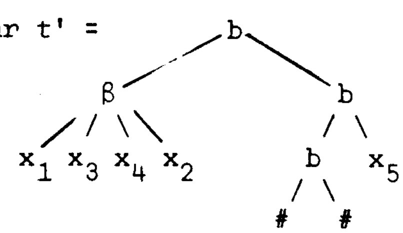
Il est clair que les arbres :



entrent dans la composition de t ainsi que l'"arbre incomplet" :



Malheureusement il existe beaucoup de possibilités pour accrocher les arbres t_1, t_2, t_3, t_4 et t_5 à ce morceau d'arbre et l'on a peu de chances de reconstituer t en procédant au hasard. Un moyen commode d'indiquer comment chaque arbre du vecteur $[t_1, t_2, t_3, t_4, t_5]$ doit être accroché est d'indicer les branches pendantes de l'arbre incomplet par des symboles spéciaux qu'on appelle **variables**, auxquelles on accorde l'arité 0 et qu'on note généralement $x_1, x_2, \ldots, x_n$. On remplacera alors avantageusement l'arbre incomplet précédent par t' =



Il est d'usage courant pour tout $n \in \mathbb{N}$ de noter $[n] = \{1, 2, \ldots, n\}$. On notera aussi $\mathcal{X} = \{x_1, x_2, \ldots, x_n, \ldots\}$ l'ensemble de toutes les variables et $\mathcal{X}_n = \{x_i \mid i \in [n]\}$. Avec ces notations $t' \in T_{\Sigma \cup \mathcal{X}_5}$.

A l'aide de tout arbre τ de $T_{\Sigma \cup \mathcal{X}_n}$ et de tout vecteur $[\tau_1, \ldots, \tau_n]$ de n arbres de T_Σ on définit un arbre de T_Σ par l'opération de **greffe** [Bo] comme suit : $\tau.[\tau_1, \ldots, \tau_n]$ est l'arbre obtenu en substituant dans τ à toute occurrence de variable x_i, $i \in [n]$, l'arbre τ_i. Ainsi $t'.[t_1, t_2, t_3, t_4, t_5] = t$.

Il est à remarquer qu'un cas particulier d'arbre incomplet est un symbole d'arité non nulle, ainsi à σ d'arité n on associera l'arbre $\sigma(x_1, x_2, \ldots, x_n)$ que l'on notera le plus souvent σ. Nous avons déjà utilisé la notion de vecteur d'arbres, ceux de l'exemple ne comportaient pas de variables, mais il est évident que si l'on veut construire un arbre complexe par greffes successives on devra considérer des vecteurs d'arbres de $T_{\Sigma \cup X}$. Par exemple en posant $t'_1 = \beta(x_1, x_3, x_4, x_2)$ et $t'_2 = b(b(\#,\#), x_5)$ et $t'' = b(x_1, x_2)$ on pourra écrire $t' = t''.[t'_1, t'_2]$ il apparait alors que cette greffe est associative et qu'on pourra écrire :

$t = t''.[t'_1, t'_2].[t_1, t_2, t_3, t_4, t_5]$

On notera qu'ici $t'_1 = \beta(x_1, x_3, x_4, x_2)$ ne s'identifie pas à β, il est le résultat de la greffe sur β du vecteur $[x_1, x_3, x_4, x_2]$ dont le seul but est de renuméroter les branches pendantes de β. En fait ce vecteur définit simplement une application d'une partie de $\mathbb{N}$ dans une partie de $\mathbb{N}$. Ces vecteurs particuliers jouent un rôle très important c'est pourquoi Arnold et Dauchet ont décidé de leur donner un nom spécial, celui de **torsion.**

On a coutume d'utiliser un vocabulaire qui s'appuie sur cette manière de dessiner les arbres. Ainsi appelle-t-on **racine** de tout arbre $\tau = \alpha(\tau_1, \ldots \tau_n)$, le symbole α situé en son **sommet** et **feuille** tout symbole d'arité 0, c'est à dire n'ayant aucun symbole au dessous de lui, dans un arbre. On appellera **feuillage** d'un arbre τ le mot formé en mettant bout à bout toutes les feuilles de τ :

- si $\tau = x_i$, feuillage (t) = x_i
- si $\tau = \#$ avec $d(\#) = 0$, feuillage (t) = #
- si $\tau = \sigma(t_1, t_2, \ldots, t_n)$ alors

  feuillage (t) = feuillage (t_1) feuillage (t_2) ... feuillage (t_n).

On a souvent à préciser l'ensemble des variables présentes dans un arbre τ et on note

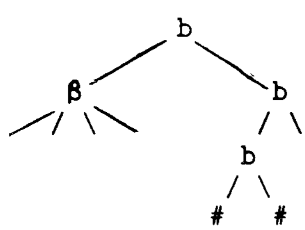
$\mathbf{var}(\tau) = \{x_i \in X \mid \exists\, u, v \in (\Sigma_o \cup X)^* \text{ tels que } u\, x_i\, v = \text{feuillage}(\tau)\}$.

<u>Exemples</u> : racine (t') = b, feuillage (t') = $x_1\ x_3\ x_4\ x_2\ \#\ \#\ x_5$ et

$\text{var}(t') = \{x_1, x_2, x_3, x_4, x_5\} = X_5$.

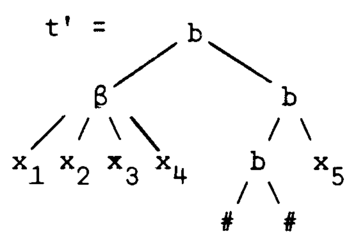
racine (t'_2) = b, feuillage (t'_2) = $\#\ \#\ x_5$ et $\text{var}(t'_2) = \{x_5\}$.

Si nous reprenons l'arbre incomplet



la numérotation des branches pendantes qui paraît la plus naturelle est certainement celle de

t' =



Un tel arbre dans lequel les variables sont numérotées à partir de un et de un en un de gauche à droite est dit **initial**. Il est clair que tout arbre peut se décomposer de façon unique en un arbre initial et une torsion [Arnold - Dauchet], ainsi $t' = \tilde{t}'.[x_1, x_3, x_4, x_2, x_5]$.

2) LES NOTATIONS DES MAGMOIDES

Les notations utilisées jusque là s'avéreraient vite insuffisantes pour indiquer avec précision des manipulations d'arbres un peu complexes, c'est pourquoi nous allons introduire les notations des magmoïdes de Arnold et Dauchet, que nous utiliserons par la suite.

Etant donnés un alphabet gradué Σ et l'ensemble X des variables, nous noterons $T(\Sigma)^p_q$ la **fibre** ou ensemble de vecteurs $\{<q ; t_1, \ldots, t_p> \mid t_1, \ldots, t_p \in T_{\Sigma \cup X_q}\}$. Toutefois quand il ne sera pas nécessaire de préciser q, le vecteur $\vec{v} = <q ; t_1, \ldots, t_p> \in T(\Sigma)^p_q$, de **degré supérieur** p et de **degré inférieur** q sera simplement noté $[t_1, \ldots, t_p]$. Le degré supérieur de $\vec{v}$ est le nombre de ses composantes et le degré inférieur indique l'ensemble des variables que l'on peut rencontrer dans chaque composante. Nous noterons $T(\Sigma)^n = \bigcup_{p \in \mathbb{N}} T(\Sigma)^n_p$, d'où $T(\Sigma) = \bigcup_{n \in \mathbb{N}} T(\Sigma)^n$ et en identifiant le vecteur $<q ; t>$ à l'arbre t de $T_{\Sigma \cup X_q}$ les notations $T(\Sigma)^1_q$ et $T(\Sigma)^1_o$ seront utilisées de préférence à $T_{\Sigma \cup X_q}$ et T_Σ, étant entendu que X_o est l'ensemble vide.

Une partie importante de $T(\Sigma)$ est l'ensemble Θ des torsions défini par ses intersections avec toute partie $T(\Sigma)^n_p$ de $T(\Sigma)$ par $\Theta^n_p = \Theta \cap T(\Sigma)^n_p = \{<p ; x_{i_1}, \ldots, x_{i_n}> \mid \forall j \in [n], i_j \in [p]\}$.
Toute torsion $\theta = <p ; x_{i_1}, \ldots, x_{i_n}>$ de Θ^n_p s'identifie à l'application θ de $[n]$ dans $[p]$ définie pour tout $k \in [n]$ par $\theta(k) = i_k$.

On définit alors la **greffe** d'un vecteur $\vec{v} = <p ; t_1, \ldots, t_n> \in T(\Sigma)^n_p$ sur un arbre T de $T(\Sigma)^1_n$, notée $t.\vec{v}$, comme l'opération définie récursivement par :

- si $t = x_i$ alors $t.\vec{v} = t_i$
- si $t = \#$ avec $\# \in \Sigma_o$ alors $t.\vec{v} = \#$
- si $t = \sigma(\tau_1, \ldots \tau_q)$ avec $\sigma \in \Sigma_q$ et $\tau_1, \ldots \tau_q \in T(\Sigma)^1_n$

alors $t.\vec{v} = \sigma(\tau_1.\vec{v}, \dots \tau_q.\vec{v})$.

Cette définition est compatible avec celle vue précédemment. On notera que $t = \#$ n'est pas incompatible avec $t \in T(\Sigma)^1_n$, $n > 0$, c'est à dire que pour que $t \in T(\Sigma)^1_n$ il n'est pas nécessaire que toutes les variables $x_1, \dots, x_n$ se trouvent dans t, par contre cela implique qu'aucune variable x_i avec $i > n$ ne se trouve dans t. On a donc pour tous $p, q, q' \in \mathbb{N}$ avec $q < q'$, $T(\Sigma)^p_q \subset T(\Sigma)^p_{q'}$, par contre tout vecteur de $T(\Sigma)^p_q$ doit avoir effectivement p composantes.

On généralise alors facilement la greffe en un produit de composition entre deux vecteurs $\vec{u} = <n\ ;\ \tau_1, \dots \tau_q> \in T(\Sigma)^q_n$ et $\vec{v} \in T(\Sigma)^n_p$, c'est à dire que le degré inférieur du premier est égal au degré supérieur du second, en écrivant :

$$\vec{u}.\vec{v} = <p\ ;\ \tau_1.\vec{v}, \dots, \tau_q.\vec{v}> \in T(\Sigma)^q_p$$

Le produit de composition est associatif ainsi que la greffe [Bo], [Th], [AD m1].

```
Exemple : u = <4 ;    b     ,      β       > ∈ T(Σ)²₄
                     / \         //\\
                   x1   x2    x1 x1 ā a
                                    | |
                                   x4 x2

v = <3 ; a , ā ,    b    , #> ∈ T(Σ)⁴₃
         |   |     / \
         x1  x2  x1   x1

w = <5 ; #, a ,    b    > ∈ T(Σ)³₅
            |     / \
            x5  x1   x3
```

```
 → →
 u.v = <3 ;   b   ,      β         > ∈ T(Σ)^2_3
             / \_      / /\_ \
            a   ā     a  a  ā  a
            |   |     |  |  |  |
            x1  x2    x1 x1 #  a
                                |
                                x2

 → →                  -
 v.w = <5 ; a , ā ,   b   , #> ∈ T(Σ)^4_5 et
            |   |    / \
            #   a   #   #
                |
                x5

  → →   →   →  → →
 (u.v).w = u.(v.w) = <5 ;  b   ,     β      > ∈ T(Σ)^2_5
                          / \_     // \\
                         a   ā    a a ā a
                         |   |    | | | |
                         #   a    # # # a
                             |          |
                             x5         a
                                        |
                                        x5
```

Il est clair que pour tout vecteur $\vec{v} \in T(\Sigma)^n_p$, la composition avec une torsion θ de $T(\Sigma)^p_q$ consiste à renuméroter les variables de $\vec{v}$, en remplaçant toute occurrence de x_i, $i \in p$ par $x_{\theta(i)}$, et à changer l'ensemble de variables sur lequel il est défini : $\vec{v}.\theta \in T(\Sigma)^n_q$. Un cas particulier de torsion est $\theta = <p \; ; x_1, \ldots x_p> \in T(\Sigma)^p_p$ pour laquelle on aura $\vec{v}.\theta = \vec{v}$, $\forall \vec{v} \in T(\Sigma)_p$. Une telle torsion est la restriction à $[p]$ de la fonction identité sur $\mathbb{N}$ et joue le rôle d'élément neutre pour la composition. Il est clair que si $\theta' \in \Theta^n_p$ et $\theta'' \in \Theta^p_q$ les deux applications θ' de $[n]$ dans $[p]$ et θ'' de $[p]$ dans $[q]$ se composent au sens des applications et qu'à la fonction $\theta'' \circ \theta'$ correspond le vecteur $\theta = \theta'.\theta''$ qui est une torsion $\theta \in \Theta^n_q$.

Exemple : $\theta' = <3 \; ; x_1, x_2, x_1, x_3> \in \Theta^4_3$

$\theta'' = <4 \; ; x_1, x_2, x_4> \in \Theta^3_4$

$\theta = \theta'.\theta'' = <3 \; ; x_1, x_2, x_1, x_4> \in \Theta^4_4$

et on a $\vec{u}.\theta' = <3\ ;\ \begin{array}{c} b \\ / \ \backslash \\ x_1 \quad x_2 \end{array}\ ,\ \begin{array}{c} \beta \\ x_1 \ \ x_2 \ \ \begin{array}{c}\bar{a} \\ | \\ x_3\end{array} \ \ \begin{array}{c}a \\ | \\ x_2\end{array} \end{array} > \ \in T(\Sigma)^2_3$

$\theta''.\vec{v} = <3\ ;\ \begin{array}{c} a \\ | \\ x_1 \end{array}\ ,\ \begin{array}{c} \bar{a} \\ | \\ x_2 \end{array}\ ,\ \# > \ \in T(\Sigma)^3_3$

et $\vec{u}.\vec{v} = \vec{u}.\theta'.\theta''.\vec{v} = \vec{u}.\theta.\vec{v}$ bien qu'aucune des torsions θ, θ', θ'' ne soient des applications identiques.

On généralise facilement la notion de feuillage à un vecteur en posant feuillage $(<p\ ;\ t_1, \ldots, t_n>)$ = feuillage (t_1) ... feuillage (t_n) et $\text{var}(<p\ ;\ t_1, \ldots, t_n>) = \bigcup_{i\in[n]} \text{var}(t_i)$. On dira alors qu'un vecteur $\vec{v}$ est initial dans $T(\Sigma)^n_p$ si les variables de son feuillage sont ordonnées et $\text{var}(v) = X_p$, c'est à dire s'il existe $w_o, w_1, \ldots, w_p \in \Sigma_o^*$ tels que feuillage $(\vec{v}) = w_o\, x_1\, w_1\, x_2\, w_2 \ldots x_p\, w_p$. On notera alors $\tilde{T}(\Sigma)^n_p$ l'ensemble des vecteurs initiaux dans $T(\Sigma)^n_p$ et $\tilde{T}(E)$ leur réunion. Dans la suite la tilde sera utilisée pour marquer les arbres ou vecteurs initiaux.

Il est clair que les seules torsions initiales sont les torsions identiques et que le produit de deux vecteurs initiaux est un vecteur initial.

Quand on greffe sur un arbre indexé d'autres arbres indexés, c'est à dire avec variables, il est nécessaire d'examiner avec soin l'origine des variables rencontrées dans chaque arbre. L'exemple suivant illustrera cette remarque. Posons

$$t = \begin{array}{c} b \\ \begin{array}{cc} \begin{array}{c} b \\ \begin{array}{cc} \begin{array}{c} a \\ | \\ \# \end{array} & \begin{array}{c} \bar{a} \\ | \\ \# \end{array} \end{array} \end{array} & \begin{array}{c} b \\ \begin{array}{cc} \# & \begin{array}{c} b \\ / \ \backslash \\ \# \quad \# \end{array} \end{array} \end{array} \end{array} \end{array} \quad , \quad t' = \begin{array}{c} b \\ \begin{array}{cc} \begin{array}{c} b \\ / \ \backslash \\ x_1 \quad x_2 \end{array} & \begin{array}{c} b \\ / \ \backslash \\ x_3 \quad x_4 \end{array} \end{array} \end{array}$$

$$t_1 = \begin{array}{c} a \\ | \\ \# \end{array}\ ,\ t_2 = \begin{array}{c} \bar{a} \\ | \\ \# \end{array}\ ,\ t_3 = \#\ ,\ t_4 = \begin{array}{c} b \\ / \ \backslash \\ \# \quad \# \end{array}$$

On peut alors écrire $t = t'.[t_1, t_2, t_3, t_4]$.

```
Ensuite, en posant t'1 =   b    , t'2 =   b   , t" =   b   ,
                          / \            / \          / \
                        x1   x2        x3   x4      x1   x2

→                            →
v1 = [a , ā], v2 = [#,   b  ] on a t"1 = t'1.v1, t"2 = t'2.v2, t = t".[t"1,t"2].
      |   |             / \
      #   #            #   #
```

Les variables dans t'_1 et t'_2 ayant une même origine (variables de t'), celà a un sens de définir le vecteur $[t'_1, t'_2]$ et pour la même raison le vecteur

```
 →   →                          →
[v1, v2] = [a , ā , #,   b  ] = v. On peut alors écrire t' = t".[t'1, t'2]
            |   |       / \
            #   #      #   #
```

et on a $t = t'.\vec{v}$.

Par contre si l'on découpe d'abord t en

```
τ =    b    , τ1 =   b  , τ2 =   b      de sorte que t = τ.[τ1, τ2]
      / \           / \_        / \
    x1   x2        a   ā       #   b
                   |   |          / \
                   #   #         #   #
```

et si l'on continue en découpant τ_1 en

```
τ'1 =    b    , τ"1 = a , τ"2 = ā  et
        / \             |       |
      x1   x2           #       #

τ2 en τ'2 =   b    , τ"3 = #, τ"4 =   b   de façon à avoir
             / \                     / \
           x1   x2                  #   #
```

$\tau_1 = \tau'_1.[\tau''_1, \tau''_2]$ et $\tau_2 = \tau'_2.[\tau''_3, \tau''_4]$, les variables dans τ'_1 et τ'_2 ont des origines différentes. Si l'on veut greffer τ'_1 et τ'_2 dans τ, il y aura des conflits entre leurs variables, ainsi

```
τ.[τ'1, τ'2].[τ"1, τ"2, τ"3, τ"4] =       b        ≠ t.
                                        /   \
                                       b     b
                                      / \_  / \_
                                     a   ā a   ā
                                     |   | |   |
                                     #   # #   #
```

Pour pouvoir juxtaposer $[\tau''_1, \tau''_2]$ et $[\tau''_3, \tau''_4]$ d'une part et τ'_1 et τ'_2 d'une part, il faut rendre distinctes leurs variables indépendantes, par exemple en translatant les indices des variables dans τ'_2 du nombre de variables sur lesquelles est défini τ'_1.

Cette opération porte, dans le cadre des magmoïdes, le nom de **produit tensoriel** et est définie comme suit :

Etant donnés deux vecteurs $\vec{u} = \langle p' \,;\, t'_1, \ldots, t'_{n'}\rangle \in T(\Sigma)^{n'}_{p'}$ et $\vec{v} = \langle p'' \,;\, t''_1, \ldots, t''_{n''}\rangle \in T(\Sigma)^{n''}_{p''}$, le produit tensoriel de $\vec{u}$ et $\vec{v}$ est
$\vec{u} \otimes \vec{v} = \langle p' + p'' \,;\, t'_1, \ldots, t'_{n'}, t''_1.\theta, \ldots, t''_{n''}.\theta\rangle$ avec
$\theta = \langle p' + p'' \,;\, x_{p'+1}, \ldots, x_{p'+p''}\rangle \in \Theta^{p''}_{p'+p''}$.
On a alors $\vec{u} \otimes \vec{v} \in T(\Sigma)^{n'+n''}_{p'+p''}$. Ce produit tensoriel est associatif et toujours défini sur $T(\Sigma)$.

En reprenant l'exemple précédent on a

$$t = \tau.[\tau'_1 \otimes \tau'_2].[\tau''_1, \tau''_2, \tau''_3, \tau''_4] \text{ avec } [\tau'_1 \otimes \tau'_2] = [\ \begin{array}{c} b \\ / \ \backslash \\ x_1 \quad x_2 \end{array}\ ,\ \begin{array}{c} b \\ / \ \backslash \\ x_3 \quad x_4 \end{array}\ ].$$

Le produit tensoriel de deux torsions est une torsion et le produit tensoriel de deux vecteurs initiaux est un vecteur initial.

L'élément neutre est le vecteur $\langle 0;\rangle \in T(\Sigma)^{o}_{o}$.

<u>Proposition 1</u> : Etant donnés les vecteurs u, u', v, v' de $T(\Sigma)$, si les produits de compositions u.u' et v.v' sont définis alors $(u \otimes v).(u' \otimes v')$ est défini et égal à $(u.u') \otimes (v.v')$. [AD m1].

Par convention de priorité on écrit $u.u' \otimes v.v'$ pour $(u.u') \otimes (v.v')$.

Pour illustrer cette proposition reprenons l'exemple précédent :

$\tau'_1.[\tau''_1.\tau''_2]$ et $\tau'_2.[\tau''_3, \tau''_4]$ sont définis et

$\tau'_1.[\tau''_1, \tau''_2] \otimes \tau'_2.[\tau''_3, \tau''_4] = \tau_1 \otimes \tau_2$. De plus $\tau_1 \otimes \tau_2 = [\tau_1, \tau_2]$ car τ_1 et τ_2 sont dépourvus de variables.

On a donc d'après la proposition 1

$\tau'_1.[\tau''_1, \tau''_2] \otimes \tau'_2.[\tau''_3, \tau''_4] = [\tau'_1 \otimes \tau'_2].([\tau''_1, \tau''_2] \otimes [\tau''_3, \tau''_4])$

et puisque τ''_1, τ''_2, τ''_3 et τ''_4 sont dépourvus de variables

$[\tau_1, \tau_2] = [\tau'_1 \otimes \tau'_2].[\tau''_1, \tau''_2, \tau''_3, \tau''_4]$. On avait défini t par $t = \tau.[\tau_1, \tau_2]$, on obtient $t = \tau.[\tau'_1 \otimes \tau'_2].[\tau''_1, \tau''_2, \tau''_3, \tau''_4]$.

L'ensemble $T(\Sigma)$ muni du produit de composition et du produit tensoriel est un **magmoïde** en ce sens qu'il en respecte les axiomes qui sont :

Un ensemble M est un magmoïde si et seulement si :

- M_1. Pour tous p et q dans $\mathbb{N}$ il existe une partie M_q^p de M appelée fibre p-q de M. M est la réunion de ses fibres qui sont disjointes deux à deux.
- M_2. Pour tous p, q, p', q' dans $\mathbb{N}$ et tous $m \in M_q^p$, $m' \in M_{q'}^{p'}$, le produit de composition m.m' est défini si et seulement si $q = p'$. On a alors $m.m' \in M_{q'}^p$. Le produit de composition est associatif.
- M_3. Pour tous p, q, p', q' dans $\mathbb{N}$ et tous $m \in M_q^p$, $m' \in M_{q'}^{p'}$ le produit tensoriel $m \otimes m'$ est défini. $m \otimes m' \in M_{q+q'}^{p+p'}$. Ce produit tensoriel est associatif.
- M_4. La proposition 1 est vraie.
- M_5. Il existe $e_1 \in M_1^1$ et $e_o \in M_o^o$ et pour tout $p \in \mathbb{N}^+$, $e_p = e_1 \otimes e_1 \otimes \ldots \otimes e_1$ (p fois), $e_p \in M_p^p$. Ces éléments sont tels que pour tout $m \in M_q^p$, $e_p.m = m$, pour tout $m' \in M_p^q$, $m'.e_p = m'$ et pour tout $m \in M_q^p$, $e_o \otimes m = m \otimes e_o = m$.

$\tilde{T}(\Sigma)$ et Θ sont eux aussi des magmoïdes.

<u>Note</u> Tout élément de $T(\Sigma)$ est obtenu à partir de Σ par les opérations de produit de composition, produit tensoriel et composition avec une torsion. La notation $<q\ ;\ t_1, \ldots, t_p>$ que nous utilisons fréquemment, apparaît alors comme une opération dérivée en ce sens que tout p-uple peut se décrire uniquement à l'aide de $\otimes$,. et de torsions.

<u>Exemple</u>

```
u = <4 ;     b      ,    b   > peut s'écrire
            / \         / \
           b   x2     x1   #
          / \
        x1   x3
```

$$((b.(b \otimes x_1)) \otimes (b.(x_1 \otimes \#))).<4\ ;\ x_1, x_3, x_2, x_1>$$

L'axiome M_1 rappelle que les vecteurs $v = <q\ ;\ t_1, \ldots, t_p> \in T(\Sigma)^p_q$ et $v' = <q'\ ;\ t_1, \ldots, t_p> \in T(\Sigma)^p_{q'}$ doivent être considérés comme distincts même si on les identifie (parfois à tort) dans la forme $[t_1, \ldots, t_p]$.

On a souvent à considérer les diverses composantes d'un vecteur $u = <p\ ;\ t_1, \ldots, t_n> \in T(\Sigma)^n_p$ isolément. Il est clair que la torsion $\theta_i = <n\ ;\ x_i> \in \Theta^1_n$ avec $i \in [n]$ permet par composition avec u d'isoler la $i^{\text{ème}}$ composante de u, ainsi $\theta_i.u = <p\ ;\ t_i>$. On associe alors à $\theta_i \in \Theta^1_n$ la fonction de **projection** π^i_n applicable à tout vecteur à n composantes. Il est clair que u est défini par ses n projections, et que tout élément de $T(\Sigma)$ est entièrement défini par ses projections. Nous dirons que $T(\Sigma)$ est **projetable**. De même Θ est projetable. Par contre $\tilde{T}(\Sigma)$ ne saurait être projetable car dans les composantes de

```
ũ = <5 ;    b   ,    b      > seule le première est un
           / \      / \
         x1   x2   a    b
                   |   / \
                  x3 x4   x5
```

arbre initial. On trouvera une définition axiomatique précise des magmoïdes projetables dans [AD m1].

Si $u = \langle p ; t_1, \dots, t_n\rangle \in T(\Sigma)^n_p$ a pour projections $\tau_1 = \langle p ; t_1\rangle$ $\tau_2 = \langle p ; t_2\rangle, \dots, \tau_n = \langle p ; t_n\rangle$, on a l'habitude de reconstituer u à partir de ses projections par l'opération de produit direct et on note $u = [\tau_1, \dots, \tau_n]$. Ainsi, si on se donne $v = \langle p ; t'_1, \dots, t'_q\rangle \in T(\Sigma)^q_p$, on écrira $[u, v] = \langle p ; t_1, \dots, t_n, t'_1, \dots, t'_q\rangle \in T(\Sigma)^{n+q}_p$.

Nous dirons qu'un arbre t' de $T(\Sigma)^1_p$ est un **sous-arbre** de $t \in T(\Sigma)^1_n$ s'il existe des entiers m, m_1, m_2, p_1, p_2 tels que $m = m_1 + m_2 + 1$, $\tilde{\tau} \in \tilde{\mathcal{T}}(\Sigma)^1_m$, $\vec{v}_1 \in T(\Sigma)^{m_1}_{p_1}$, $\vec{v}_2 \in T(\Sigma)^{m_2}_{p_2}$ et $\vec{w} \in T(\Sigma)^{p_1+p_2+p}_n$ tels que $t = \tilde{\tau}.[\vec{v}_1 \otimes t' \otimes \vec{v}_2].\vec{w}$.

Par exemple dans t =

```
          b
        /   \
      b       b
     / \     / \
    a   ā   x1  b
    |   |      / \
    x3  #     #   x2
```

de $T(\Sigma)^1_3$, l'arbre t' =

```
     b
    / \
  x1   ā
       |
       x2
```

de $T(\Sigma)^1_2$

est sous arbre de t. Il suffit de poser

$\tilde{\tau}$ =

```
     b
    / \
  x1   b
      / \
    x2   b
        / \
       #   x3
```

$\in \tilde{\mathcal{T}}(\Sigma)^1_3$, $\vec{v}_1 = \langle 0;\rangle \in T(\Sigma)^o_o$, $\vec{v}_2 = \langle 2 ; x_1, x_2\rangle \in T(\Sigma)^2_2$

et $\vec{w}$ = <3 ;

```
 a
 |
 x3
```

, #, x_1, x_2>. On a alors $\vec{v}_1 \otimes t' \otimes v_2$ = <4 ;

```
     b
    / \
  x1   ā
       |
       x2
```

, x_3, x_4>

et $[\vec{v}_1 \otimes t' \otimes \vec{v}_2].\vec{w} = <3 ;$ b $, x_1, x_2>$ d'où $\tilde{\tau}.[\vec{v}_1 \otimes t' \otimes \vec{v}_2].\vec{w} = t$

```
    b
   / \
  a   ā
  |   |
  x3  #
```

On notera que $t \in T(\Sigma)^1_n$ est bien sous arbre de lui-même ; il suffit de prendre $\tilde{\tau} = x_1$, $\vec{v}_1 = \vec{v}_2 = <0;>$ et $\vec{w} = <n ; x_1, \ldots, x_n>$.

Il est essentiel que $\tilde{\tau}$ soit initial dans $T(\Sigma)^1_m$. En effet en prenant

t = b on peut écrire $t = \tau.<3 ; a\#, \bar{a}\#, a\#>$ avec $\tau = b(x_1, x_3)$,

```
    b
   / \
  a   a
  |   |
  #   #
```

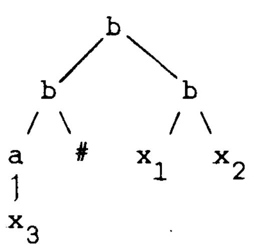
dans ce cas $\bar{a}\#$ n'est cependant pas sous arbre de t car τ n'est pas initial dans $T(\Sigma)^1_3$. On aurait aussi pu écrire que $t = \tau'.[a\#]$ avec $\tau' = b(x_1, x_2)$. Dans ce cas a# est bien un sous arbre de t en un certain sens, mais il existe deux occurrences de a# dans t. Si $t \in T(\Sigma)^1_p$, $\tilde{\tau} \in \tilde{T}(\Sigma)^1_n$, $\vec{w} \in T(\Sigma)^n_p$ et $t = \tilde{\tau}.\vec{w}$, $\tilde{\tau}$ sera dit **sous arbre initial** de t. C'est de cette décomposition qu'est issu le terme "initial". Quand dans un arbre on veut distinguer le sous arbre situé en son sommet, il est naturel de numéroter de un en un de gauche à droite les branches pendantes, c'est à dire de prendre le sous arbre initial. Il arrivera parfois de parler de **sous arbre final** à propos d'une composante de $\vec{w}$.

On dit qu'il y a une occurrence de symbole α dans un arbre t si α est un sous arbre de t.

On dit qu'une occurrence de symbole α **est au-dessus** d'une occurrence de symbole β (et donc que β est **au-dessous** de α) dans un arbre t, si α est la racine d'un sous arbre t' de t et β la racine d'un sous arbre t" de t'. De même on dit que dans t une occurrence x de variable est **sous** une occurrence de symbole α si α est la racine d'un sous arbre final t' de t et x est un sous arbre final de t'. On dit aussi que α est **immédiatement au-dessus** de β ou **père** de β ou que β est **imméditement sous** α, si α est au-dessus de β et qu'il n'existe aucune occurrence γ de symbole à la fois au-dessous de α et au dessus de β.

On dit que dans un arbre t un sous arbre t' de t est le **i^{ème} fils** d'une occurrence de symbole α de degré n dans t si l'arbre $\alpha.(\langle i-1 \;;\; x_1, \ldots, x_{i-1}\rangle \otimes t' \otimes \langle n-i \;;\; x_1, \ldots, x_{n-i}\rangle)$ est un sous-arbre de t.

Etant donnés un arbre t de racine α et une occurrence β de symbole dans t autre que la racine, α est alors au-dessus de β, et on appelle **branche jusqu'à** β dans t la liste $\gamma_1, \ldots, \gamma_p$, d'occurrences de symboles dans t avec $\gamma_p = \beta$, $\gamma_1 = \alpha$ pour tout $i < p$, γ_i est immédiatement au dessus de γ_{i+1}. On peut aussi noter les branches de façon plus précise en indiquant à chaque noeud la direction prise. Pour celà, à tout $\alpha \in \Sigma_n$, on fait correspondre les n+1 symboles $\langle\alpha, i\rangle$ avec $i \in \{0, 1, \ldots, n\}$ et toute branche $\alpha(t_1, \ldots, t_n)$ est de la forme $\langle\alpha, i\rangle\, b_i$ où b_i est une branche de t_i si $i \neq 0$ et est vide si $i = 0$. Ainsi l'arbre

t =  a pour branches $\langle b, 0\rangle$, $\langle b, 1\rangle \langle b, 0\rangle$, $\langle b, 1\rangle \langle b, 1\rangle \langle a, 0\rangle$, $\langle b, 1\rangle \langle b, 1\rangle \langle a, 1\rangle x_3$, $\langle b, 1\rangle \langle b, 2\rangle \langle \#, 0\rangle$, $\langle b, 2\rangle \langle b, 0\rangle$, $\langle b, 2\rangle \langle b, 1\rangle x_1$, $\langle b, 2\rangle \langle b, 2\rangle x_2$ et est entièrement déterminé par ses branches maximales.

En théorie des langages bon nombre de preuves se font par induction sur la longueur des mots, en théorie des arbres on peut procéder par induction sur la longueur des branches et le plus souvent sur la plus grande, c'est pourquoi on définit la **profondeur** $|t|$ d'un arbre de $T(\Sigma)_0^1$ comme la longueur de sa plus grande branche. Plus généralement on définit $|t|$, $t \in T(\Sigma)$ par :

- si $t = x_i$ alors $|t| = 0$
- si $t = \#$ avec $\# \in \Sigma_o$ alors $|t| = 1$
- si $t = \alpha(t_1, \ldots, t_n)$ avec $\alpha \in \Sigma_n$
  alors $|t| = 1 + \mathrm{Sup}\ (|t_1|, \ldots, |t_n|)$.

Nous utiliserons parfois la notation $|t|_b$ pour $t \in T(\Sigma)$ et $b \in \Sigma$ pour indiquer la profondeur de t en b qui est le plus grand nombre de symboles b rencontrés sur une même branche.

Une autre dimension importante d'un arbre est sa **largeur** qui est la longueur de son feuillage.

<u>Exemple</u> : pour l'arbre t =

```
        b
       / \
      b   b
     / \  / \
    a  #  x1  x2
    |
    x3
```

on a $|t| = 3$, $|t|_b = 2$ et largeur $(t) = 4$.

Une remarque, qui nous sera utile par la suite, est que tout arbre de profondeur bornée est de largeur bornée (mais la réciproque est fausse). En effet si $t \in T(\Sigma)$ et $|t| \leq L$, en posant $D = \sup\ (d(\alpha) \mid \alpha \in \Sigma)$, il est clair que l'arbre le plus large de profondeur L est composé exclusivement de symboles de degré D et par induction sur L on montre facilement que sa largeur est D^L d'où l'implication $|t| \leq L \Rightarrow$ largeur $(t) \leq D^L$. par contre pour que la réciproque soit vraie il faut que t contienne un nombre borné de symboles d'arité 1.

Il est facile de voir la similitude qu'il y a entre les arbres composés de symboles d'arité 1 et les mots en théorie des langages. Aussi nous utiliserons souvent les notations des mots pour de tels arbres en posant pour tout a de degré 1, $a^o = x_1$ et $a^n = a(a^{n-1})$, le produit de composition se confondant avec la concaténation.

Pour tout symbole b d'arité n nous généraliserons cette notation en écrivant b^i l'arbre balancé de profondeur i, c'est à dire dont toutes les branches maximales sont de longueur i, construit sur le symbole b. Plus précisément $b^o = x_1$ et $b^1 = b(x_1, x_1)$ et $b^i = b^1.b^{i-1}$ et pour tout $i > 0$. Alors pour tout $i \in \mathbb{N}$, $b^i \in T(\{b\})_1^1$ et est de largeur n^i. En notant $e_1^{n^i}$ l'unique élément de $\Theta_1^{n^i}$ que l'on peut écrire $e_1^{n^i} = <1 ; x_1, \ldots, x_1>$, on peut décomposer $b^i = \tilde{b}^i.e_1^{n^i}$ avec $\tilde{b}^i \in \tilde{T}(\{b\})^1_{n^i}$.

Ainsi l'arbre

```
          b
        /   \
      b       b
     / \     / \
    a   x1  a   #
    |       |
    a       a
    |       |
    a       a
    |       |
    #       x2
```

pourra être noté $\tilde{b}^2.[a^3\#, x_1, a^2\,\bar{a}\,x_2, \#]$.

L'essentiel des caractéristiques définies jusqu'alors pour les arbres s'appliquent aussi bien aux arbres indexés qu'aux autres, elles donnent peu d'indications sur les possibilités de greffe d'un arbre. Ainsi si nous posons $t = \tilde{t}.\theta$ avec $t \in T(\Sigma)$, $\tilde{t} \in \tilde{T}(\Sigma)$, $\theta \in \Theta$ on a $|t| = |\tilde{t}|$, largeur (t) = largeur $(\tilde{t})$, t et $\tilde{t}$ ont mêmes sous-arbres (sauf les sous arbres finaux) et les ensembles de symboles trouvés dans t et $\tilde{t}$ sont les mêmes. Par contre c'est θ qui contient l'essentiel des informations utiles pour greffer un vecteur sur t. Aussi donnons nous maintenant quelques propriétés importantes des torsions.

Une torsion $\theta \in \Theta_p^n$ est **linéaire** si l'application de $[n]$ dans $[p]$ qui lui est associée est injective, c'est à dire si θ s'écrit $<p ; x_{i_1}, x_{i_2}, \ldots x_{i_n}>$ et pour tous k et j dans $[n], i_k \neq i_j \iff k \neq j$. On a alors $n \geq p$.

Une torsion $\theta \in \Theta_p^n$ est **complète** si l'application de [n] dans [p] qui lui est associée est surjective, c'est à dire que θ s'écrit $<p \,;\, x_{i_1}, x_{i_2}, \ldots, x_{i_n}>$ et $\forall\, k \in [p],\ \exists\, j \in [n]$ tel que $i_j = k$. On a alors $n \leq p$.

Une torsion $\theta \in \Theta_p^n$ est **ordonnée** si l'application de [n] dans [p] qui lui est associée est non décroissante, c'est à dire que θ s'écrit $<p \,;\, x_{i_1}, x_{i_2}, \ldots, x_{i_n}>$ et $\forall\, k, j \in [n]\ k < j \Rightarrow i_k \leq i_j$. On pourra dire qu'elle est strictement ordonnée si $k < j \Rightarrow i_k < i_j$.

Une torsion à la fois linéaire complète et ordonnée est une torsion identité $<p \,;\, x_1, \ldots, x_n>$.

<u>Proposition 2</u> Pour tout $u \in T(\Sigma)_q^p$, il existe $r \in \mathbb{N}$, $\tilde{u} \in \tilde{T}(\Sigma)_r^p$, $\theta \in \Theta_q^r$ tels que $u = \tilde{u}.\theta$. Cette décomposition est unique. [AD m1].

<u>Lemme 3</u> Pour toute torsion $\theta \in \Theta_p^n$ en posant $q = \mathrm{card}(\mathrm{var}(\theta))$, on peut trouver une torsion complète $\hat{\theta} \in \Theta_q^n$ et une torsion linéaire $\bar{\theta} \in \Theta_p^q$ telles que $\theta = \hat{\theta}.\bar{\theta}$.

<u>Preuve</u>

θ est une application de [n] dans [p] et $q = \mathrm{card}(\theta([n]))$. Prenons une bijection $\bar{\theta}$ de [q] dans $\theta([n])$, $\bar{\theta}$ est une torsion linéaire de Θ_p^q. Définissons alors la surjection $\hat{\theta}$ de [p] dans [q] par $\hat{\theta} = \bar{\theta}^{-1} \circ \theta$, c'est une torsion complète de Θ_q^p. On a $\theta = \bar{\theta} \circ \hat{\theta}$ donc $\theta = \hat{\theta}.\bar{\theta}$.

□

<u>Corollaire 4</u> Pour tout $u \in T(\Sigma)_q^p$, il existe $\tilde{u} \in \tilde{T}(\Sigma)_r^p$, $\hat{\theta}$ complète dans Θ_s^r et $\bar{\theta}$ linéaire dans Θ_q^s tels que $u = \tilde{u}.\hat{\theta}.\bar{\theta}$.

<u>Remarque</u> La décomposition du lemme 3 n'est pas unique. Si l'on avait besoin de décompositions canoniques, il suffirait par exemple d'imposer à $\bar{\theta}$ d'être une fonction croissante. En l'absence d'indications contraires, les torsions surmontées du signe ^ seront complètes, celles surmontées de -, linéaires.

A première vue, l'opération de greffe pourrait sembler être une extension simple de l'opération de concaténation sur les langages. Tel est le cas si on greffe un vecteur $\vec{v} \in T(\Sigma)^p_n$ à un arbre t initial dans $T(\Sigma)^1_p$. Dans ce cas la greffe se résume en une concaténation multiple sur chacune des extrémités marquées par une variable dans t.

Si la torsion associée à t n'est pas ordonnée mais complète et linéaire, il y a peu de changements, cette torsion va seulement provoquer une greffe dans t dans un ordre différent de celui qu'ont les arbres dans $\vec{v}$.

```
exemple   t =   b       v = [a , ā]     t.v  =   b
               / \           |   |              / \
             x2   x1         #   #             ā   a
                                               |   |
                                               #   #
```

en fait on a $t = \tilde{t}.\theta$ avec $\tilde{t} = b$ et $\theta = [x_2, x_1]$

```
on en déduit par associativité t.v = t̃.θ.v = t̃.[ā , a]
                                                 |   |
                                                 #   #
```

Si la torsion associée à t n'est pas complète, cela signifie que certains arbres de $\vec{v}$ ne seront pas effectivement greffés dans t.

```
exemple  t =   b      v = [a ,   b  , ā]  t.v =   b
              / \          |    / \   |          / \
            x1   x3        #   #   #  #         a   ā
                                                |   |
                                                #   #
```

```
  b   ne participe pas effectivement à la greffe.
 / \
#   #
```

Ceci est assez gênant mais ne peut pas toujours être évité, par exemple, si t n'est pas un arbre isolé mais fait partie d'un ensemble d'arbres, F qui contient aussi l'arbre $t' = \beta(x_1, \#, x_2, x_3)$

Si alors on considère la nouvelle forêt F' obtenue en greffant sur chaque arbre de F le vecteur $\vec{v}$ alors $b(\#, \#)$ participera effectivement à certaines greffes en particulier $t'.v = \beta(a(\#), \#, b(\#, \#), \bar{a}(\#))$

La non complétude fera que même dans le cas d'arbres construits à l'aide de symboles d'arité 0 ou 1, qui ressembleront donc beaucoup à des mots, certains problèmes nouveaux se présenteront.

Il est bien connu que ce sont les torsions non linéaires qui posent le plus de problèmes délicats car elles provoquent la duplication des arbres lors des greffes :

$t = \beta(x_1, x_2, x_3, x_2)$ $\quad \vec{v} = [\#, a(\#), \bar{a}(\#)]$ $\quad t.\vec{v} = \beta(\#, a(\#), \bar{a}(\#), a(\#))$

En théorie des langages il est immédiat de définir l'opération de concaténation entre des ensembles de mots en posant $A.B = \{u.v \mid u \in A, v \in B\}$. Etant données deux parties $F_1 \subset T(\Sigma)^n_p$ et $F_2 \subset T(\Sigma)^p_q$, on peut définir le composé de F_1 et F_2 comme l'ensemble des composés d'un élément de F_1 et d'un élément de F_2 : $F_1 \circledcirc F_2 = \{u.v \mid u \in F_1, v \in F_2\}$.

A partir de cette opération de composition Arnold et Dauchet définissent le **magmoïde des parties ascendantes** de $T(\Sigma)$: $P_A(T(\Sigma))$ par $P_A(T(\Sigma))^p_q = \mathcal{P}(T(\Sigma)^p_q)$, ce qui en définit les fibres, et si $F_1 \subset T(\Sigma)^p_q$ et $F_2 \subset T(\Sigma)^{p'}_{q'}$

$F_1 \otimes F_2 = \{u \otimes v \mid u \in F_1,\ v \in F_2\}$ ce qui signifie que $F_1 \otimes F_2 \in T(\Sigma)^{p+p'}_{q+q'}$.
Les éléments neutres sont alors les ensembles $\{<i\ ;\ x_1, \ldots, x_i>\}$ et l'injection canonique qui à tout élément u de $T(\Sigma)$
fait correspondre $\{u\} \in P_A(T(\Sigma))$, fait de $T(\Sigma)$ un sous-magmoïde de $P_A(T(\Sigma))$.

Le magmoïde $P_A(T(\Sigma))$ n'est pas projetable. Il suffit pour s'en convaincre de prendre $\Sigma_o \supset \{\#, \bar{\#}\}$ $A = \{<0\ ;\ \#, \bar{\#}>, <0\ ;\ \bar{\#}, \#>\} \in P_A(T(\Sigma))^2_o$ on a
$\{<2\ ;\ x_1>\}.A = \{<0\ ;\ \#>, <0\ ;\ \bar{\#}>\}$ et $\{<2\ ;\ x_2>\}.A = \{<0\ ;\bar{\#}>, <0\ ;\#>\}$.
On remarque alors que ces projections sont aussi celles de
$B = \{<0\ ;\ \#, \#>, <0\ ;\ \bar{\#}, \bar{\#}>\}$ qui est disjoint de A.

Ceci est estrêmement génant et obligera à prendre des précautions supplémentaires chaque fois que l'opération de composition ascendante $\circledcirc$ devra être décrite au niveau des composants des vecteurs. Pour le mettre en évidence il suffit de rappeler la définition de la greffe restreinte selon Boudol [Bo], que nous noterons Ⓡ :
Si $F \in T(\Sigma)^1_n$ et pour $i \in [n]$, $F_i \in T(\Sigma)^1_p$ on pose
F Ⓡ $[F_1, \ldots, F_n] = \{t.[t_1, \ldots, t_n] \mid t \in F \text{ et } \forall\ i \in [n],\ t_i \in F_i\}$.
On a alors évidemment en posant $F' = \{[t_1, \ldots, t_n] \mid \text{pour tout } i \in [n], t_i \in F_i\}$,

$$F \text{ Ⓡ } [F_1, \ldots, F_n] = F \circledcirc F'.$$

Prenons l'exemple suivant :

$F = \left\{ \begin{array}{c} b \\ / \ \backslash \\ x_1 \quad x_2 \end{array} \right\}$, $F'_1 = \left\{\begin{array}{c} a \\ | \\ x_1 \end{array}\right\}$, $F'_2 = \left\{\begin{array}{c} \bar{a} \\ | \\ x_1 \end{array}\right\}$, $F'' = \{\#, \bar{\#}\}$ c'est à dire

que $F' = [F'_1, F'_2]$ est la partie saturée $F' = \left\{<1\ ;\ \begin{array}{c} a \\ | \\ x_1 \end{array},\ \begin{array}{c} \bar{a} \\ | \\ x_1 \end{array}>\right\}$.

Il est alors immédiat que F Ⓡ $[F'_1, F'_2] = F \circledcirc F' = \left\{ \begin{array}{c} b \\ / \ \backslash \\ a \quad \bar{a} \\ | \quad\ | \\ x_1 \quad x_1 \end{array} \right\}$

et on obtient

$(F \circledR [F'_1, F'_2]) \circledR F'' = (F \,\Theta\, F') \,\Theta\, F'' = \{$ b , b $\}$.

```
 b  ,  b  }.
/ \_  / \_
a  ā  a  ā
|  |  |  |
#  #  #̄  #̄
```

Par contre si l'on écrit $F \circledR ([F'_1, F'_2] \circledR F'')$, pour donner un sens à cette expression on devra associer à $([F'_1, F'_2] \circledR F'')$ une partie saturée G pour que $F \circledR G$ ait un sens. Il est assez naturel de définir G comme la partie saturée dont les projections sont $F'_1 \circledR F'' = \{$a , a$\}$ et

```
{a , a}
 |   |
 #   #̄
```

$F'_2 \circledR F'' = \{\bar{a} , \bar{a}\}$. On a alors $G = \{<0;a,\bar{a}>,<0;a,\bar{a}>,<0;a,\bar{a}>,<0;a,\bar{a}>\}$

```
{ā , ā}     {<0;a,ā>,<0;a,ā>,<0;a,ā>,<0;a,ā>}
 |   |          | |     | |     | |     | |
 #   #̄          # #     # #̄     #̄ #     #̄ #̄
```

et $F \circledR G = \{$ b , b , b , b $\} \neq (F \circledR [F'_1, F'_2]) \circledR F''$

```
{ b  ,  b  ,  b  ,  b  }
 / \_  / \_  / \_  / \_
 a  ā  a  ā  a  ā  a  ā
 |  |  |  |  |  |  |  |
 #  #  #  #̄  #̄  #  #̄  #̄
```

ce qui fait dire que la greffe restreinte n'est pas associative.

Cependant $F' \,\Theta\, F'' = \{<1 ; a , \bar{a}>\} \,\Theta\, \{ \#, \bar{\#}\} = \{<0 ; a, \bar{a}>, <0 ; a, a>\}$

```
{<1 ; a , ā>}              {<0 ; a, ā>, <0 ; a, a>}
      |   |                      |  |         |  |
      x1  x1                     #  #         #̄  #̄
```

donc $F \,\Theta\, (F' \,\Theta\, F'') = \{$ b , b $\} = (F \,\Theta\, F') \,\Theta\, F''$.

```
{ b  ,  b  }
 / \_  / \_
 a  ā  a  ā
 |  |  |  |
 #  #  #̄  #̄
```

Arnold et Dauchet appellent **magmoïde des parties descendantes** de $T(\Sigma)$ le magmoïde construit sur les parties saturées de $T(\Sigma)$, $P_D(T(\Sigma))$ dont les fibres sont $P_D(T(\Sigma))^p_q = [P(T(\Sigma)^1_q)]^p$ c'est à dire que tout élément $U \in P_D(T(\Sigma))^p_q$ peut s'écrire $<q ; U_1, \ldots, U_p>$ avec pour tout $i \in [p]$, $U_i \subset T(\Sigma)^1_q$. On a alors $U = \{<q ; t_1, \ldots, t_p> \mid \forall\, i \in [p]\ \ t_i \in U_i\}$. On définit alors un produit de composition descendant noté Θ identique à la greffe complète de Boudol et aux OI-substitutions d'Engelfriet :

Pour tous $U = <q\ ;\ U_1, \ldots, U_p> \in P_D(T(\Sigma))^p_q$ et $V = <r\ ;\ V_1, \ldots, V_q> \in P_D(T(\Sigma))^q_r$ on pose

$U \circledcirc V = \{\tilde{u}.[v_1, \ldots v_s] \mid \exists\ s \in \mathbb{N},\ \tilde{u} \in \tilde{T}(\Sigma)^p_s,\ \theta \in \Theta^s_q,\ \tilde{u}.\theta \in U$ et $\forall i \in [s],\ v_i \in V_{\theta(i)}\}$. Evidemment $U \circledcirc V = <r\ ;\ U_1 \circledcirc V, \ldots U_p \circledcirc V> \in P_D(T(\Sigma))^p_r$.

Pour tous $U = <q\ ;\ U_1, \ldots, U_p> \in P_D(T(\Sigma))^p_q$ et $V = <q'\ ;\ V_1, \ldots V_{p'}> \in P_D(T(\Sigma))^{p'}_{q'}$ on pose $U \otimes V = \{u \otimes v \mid u \in U$ et $v \in V\}$. En posant $\theta = <q + q'\ ;\ x_{q+1}, \ldots, x_{q+q'}>$ et en écrivant $u = <q\ ;\ u_1, \ldots u_p>$ et $v = <q'\ ;\ v_1, \ldots, v_{p'}>$ on a $u \otimes v = [u, v.\theta] = <q+q'\ ;\ u_1, \ldots, u_p, v_1.\theta, \ldots v_{p'}.\theta>$.

Toute partie réduite à un élément $\{<q\ ;\ u_1, \ldots u_p>\}$ est la partie saturée correspondant au vecteur $<q\ ;\ \{<q\ ;\ u_1>\}, \ldots \{<q\ ;\ u_p>\}>$, on identifiera alors facilement θ, $\{\theta\}$ et $<q+q'\ ;\ \{<q+q'\ ;\ x_{q+1}>\}, \ldots \{<q+q'\ ;\ x_{q+q'}>\}>$ d'où $U \otimes V = [U, V.\theta] = <q+q'\ ;\ U_1, \ldots U_p, V_1.\theta, \ldots V_{p'}.\theta> \in P_D(T(\Sigma))^{p+p'}_{q+q'}$.

Les éléments neutres de $T(\Sigma)$ étant des torsions on obtient par identification des torsions de $T(\Sigma)$ et de $P_D(T(\Sigma))$, les éléments neutres de $P_D(T(\Sigma))$.

Les éléments de $P_D(T(\Sigma))$ étant des parties saturées de $T(\Sigma)$ il est clair que $P_D(T(\Sigma))$ est projetable.

En identifiant $<q\ ;\ u_1, \ldots u_p> \in T(\Sigma)^p_q$ à $<q\ ;\ \{<q\ ;\ u_1>\}, \ldots \{<q\ ;\ u_p>\}> \in P_D(T(\Sigma))^p_q$ on fait de $T(\Sigma)$ un sous magmoïde de $P_D(T(\Sigma))$.

Etant données n forêts $F_1, \ldots, F_n$ de $T(\Sigma)^1_p$, on identifie la partie saturée $\{<p\ ;\ t_1, \ldots t_n> \mid \forall\ i \in [n],\ t_i \in F_i\} \subset T(\Sigma)^n_p$ à un élément de $P_A(T(\Sigma))^n_p$ et à l'élément $<p\ ;\ F_1, \ldots F_n>$ de $P_D(T(\Sigma))^n_p$ et on la note souvent $[F_1, \ldots F_n]$.

D'après les définitions, pour tout symbole # d'arité 0, $\{<0\ ;\ \#>\} \circledcirc \emptyset = \emptyset$ alors que $<0\ ;\ \{\#\}> \circledcirc \emptyset = <0\ ;\ \{\#\}>$.

<u>**Lemme 5**</u> Etant donnés un arbre $\tilde{t}.\theta$ avec $\tilde{t} \in \tilde{T}(\Sigma)^1_p$, $\theta \in \Theta^p_n$ et n parties $F_1, \ldots, F_n$ de $T(\Sigma)^1_q$ on a $\tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n] \subseteq \tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n]$. De plus si pour tout x_i ayant plusieurs occurrences dans θ, $\mathrm{Card}(F_i) = 1$,(ce qui est le cas si θ est linéaire) et si $\forall\, i \in [n]$, $F_i \neq \emptyset$ alors on a l'égalité.

<u>Preuve</u>

$\tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots, F_n] = \{\tilde{t}.[t_1, \ldots, t_p] \mid \forall\, i \in [p],\ t_i \in F_{\theta(i)}\}$.

Si $\tau \in \tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n]$ on a $\tau = \tilde{t}.\theta.[\tau_1, \ldots \tau_n]$ et $\forall\, i \in [n]$, $\tau_i \in F_i$ d'où en posant $\tau = \tilde{t}.[\tau'_1, \ldots, \tau'_p]$ avec $\tau'_i = \tau_{\theta(i)}$ on a $\tau'_i \in F_{\theta(i)}$ et $\tau \in \tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n]$. D'où $\tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n] \subseteq \tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n]$.

Prenons $t' \in \tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n]$ alors $t' = \tilde{t}.[t_1, \ldots t_p]$ avec pour tout $i \in [p]$, $t_i \in F_{\theta(i)}$. Si pour tous j, k tels que $\theta(j) = \theta(k) = r$, F_r contient un seul élément, on a évidemment pour tous j et k de [p] tels que $\theta(j) = \theta(k)$, $t_j = t_k$ et $[t_1, \ldots t_p] = \theta.[t'_1, \ldots t'_n]$ avec pour tout x_i apparaissant dans θ, $t'_i \in F_i$ et pour tout x_j n'apparaissant pas dans θ t'_j choisi de façon quelconque, éventuellement dans F_j si $F_j \neq \emptyset$. On a alors $t' = \tilde{t}.\theta.[t'_1, .. t'_n]$.
Si pour tout $i \in [n]$, $F_i \neq \emptyset$, on peut imposer $t'_i \in F_i$, on a alors $t' \in \tilde{t}.\theta \boldsymbol{\Theta} [F_1, \ldots F_n]$.

$\square$

3) HOMOMORPHISMES

Les homomorphismes de magmoïdes sont des applications d'un magmoïde dans un autre qui sont compatibles avec les opérations de produit de composition et de produit tensoriel, et qui préservent les degrés. Un homomorphisme de magmoïdes projetables préservera aussi les torsions. Formellement on aura les définitions suivantes :

ϕ est un homomorphisme du magmoïde M dans le magmoïde N ssi

$\forall$ p, q entiers, on a $\phi(M_q^p) \subset N_q^p$,

$\forall$ u, v $\in$ M tels que u.v soit défini, on a $\phi(u.v) = \phi(u).\phi(v)$,

$\forall$ u, v $\in$ M, on a $\phi(u \otimes v) = \phi(u) \otimes \phi(v)$.

Si M et N sont projetables, identifions leurs torsions à Θ. L'homomorphisme ϕ de M dans N est un homomorphisme de magmoïdes projetables si de plus ϕ est l'identité sur Θ.

Nous allons maintenant préciser comment toute application, respectant les degrés, d'un alphabet gradué Σ dans un magmoïde M, que l'on prendra ici de la forme $T(\Delta)$, s'étend de façon unique en un homomorphisme de $T(\Sigma)$ dans M qui soit l'identité sur les torsions. D'un point de vue algébrique, ceci revient à établir que $T(\Sigma)$ est le magmoïde projetable libre engendré par Σ.

Etant donnés deux alphabets gradués Σ et Δ et une application ϕ de Σ dans $T(\Delta)$ qui à tout symbole σ de Σ_n fait correspondre l'arbre $\phi(\sigma)$ de $T(\Delta)_n^1$, on étend cette application à $T(\Sigma)$ en posant :

$\forall \theta \in \Theta$, $\phi(\theta) = \theta$,

$\forall n \in \mathbb{N}$, $\forall \sigma \in \Sigma_n$, $\forall t_1, \ldots, t_n \in T(\Sigma)^1$,

$\phi(\sigma(t_1 \otimes t_2 \otimes \ldots \otimes t_n)) = \phi(\sigma).[\phi(t_1) \otimes \phi(t_2) \otimes \ldots \otimes \phi(t_n)]$.

<u>**Lemme 6**</u> $\forall\, u = \langle q\ ;\ t_1, \ldots, t_p\rangle \in T(\Sigma)^p_q$ on a $\phi(u) = \langle q\ ;\ \phi(t_1), \ldots \phi(t_p)\rangle$.

<u>Preuve</u>

Il suffit d'écrire $u = (\tilde{t}_1 \otimes \tilde{t}_2 \otimes \ldots \otimes \tilde{t}_p).\theta$ avec pour tout $i \in [p]$, $\pi^i_p(u) = t_i = \tilde{t}_i.\theta_i$. On a alors $\phi(u) = (\phi(\tilde{t}_1) \otimes \phi(t_2) \otimes \ldots \otimes \phi(\tilde{t}_p)).\theta$ et pour tout $i \in [p]$, $\pi^i_p(\phi(u)) = \phi(\tilde{t}_i).\theta_i = \phi(t_i)$.

□

A tout symbole σ de Σ_n, l'homomorphisme ϕ fait correspondre un arbre t_σ de $T(\Delta)^1_n$ que l'on peut décomposer en $t_\sigma = \tilde{t}_\sigma.\theta_\sigma$ avec $\tilde{t}_\sigma \in \tilde{T}(\Sigma)^1_p$ et $\theta_\sigma \in \Theta^p_n$ selon la proposition 2. Il sera alors commode de dire que ϕ possède la propriété P si cette propriété est vraie soit pour tous les arbres initiaux $\tilde{t}_\sigma$ soit pour toutes les torsions θ_σ tels que $\sigma \in \Sigma$.

Si $\forall\, \sigma \in \Sigma$, θ_σ est linéaire (resp. complète, ordonnée) ϕ sera dit **linéaire** (resp. **complet, ordonné**).

Si $\forall\, \sigma \in \Sigma$, $|\tilde{t}_\sigma| \leq 1$, c'est à dire que soit $\tilde{t}_\sigma = \delta$ avec $\delta \in \Delta$ soit $\tilde{t}_\sigma = x_1$ et ϕ fait correspondre à tout symbole de Σ un symbole de Δ ou l'efface, ϕ sera dit **alphabétique,** on dira aussi que c'est un **démarquage** [D.t3].

Si $\forall\, \sigma \in \Sigma$, $|\tilde{t}_\sigma| > 0$, c'est à dire que $\phi(\sigma)$ n'est pas résuit à une variable, ϕ sera dit **strict** ou **non effaçant.**

On appelle **démarquage propre** tout homomorphisme alphabétique strict linéaire complet ordonné. Si ϕ est un démarquage propre, pour tout $\sigma \in \Sigma_n$ on a $\phi(\sigma(x_1, \ldots x_n)) = \delta(x_1, \ldots x_n)$ avec $\delta \in \Delta_n$. On peut alors dire que ϕ sert seulement à changer les noms des symboles dans un arbre sans en changer la structure.

<u>Exemple</u> : $\Sigma_o = \Delta_o = \{\#\}$, $\Sigma_2 = \{\gamma\}$, $\Sigma_3 = \{\alpha, \beta\}$, $\Delta_2 = \{\gamma, \delta\}$ et ϕ l'homomorphisme de $T(\Sigma)$ dans $T(\Delta)$ défini par

$\phi(\#) = \#$

$\phi(\gamma) = x_1$ non complet et non strict,

$\phi(\alpha) = \gamma(\delta(x_1, x_3), \delta(x_2, x_2))$ non alphabétique et non linéaire,

$\phi(\beta) = \gamma(x_1, \delta(\#, \#))$ non alphabétique et non complet.

En posant $t = \begin{array}{c}\gamma\\ / \quad \backslash\\ t_1 \qquad x_3\end{array}$ avec $t_1 = \begin{array}{c}\alpha\\ / \ | \ \backslash\\ x_2 \ \ t_2 \ \ x_1\end{array}$ et $t_2 = \begin{array}{c}\beta\\ / \ | \ \backslash\\ x_4 \ \ x_5 \ \ x_6\end{array}$

On obtient $\phi(t_2) = \begin{array}{c}\gamma\\ / \quad \backslash\\ x_4 \qquad \begin{array}{c}\delta\\ / \ \backslash\\ \# \quad \#\end{array}\end{array}$, $\phi(t_1) = \begin{array}{c}\gamma\\ \swarrow \qquad \searrow\\ \begin{array}{c}\delta\\ / \ \backslash\\ x_2 \quad x_1\end{array} \qquad \begin{array}{c}\delta\\ / \ \backslash\\ \phi(t_2) \ \ \phi(t_2)\end{array}\end{array}$, $\phi(t) = \phi(t_1)$

d'où $\phi\left(\begin{array}{c}\gamma\\ / \quad \backslash\\ \begin{array}{c}\alpha\\ / \ | \ \backslash\\ x_2 \ \ \begin{array}{c}\beta\\ / | \backslash\\ x_4 \, x_5 \, x_6\end{array} \ \ x_1\end{array} \quad x_3\end{array} \right) = \begin{array}{c}\gamma\\ \swarrow \qquad \searrow\\ \begin{array}{c}\delta\\ / \ \backslash\\ x_2 \quad x_1\end{array} \qquad \begin{array}{c}\delta\\ / \quad \backslash\\ \begin{array}{c}\gamma\\ / \ \backslash\\ x_4 \ \begin{array}{c}\delta\\ / \ \backslash\\ \# \ \#\end{array}\end{array} \quad \begin{array}{c}\gamma\\ / \ \backslash\\ x_4 \ \begin{array}{c}\delta\\ / \ \backslash\\ \# \ \#\end{array}\end{array}\end{array}\end{array}$

On conviendra volontiers qu'un démarquage propre est un homomorphisme très simple, par contre celui de l'exemple précédent est assez complexe en ce sens qu'il transforme profondément la structure des arbres auxquels on l'applique en augmentant ou en diminuant leur profondeur ou leur largeur. Des homomorphismes qui n'opéreraient une transformation importante que sur un seul symbole pourraient être considérés comme les plus simples après les démarquages. L'étude de tels homomorphismes et de leurs effets ne serait pas trop compliquée ce qui justifie l'étude suivante qui montrera que tout homomorphisme peut être considéré comme le composé de tels homomorphismes "assez simples".

Afin de nous affranchir de l'imprécision dell'expression "assez simple", nous allons nous donner une mesure de la complexité d'un homomorphisme. Pour celà notons pour tout arbre t de $T(\Delta)$, $\nu(t)$ le nombre de symboles de Δ qu'il contient. C'est à dire que $\forall\, x_i \in X$, $\nu(x_i) = 0$, pour tout $\# \in \Delta_o$, $\nu(\#) = 1$ et pour tous $\alpha \in \Delta_n$ et $t_1, \dots t_n \in T(\Delta)$, $\nu(\alpha(t_1, \dots t_n)) = 1 + \sum_{i=1}^{n} \nu(t_i)$.

Pour tout homomorphisme ϕ de $T(\Sigma)$ dans $T(\Delta)$ nous noterons $C(\phi)$ la **complexité** de ϕ définie comme le nombre $C(\phi) = \sum_{\sigma\in\Sigma} |\nu(\phi(\sigma)) - 1|$.

Avec ces définitions tout démarquage a une complexité égale au nombre de symboles qu'il efface.

On appelle homomorphisme **élémentaire ordonné** [AD] un homomorphisme linéaire complet ordonné strict ϕ défini sur un alphabet gradué $\Sigma \cup \{\sigma\}$ avec $\sigma \notin \Sigma$ et à images dans $T(\Sigma \cup \{\delta_1, \delta_2\})$ avec $\{\delta_1, \delta_2\} \cap \Sigma = \emptyset$ et $d(\sigma) = d(\delta_1) + d(\delta_2) - 1$, tel que $\forall\, \gamma \in \Sigma$, $\phi(\gamma) = \gamma$ et en posant $n = d(\sigma)$, et $p = d(\delta_2)$ on ait $\phi(\sigma(x_1, \dots x_n)) = \delta_1(x_1, \dots x_{k-1}, \delta(x_k, \dots x_{k+p-1}), x_{k+p}, \dots x_n)$. Un tel homomorphisme est de complexité 1.

<u>**Lemme 7**</u> Tout homomorphisme ϕ (éventuellement strict, linéaire, complet ou ordonné) est le composé d'un homomorphisme strict, linéaire, complet, ordonné ϕ'' et d'un démarquage ϕ' (strict, linéaire, complet ou ordonné comme ϕ). [D-t3].

<u>Preuve</u>

Soit ϕ un homomorphisme de $T(\Sigma)$ dans $T(\Delta)$ défini par :

$\forall\, \alpha \in \Sigma$, $\phi(\alpha) = \tilde{t}_\alpha . \theta_\alpha$ avec $\tilde{t}_\alpha \in \tilde{T}(\Delta)^1_{p_\alpha}$ et $\theta_\alpha \in \theta^{p_\alpha}_{d(\alpha)}$.

Posons $\bar{\Sigma} = \{\bar{\sigma} \mid \sigma \in \Sigma, |\phi(\sigma)| > 0, \mathrm{var}(\bar{\sigma}) = \mathrm{var}(\tilde{t}_\sigma)\}$

ϕ' est le démarquage de $T(\Sigma)$ dans $T(\bar{\Sigma})$ défini par

si $\phi(\sigma) = x_i$ $\quad \phi'(\sigma) = x_i$

sinon $\phi'(\sigma(x_1, \dots x_n)) = \bar{\sigma}(x_1, \dots x_{p_\sigma}).\theta_\sigma$

ϕ'' est l'homomorphisme complet strict ordonné et linéaire de $T(\bar{\Sigma})$ dans $T(\Delta)$ défini par

$\forall\, \bar{\sigma} \in \bar{\Sigma} \quad \phi''(\bar{\sigma}) = \tilde{t}_\sigma$.

On a alors $\forall\, \sigma \in \Sigma$

ou bien $\phi'(\sigma) = x_i$ alors $\phi''(\phi'(\sigma)) = x_i$

et par définition de ϕ', $\phi(\sigma) = x_i$

ou bien $\phi'(\sigma) = \bar{\sigma}.\theta_\sigma$ et $\phi''(\bar{\sigma}) = \tilde{t}_\sigma$

alors $\phi''(\phi'(\sigma)) = \phi''(\bar{\sigma}.\theta_\sigma) = \phi''(\bar{\sigma}).\theta_\sigma = \tilde{t}_\sigma.\theta_\sigma = \phi(\sigma)$.

Il est alors clair que $\phi = \phi'' \circ \phi'$.

De plus $|\phi(\sigma)| = 0 \Leftrightarrow |\phi'(\sigma)| = 0$

et la torsion associée à $\phi'(\sigma)$ est la même que celle associée à $\phi(\sigma)$ donc si ϕ est strict, linéaire, complet ou ordonné, ϕ' l'est aussi.

$\square$

<u>Lemme 8</u> Tout homomorphisme strict (éventuellement linéaire, complet ou ordonné) de complexité supérieure à 1 est lecomposé d'homomorphismes stricts (également linéaires, complets ou ordonnés) de complexité 1.

<u>Preuve</u>

Soit ϕ un homomorphisme strict de $T(\Sigma)$ dans $T(\Delta)$ de complexité $C(\phi) > 1$. La preuve se fait par récurrence sur $C(\phi)$ en montrant qu'il existe ϕ'' et ϕ' stricts avec $C(\phi'') = 1$, $C(\phi') < C(\phi)$ et $\phi = \phi'' \circ \phi'$.

Deux cas sont à considérer selon qu'il existe ou non un symbole σ de Σ avec $\nu(\phi(\sigma)) > 2$.

a) $\exists\ \sigma \in \Sigma$ avec $\nu(\phi(\sigma)) > 2$

alors $\exists\ \delta_1, \delta_2 \in \Delta$ deux occurrences de symboles dans $\phi(\sigma)$ avec δ_1 = racine de $(\phi(\sigma))$ et δ_2 immédiatement sous δ_1 dans $\phi(\sigma)$.

$\phi(\sigma)=\tilde{t}_\sigma.\theta_\sigma$ avec $\tilde{t}_\sigma \in \tilde{T}(\Delta)^1_{p_\alpha}$ et $\theta_\sigma \in \Theta^{p_\sigma}_{d(\sigma)}$ on a donc $\tilde{t}=\delta_1(t_1,..t_{k-1},\delta_2.\vec{u},t_{k+1},..t_n)$.

Introduisons $\bar{\delta}$ un nouveau symbole d'arité $d(\delta_1) + d(\delta_2) - 1$ et définissons l'homomorphisme ϕ' de $T(\Sigma)$ dans $T(\Delta \cup \{\bar{\delta}\})$ par $\forall\ \gamma \in \Sigma,\ \gamma \neq \sigma, \phi'(\gamma) = \phi(\gamma)$ et $\phi'(\sigma) = \tilde{t}'.\theta_\sigma$ avec $\tilde{t}' = \bar{\delta}(t_1, \ldots t_{k-1}, \vec{u}, t_{k+1}, \ldots t_n) \in \tilde{T}(\Delta \cup \{\bar{\delta}\})$.

Il est clair que ϕ' est strict, $1 \leq C(\phi') < C(\phi)$ et les torsions de ϕ' sont les mêmes que celles de ϕ.

Posons ϕ'' un homomorphisme de $T(\Delta \cup \{\bar{\delta}\})$ dans $T(\Delta)$ défini par :

$\forall\ \gamma \in \Delta \quad \phi''(\gamma) = \gamma$ et $\phi''(\bar{\delta}(x_1,\ldots x_m)) = \delta_1(x_1,\ldots x_{k-1}, \delta_2(x_k,\ldots x_{k+p-1}), x_{k+p},\ldots x_m)$ avec $m = d(\delta_1) + d(\delta_2)-1 = d(\bar{\delta})$, $p = d(\delta_2)$ et k défini comme dans $\tilde{t}_\sigma$.

Alors ϕ'' est strict linéaire complet ordonné, mais peut-être pas élémentaire car δ_1 et δ_2 peuvent être deux occurrences d'un même symbole. ϕ'' est de complexité 1.

Par ailleurs il est facile de voir que $\phi = \phi'' \circ \phi'$.

b) S'il n'existe pas σ tel que $\nu(\phi(\sigma)) > 2$ alors puisque $C(\phi) > 1$ il existe σ, σ' dans Σ tels que $\nu(\phi(\sigma)) = \nu(\phi(\sigma')) = 2$.

Introduisons le nouveau symbole $\bar{\sigma}$ avec $d(\bar{\sigma}) = d(\sigma)$.

Posons ϕ' un homomorphisme de $T(\Sigma)$ dans $T(\Delta \cup \{\bar{\sigma}\})$ défini par :

$\forall\ \gamma \in \Sigma, \gamma \neq \sigma \quad \phi'(\gamma) = \phi(\gamma)$

et $\phi'(\sigma) = \bar{\sigma}$.

Il est clair que $1 \leq C(\phi') < C(\phi)$ et ϕ' est strict. D'autre part ϕ' hérite des torsions de ϕ sauf θ_σ qui est remplacée par une torsion identité.

Posons ϕ'' un homomorphisme de $T(\Delta \cup \{\bar{\sigma}\})$ dans $T(\Delta)$ défini par :

$\forall\ \gamma \in \Delta \quad \phi''(\gamma) = \gamma$

et $\phi''(\bar{\sigma}) = \phi(\sigma)$.

Alors $C(\phi'') = 1$ et ϕ'' est strict, ses torsions sont des torsions identité ou θ_σ qui lui vient de ϕ.

De plus il est facile de montrer que $\phi = \phi'' \circ \phi'$.

□

<u>Lemme 9</u> Tout homomorphisme linéaire, complet, ordonné et strict de complexité 1 est le composé d'un démarquage propre et d'un homomorphisme élémentaire ordonné.

<u>Preuve</u>

Soit ϕ linéaire complet strict ordonné de complexité 1 de $T(\Sigma)$ dans $T(\Delta)$. S'il n'est pas élémentaire ce ne peut être que pour des raisons de noms de symboles, un démarquage suffira donc à changer ces noms.

$C(\phi) = 1$ et ϕ est strict, donc $\exists\ \sigma \in \Sigma$ tel que $\nu(\phi(\sigma)) = 2$, il existe donc deux occurrences ϕ_1 et ϕ_2 de symboles de Δ dans $\phi(\sigma)$. On pose $\phi(\sigma(x_1, \dots x_n)) = \delta_1(x_1, \dots x_k, \delta_2(x_{k+1}, \dots x_{k+p}), x_{k+p+1}, \dots x_n)$. De plus $\forall\ \gamma \in \Sigma$, $\gamma \neq \sigma$ $\phi(\delta) = \gamma'$ avec $\gamma' \in \Delta$ et $d(\gamma) = d(\gamma')$. Posons $\hat{\Sigma} = (\Sigma\ /\ \{\sigma\}) \cup \{\sigma_1, \sigma_2\}$ où σ_1 et σ_2 sont deux nouveaux symboles avec $d(\sigma_1) = d(\delta_1)$ et $d(\sigma_2) = d(\delta_2)$. Ces deux symboles sont introduits car si δ_1 et δ_2 dont deux occurrences d'un même symbole, celà interdit que ϕ soit élémentaire. ϕ' est un homomorphisme élémentaire ordonné de $T(\Sigma)$ dans $T(\hat{\Sigma})$ défini par : $\forall\ \gamma \in \Sigma\ /\ \{\sigma\}$ on a $\sigma'(\gamma) = \gamma$ et $\phi'(\sigma(x_1, \dots x_n)) = \sigma_1(x_1, \dots x_k, \sigma_2(x_{k+1}, \dots x_{k+p}), x_{k+p+1}, \dots x_n)$.

π est un démarquage propre de $T(\hat{\Sigma})$ dans $T(\Delta)$ défini par :

$\forall\ \gamma \in \Sigma\ /\ \{\sigma\}$ on a $\pi(\gamma) = \phi(\sigma)$ et $\pi(\sigma_1) = \delta_1$, $\pi(\sigma_2) = \delta_2$.

π est bien un démarquage propre car $\forall\ \gamma \neq \sigma$, $\nu(\phi(\gamma)) = 1$.

Il est clair que $\phi = \pi \circ \phi'$.

□

<u>**Proposition 10**</u> Tout homomorphisme (éventuellement linéaire, complet, strict ou ordonné) est le produit d'homomorphismes élémentaires ordonnés et d'homomorphismes alphabétiques (linéaire, complets, stricts ou ordonnés comme le premier).

<u>Preuve</u>

Il suffit d'appliquer le lemme 7 puis les lemmes 8 et 9.

□

<u>Note</u> : Si les démarquages s'évéraient encore trop complexes, on pourrait définir des démarquages simples effaçant un seul symbole d'arité 1 et laissant les autres inchangés, ces démarquages seraient alors de complexité 1.

Exemple : ϕ linéaire de $\{\alpha, \beta, \gamma, \#\}$ dans $\{\gamma, \delta, \#\}$ défini par

$\phi(\alpha(x_1, x_2, x_3, x_4)) = \gamma(\delta(x_1, x_3), \delta(x_2, x_4))$

$\phi(\beta(x_1, x_2, x_3)) = \gamma(x_1, \delta(\#, \#))$

$\phi(\gamma(x_1, x_2)) = x_1$

$\phi(\#) = \#$

ϕ_1 alphabétique linéaire de $\{\alpha, \beta, \gamma, \#\}$ dans $\{\bar{\alpha}, \bar{\beta}, \bar{\#}\}$

$\phi_1(\alpha(x_1, x_2, x_3, x_4)) = \bar{\alpha}(x_1, x_3, x_2, x_4)$ (torsion)

$\phi_1(\beta(x_1, x_2, x_3)) = \bar{\beta}(x_1)$

$\phi_1(\gamma(x_1, x_2)) = x_1$

$\phi(\#) = \bar{\#}$.

ϕ_2 élémentaire ordonné de $\{\bar{\alpha}, \bar{\beta}, \bar{\#}\}$ dans $\{\bar{\gamma}, \delta, \bar{\beta}, \bar{\#}\}$

$\phi_2(\bar{\alpha}(x_1, x_2, x_3, x_4)) \rightarrow \bar{\gamma}(\delta(x_1, x_2), x_3, x_4)$

et identité sur $\bar{\beta}$ et $\bar{\#}$.

ϕ_3 élémentaire ordonné de $\{\bar{\gamma}, \delta, \bar{\beta}, \bar{\#}\}$ dans $\{\bar{\gamma}, \delta, \gamma, \hat{\delta}, \bar{\#}\}$

$\phi_3(\bar{\beta}(x_1)) = \gamma(x_1, \hat{\delta})$

et identité sur $\bar{\gamma}$, δ, $\bar{\#}$.

ϕ_4 démarquage propre de $\{\bar{\gamma}, \delta, \gamma, \hat{\delta}, \bar{\#}\}$ dans $\{\bar{\gamma}, \delta, \gamma, \hat{\delta}, \#\}$

$\phi_4(\bar{\#}) = \#$ et identité sur les autres symboles.

ϕ_5 élémentaire ordonné de $\{\bar{\gamma}, \delta, \gamma, \hat{\delta}, \#\}$ dans $\{\bar{\gamma}, \delta, \gamma, \bar{\delta}, \hat{\#}, \#\}$

$\phi_5(\hat{\delta}) = \bar{\delta}(\hat{\#})$ et identité ailleurs.

ϕ_6 démarquage propre de $\{\bar{\gamma}, \delta, \gamma, \bar{\delta}, \hat{\#}, \#\}$ dans $\{\bar{\gamma}, \delta, \gamma, \bar{\delta}, \#\}$

$\phi_6(\hat{\#}) = \#$ et identité ailleurs.

ϕ_7 élémentaire ordonné de $\{\bar{\gamma}, \delta, \gamma, \bar{\delta}, \#\}$ dans $\{\bar{\gamma}, \delta, \gamma, \delta_1, \#_1, \#\}$

$\phi_7(\bar{\delta}(x_1)) = \delta_1(x_1, \#_1)$ et identité ailleurs.

ϕ_8 démarquage propre de $\{\bar{\gamma}, \delta, \gamma, \delta_1, \#_1, \#\}$ dans $\{\bar{\gamma}, \delta, \gamma, \#\}$

$\phi_8(\delta_1(x_1, x_2)) = \delta(x_1, x_2)$

$\phi_8(\#_1) = \#$ et identité ailleurs.

ϕ_9 élémentaire ordonné de $\{\bar{\gamma}, \delta, \gamma, \#\}$ dans $\{\gamma_1, \delta_2, \delta, \gamma, \#\}$

$\phi_9(\bar{\gamma}(x_1, x_2, x_3)) = \gamma_1(x_1, \delta_2(x_1, x_2))$

et identité ailleurs.

ϕ_{10} démarquage propre de $\{\gamma_1, \delta_2, \delta, \gamma, \#\}$ dans $\{\gamma, \delta, \#\}$

$\phi_{10}(\gamma_1(x_1, x_2)) = \gamma(x_1, x_2)$

$\phi_{10}(\delta_2(x_1, x_2)) = \delta(x_1, x_2)$ et identité ailleurs.

Alors $\phi = \phi_{10} \circ \phi_9 \circ \phi_8 \circ \phi_7 \circ \phi_6 \circ \phi_5 \circ \phi_4 \circ \phi_3 \circ \phi_2 \circ \phi_1$

4) SUBSTITUTIONS

Nous commencerons par situer le problème de la définition des substitutions dans un cadre algébrique. Après quoi nous reprendrons ce problème de façon plus descriptive.

Le point de vue algébrique

Σ et Δ étant deux alphabets gradués, nous nous intéressons ici à des applications de $T(\Sigma)$ dans $P_D(T(\Delta))$ ou dans $P_A(T(\Delta))$ que nous voulons définir comme extensions d'applications définies sur Σ. C'est la démarche qui consiste à construire l'homomorphisme ayant pour source une structure libre comme $T(\Sigma)$.

a) Substitutions descendantes

$T(\Sigma)$ étant le magmoïde projetable libre engendré par Σ, toute application σ de Σ dans le magmoïde projetable $P_D(T(\Delta))$ s'étend de façon unique en un homomorphisme σ_D de magmoïdes projetables qui soit l'identité sur les torsions. σ_D sera une substitution descendante.

b) Substitutions ascendantes

$P_A(T(\Delta))$ est un magmoïde non projetable. On ne peut donc étendre de façon canonique une application de Σ dans $P_A(T(\Delta))$ en un homomorphisme de $T(\Sigma)$ dans $P_A(T(\Delta))$. Le magmoïde libre engendré par Σ est $\tilde{T}(\Sigma)$, toute application σ de Σ dans $P_A(T(\Delta))$ s'étend donc de façon unique en un homomorphisme σ_A de $\tilde{T}(\Sigma)$ dans $P_A(T(\Delta))$. On ne dispose pas de moyen canonique d'extension à $T(\Sigma)$. Grâce à la propriété de décomposition canonique de tout $t \in T(\Sigma)$ en $\tilde{t}.\theta$ avec $\tilde{t} \in \tilde{T}(\Sigma)$ et $\theta \in \Theta$, on peut étendre σ_A en une application de $T(\Sigma)$ dans $P_A(T(\Delta))$ en posant $\sigma_A(t) = \sigma_A(\tilde{t}).\theta$. L'application σ_A ainsi définie sera appelée substitution ascendante mais ne sera pas en général un homomorphisme (cf lemme 12 et l'exemple qui le précède).

c) Composition de substitutions

Soit σ une application d'un ensemble E dans un ensemble F. En posant pour toute partie $A \in \mathcal{P}(E)$, $\sigma(A) = \{\sigma(a) \mid a \in A\}$ on étend σ à $\mathcal{P}(E)$. Les magmoïdes $P_D(T(\Sigma))$ et $P_A(T(\Sigma))$ n'étant pas libres, on ne pourra pas définir de façon canonique des substitutions sur ces magmoïdes. De fait les lemmes 13 et 14 montreront que l'extension homomorphe d'une substitution de $T(\Sigma)$ à $P_D(T(\Sigma))$ ou à $P_A(T(\Sigma))$ n'est en général pas possible. On ne saura donc pas en général associer à une substitution σ_1 de $T(\Sigma)$ dans $P_D(T(\Delta))$ ou $P_A(T(\Delta))$ et une substitution σ_2 de $T(\Delta)$ dans $P_D(T(\Gamma))$ ou $P_A(T(\Gamma))$, une substitution σ_3 de $T(\Sigma)$ dans $P_D(T(\Gamma))$ ou $P_A(T(\Gamma))$ telle que $\sigma_3 = \sigma_2 \circ \sigma_1$.

Le point de vue élémentaire

En théorie des langages on étend facilement la notion d'homomorphisme à la notion de substitution qui est un homomorphisme entre un monoïde et l'ensemble des parties d'un monoïde . En théorie des arbres il est utile d'introduire des substitutions qui sont la généralisation des substitutions de la théorie des langages. L'existence de deux magmoïdes des parties nous oblige à définir deux sortes de substitutions : les substitutions ascendantes sont des applications d'un magmoïde $T(\Sigma)$ dans l'ensemble des parties ascendantes $P_A(T(\Delta))$ d'un autre magmoïde et les substitutions descendantes sont des applications de $T(\Sigma)$ dans $P_D(T(\Delta))$. Ces substitutions seront définies par le résultat de leur application à chaque symbole de Σ. Appelons σ une telle substitution. Pour tout symbole $\alpha \in \Sigma_n$, $\sigma(\alpha)$ sera une partie de $T(\Delta)_n^1$. Comme toute partie F de $T(\Delta)_n^1$ s'identifie à la fois à un élément de $P_A(T(\Delta))_n^1 = \mathcal{P}(T(\Delta)_n^1)$ et un élément de $P_D(T(\Delta))_n^1 = [\mathcal{P}(T(\Delta)_n^1)]^1$, l'application σ de Σ dans $\mathcal{P}(T(\Delta)^1)$ ainsi définie nous permettra de définir à la fois une substitution ascendante et une substitution descendante.

<u>Exemple</u> $\Sigma_o = \{\#\}$, $\Sigma_1 = \{a\}$, $\Delta_o = \{0, 1\}$, $\Delta_2 = \{b\}$, $\sigma(\#) = \{0, 1\}$, $\sigma(a(x_1)) = (b(x_1, x_1))$.

Cette application a elle seule est insuffisante pour définir l'ensemble des transformés d'un arbre. Il est nécessaire de préciser comment on effectue les transformations. Si on opère de bas en haut, l'arbre $\begin{array}{c} a \\ | \\ \# \end{array}$ pourra donner les intermédiaires $\begin{array}{c} \sigma(a) \\ | \\ o \end{array}$ et $\begin{array}{c} \sigma(a) \\ | \\ 1 \end{array}$ après avoir transformé #, puis $\begin{array}{c} b \\ / \ \backslash \\ o \quad o \end{array}$ et $\begin{array}{c} b \\ / \ \backslash \\ 1 \quad 1 \end{array}$. Par contre si l'on opère de haut en bas, $\begin{array}{c} a \\ | \\ \# \end{array}$ donne $\begin{array}{c} b \\ / \ \backslash \\ \sigma(\#) \quad \sigma(\#) \end{array}$ ce qui permettra d'aboutir à $\begin{array}{c} b \\ / \ \backslash \\ o \quad o \end{array}$, $\begin{array}{c} b \\ / \ \backslash \\ o \quad 1 \end{array}$, $\begin{array}{c} b \\ / \ \backslash \\ 1 \quad o \end{array}$, $\begin{array}{c} b \\ / \ \backslash \\ 1 \quad 1 \end{array}$ car il n'y a pas de raison pour que deux occurrences de $\sigma(\#)$ donnent simultanément le même résultat. Les notions de substitutions ascendantes et descendantes précisent ces deux façons d'opérer les transformations.

Etant donnée l'application σ qui à tout symbole α de Σ associe $\sigma(\alpha) \subset T(\Delta)^1_{d(\alpha)}$, on définit la **substitution ascendante** σ_A de $T(\Sigma)$ dans $P_A(T(\Delta))$ en posant :

$\forall\, x_i \in X$, $\sigma_A(x_i) = \{x_i\}$.

$\forall\, \alpha \in \Sigma$, $\sigma_A(\alpha) = \sigma(\alpha)$.

Si $t = \alpha(t_1, \dots t_n)$ avec $\alpha \in \Sigma_n$ et $t_1, \dots t_n \in T(\Sigma)^1_q$

alors $\sigma_A(t) = \sigma(\alpha) \circledast \sigma_A([t_1, \dots t_n])$

avec $\sigma_A(\ t_1, \dots t_n\ ) = \{[\tau_1, \dots \tau_n] \mid \forall\, i \in [n],\ \tau_i \in \sigma_A(t_i)\}$.

Si $u = \langle q\ ;\ u_1, \dots u_p\rangle \in T(\Sigma)^p_q$

$\sigma_A(u) = \{\langle q\ ;\ \nu_1, \dots \nu_p\rangle \mid \forall\, i \in [n],\ \nu_i \in \sigma_A(u_i)\}$.

en reprenant l'exemple précédent, on a bien $\sigma_A\begin{pmatrix} a \\ | \\ \# \end{pmatrix} = \left\{ \begin{array}{c} b \\ / \ \backslash \\ o \quad o \end{array},\ \begin{array}{c} b \\ / \ \backslash \\ 1 \quad 1 \end{array} \right\}$ ce qui justifie l'appellation d'ascendante.

De même on définit la **substitution descendante** σ_D de $T(\Sigma)$ dans $P(T(\Sigma))$ en posant :

$\forall\, x_i \in X,\ \sigma_D(x_i) = \{x_i\}$

$\forall\, \alpha \in \Sigma,\ \sigma_D(\alpha) = \sigma(\alpha)$

si $t = \alpha(t_1, \ldots t_n)$ avec $\alpha \in \Sigma_n$ et $t_1, \ldots t_n \in T(\Sigma)^1_q$

alors $\sigma_D(t) = \sigma(\alpha)\ @ <q\ ;\ \sigma_D(t_1), \ldots \sigma_D(t_n)>$.

si $n = <q\ ;\ t_1, \ldots t_p> \in T(\Sigma)^p_q$

alors $\sigma_D(u) = <q\ ;\ \sigma_D(t_1), \ldots \sigma_D(t_p)>$.

on a alors $\sigma_D\left(\begin{array}{c} a \\ | \\ \# \end{array}\right) = \left\{ \begin{array}{c} b \\ / \ \backslash \\ x_1 \quad x_1 \end{array} \right\}\ @ <0\ ;\ \{0,\ 1\}> = \left\{ \begin{array}{c} b \\ / \ \backslash \\ 0 \quad 0 \end{array},\ \begin{array}{c} b \\ / \ \backslash \\ 0 \quad 1 \end{array},\ \begin{array}{c} b \\ / \ \backslash \\ 1 \quad 0 \end{array},\ \begin{array}{c} b \\ / \ \backslash \\ 1 \quad 1 \end{array} \right\}.$

Comme pour les homomorphismes, on dira que la substitution σ (ascendante ou descendante) est **linéaire** (resp. **complète, ordonnée**) si pour tout $\alpha \in \Sigma$, $\forall\, \tilde{t}.\theta \in \sigma(\alpha)$ avec $\tilde{t} \in \tilde{T}(\Delta)$ et $\theta \in \Theta$, la torsion θ est linéaire (resp. complète, ordonnée). On dit que σ est stricte si pour tout $\alpha \in \Sigma$, $\forall\, t \in \sigma(\alpha)$, $|t| > 0$.

Pour tout arbre $t \in T(\Sigma)^1_n$, $\sigma_A(t)$ et $\sigma_D(t)$ s'identifient à des ensembles d'arbres de $T(\Delta)^1_n$, en ce sens on peut les comparer. L'exemple a mis en évidence que dans le cas général ces deux ensembles sont différents. Le lemme suivant nous montre que ceci était dû à la non linéarité de σ.

Lemme 11 Pour tout $t \in T(\Sigma)$ on a $\sigma_A(t) \subseteq \sigma_D(t)$. Si σ est linéaire et si $\forall\, \alpha \in \Sigma$, $\sigma(\alpha) \neq \emptyset$ alors on a l'égalité.

Preuve

La preuve se fait par récurrence sur $|t|$ et utilise le lemme 5.

Hypothèse de récurrence : $\forall\, t$ avec $|t| < \ell$ on a $\sigma_A(t) \subseteq \sigma_D(t)$ et si σ est linéaire $\sigma_A(t) = \sigma_D(t)$.

Par définition de σ_A et σ_D à partir de σ, on a $|t| \leq 1 \Rightarrow \sigma_A(t) = \sigma_D(t)$.

Posons $|t| = \ell \geq 2$ alors $t = \alpha(t_1, \ldots t_n)$ avec $\alpha \in \Sigma_n$ et pour tout $i \in [n]$, $t_i \in T(\Sigma)$ et $|t_i| < \ell$.

$$\sigma_A(t) = \bigcup_{\tau \in \sigma(\alpha)} \{\tau \mathbin{\Theta} [\sigma_A(t_1), \ldots \sigma_A(t_n)]\}$$

$$\sigma_D(t) = \bigcup_{\tau \in \sigma(\alpha)} \{\tau \mathbin{\Theta} [\sigma_D(t_1), \ldots \sigma_D(t_n)]\}$$

par hypothèse de récurrence, pour tout $i \in [n]$, $\sigma_A(t_i) \subseteq \sigma_D(t_i)$ et on a l'égalité si σ est linéaire, d'où

$$\tau \mathbin{\Theta} [\sigma_A(t_1), \ldots \sigma_A(t_n)] \subseteq \tau \mathbin{\Theta} [\sigma_D(t_1), \ldots \sigma_D(t_n)]$$

avec l'égalité si σ est linéaire et d'après le lemme 5

$$\tau \mathbin{\Theta} [\sigma_D(t_1), \ldots \sigma_D(t_n)] \subseteq \tau \mathbin{\Theta} [\sigma_D(t_1), \ldots \sigma_D(t_n)],$$

de plus si σ est linéaire, $\tau = \tilde{\tau}.\theta$ avec $\tilde{\tau} \in \tilde{T}(\Delta)$ et θ linéaire, on a donc l'égalité d'après le lemme 5.

On en déduit immédiatement $\sigma_A(t) \subseteq \sigma_D(t)$ et on a l'égalité si σ est linéaire et si $\forall\, \alpha \in \Sigma$, $\sigma(\alpha) \neq \emptyset$ en utilisant le lemme 5.

□

Le parallèle avec la théorie des langages tendrait à suggérer une certaine compatibilité entre les substitutions et les greffes. Cependant les choses ne sont pas simples et l'exemple suivant montre que les substitutions ascendantes ne respectent pas toujours les greffes :

Posons $\Sigma_o = \{\#\}$, $\Sigma_2 = \Delta_2 = \{a\}$, $\Delta_o = \{0, 1\}$ et donnons nous l'application σ définie par $\sigma(a) = \{a\}$ et $\sigma(\#) = \{0, 1\}$.

On a alors $\sigma_A\left(\begin{array}{c} a \\ / \ \backslash \\ x_1 \quad x_1 \end{array} \cdot \#\right) = \sigma_A\left(\begin{array}{c} a \\ / \ \backslash \\ \# \quad \# \end{array}\right) = \left\{ \begin{array}{c} a \\ / \ \backslash \\ 0 \quad 0 \end{array}, \begin{array}{c} a \\ / \ \backslash \\ 0 \quad 1 \end{array}, \begin{array}{c} a \\ / \ \backslash \\ 1 \quad 0 \end{array}, \begin{array}{c} a \\ / \ \backslash \\ 1 \quad 1 \end{array} \right\}$

alors que $\sigma_A\left(\begin{array}{c} a \\ / \ \backslash \\ x_1 \quad x_1 \end{array}\right) \mathbin{\Theta} \sigma_A(\#) = \left\{ \begin{array}{c} a \\ / \ \backslash \\ x_1 \quad x_1 \end{array} \right\} \mathbin{\Theta} \{0, 1\} = \left\{ \begin{array}{c} a \\ / \ \backslash \\ 0 \quad 0 \end{array}, \begin{array}{c} a \\ / \ \backslash \\ 1 \quad 1 \end{array} \right\}$.

Ceci est dû à la torsion $<1\ ;\ x_1,\ x_1>$ non linéaire associée au symbole a dans $\begin{array}{c} a \\ / \ \backslash \\ x_1 \quad x_1 \end{array}$. On voit que l'on aurait par contre

$$\sigma_A\left(\begin{array}{c} a \\ / \ \backslash \\ x_1 \quad x_2 \end{array} . <1\ ;\ x_1,\ x_1> . \#\right) = \sigma_A\left(\begin{array}{c} a \\ / \ \backslash \\ x_1 \quad x_2 \end{array}\right) \odot \sigma_A(<1\ ;\ x_1,\ x_1> . \#).$$

<u>Lemme 12</u> Si $\vec{u} = \tilde{u}.\theta$ avec $\tilde{u} \in \tilde{T}(\Sigma)^n_p$ et $\theta \in \Theta^p_q$, θ linéaire alors $\forall\ \vec{v} \in T(\Sigma)^q$ on a $\sigma_A(\vec{u}.\vec{v}) = \sigma_A(\vec{u}) \odot \sigma_A(\vec{v})$.

<u>Preuve</u>

Par récurrence sur $|\vec{u}|$, le sup des profondeurs des composantes de $\vec{u}$.
Hypothèse de récurrence : le lemme est vrai si $|t| > \ell$; ce qui est évident si $\ell = 1$. En posant $\vec{u} = \tilde{u}.\theta = [u_1, \ldots u_n]$, puisque θ est linéaire, pour tous i et j dans $[n]$ avec $i \neq j$ on a var $(u_i) \cap$ var $(u_j) = \emptyset$ donc $\sigma_A(\vec{u}.\vec{v}) = \sigma_A(\vec{u}) \odot \sigma_A(\vec{v}) \iff \forall\ i \in [n]\ \sigma_A(u_i.\vec{v}) = \sigma_A(u_i) \odot \sigma_A(\vec{v})$.
Prenons $\vec{u}$ avec $|\vec{u}| = \ell > 1$, $\vec{u} = \tilde{u}.\theta = [u_1, \ldots u_n]$, pour chaque composante u_i avec $|u_i| = \ell > 1$ on a $u_i = \alpha.[t_1, \ldots t_r]$ avec $\alpha \in \Sigma_r$ et $[t_1, \ldots t_r]$ linéaire et de profondeur inférieure à ℓ donc par hypothèse de récurrence $\sigma_A([t_1, \ldots t_r].\vec{v}) = \sigma_A([t_1, \ldots t_r]) \odot \sigma_A(\vec{u})$. Alors $\sigma_A(u_i.\vec{v}) = \sigma_A(\alpha.[t_1, \ldots t_r].\vec{v}) = \sigma_A(\alpha) \odot \sigma_A([t_1, \ldots t_r].\vec{v})$ par définition de σ_A et $\sigma_A(u_i.\vec{v}) = \sigma_A(\alpha) \odot \sigma_A([t_1, \ldots t_r]) \odot \sigma_A(\vec{v})$ par hypothèse de récurrence. Par ailleurs on a $\sigma_A(u_i) \odot \sigma_A(\vec{v}) = \sigma_A(\alpha) \odot \sigma_A([t_1, \ldots t_r]) \odot \sigma_A(\vec{v})$.
Pour les autres composantes de u_i on a $|u_i| < \ell$ donc $\sigma_A(u_i.\vec{v}) = \sigma_A(u_i) \odot \sigma_A(\vec{v})$, par hypothèse de récurrence, donc $\sigma_A(\vec{u}.\vec{v}) = \sigma_A(\vec{u}) \odot \sigma_A(\vec{v})$.

$\square$

En posant pour toute substitution σ et toute forêt F, $\sigma(F) = \bigcup_{t\in F} \sigma(t)$ on obtient le lemme suivant qui généralise le lemme 12.

<u>Lemme 13</u> Pour toute substitution ascendante σ_A de $T(\Sigma)$ dans $P_A(T(\Delta))$ et toutes forêts $F \in T(\Sigma)$, $F_1, \dots F_n \in T(\Sigma)^1$ telles que $F \circledast [F_1, \dots F_n]$ soit défini et que $\forall\, \tilde{u}.\theta \in F$, $\tilde{u} \in \tilde{T}(\Sigma)$, θ soit linéaire, on a $\sigma_A(F \circledast [F_1, \dots F_n]) = \sigma_A(F) \circledast [\sigma_A(F_1), \dots \sigma_A(F_n)]$. Si on ne suppose pas θ linéaire, on a seulement $\supseteq$.

<u>Preuve</u>

On peut écrire $\sigma_A(F \circledast [F_1, \dots F_n]) = \bigcup_{u\in F} \sigma_A(\{u\} \circledast [F_1, \dots F_n])$ et $\sigma_A(F) \circledast [\sigma_A(F_1), \dots \sigma_A(F_n)] = \{\bigcup_{u\in F} \sigma_A(u)\} \circledast [\sigma_A(F_1), \dots \sigma_A(F_n)]$.

Il suffit alors de montrer que l'on a

$\sigma_A(\{u\} \circledast [F_1, \dots F_n]) \supseteq \sigma_A(u) \circledast [\sigma_A(F_1), \dots \sigma_A(F_n)]$ et

Si θ est linéaire on a l'égalité.

On a $\{u\} \circledast [F_1, \dots F_n] = \{u.v \mid v \in [F_1, \dots F_n]\}$

et $[\sigma_A(F_1), \dots \sigma_A(F_n)] = \bigcup_{v\in[F_1,..F_n]} \sigma_A(\vec{v})$.

Il suffit alors de montrer que $\forall\, v \in [F_1, \dots F_n]$ on a $\sigma_A(u.v) \supseteq \sigma_A(u) \circledast \sigma_A(v)$ et si θ est linéaire on a l'égalité en vertu du lemme 12.

□

Les substitutions descendantes de $T(\Sigma)$ dans $P(T(\Delta))$ sont bien des homomorphismes de magmoïdes, en effet on montre facilement par induction sur $|t|$ que $\sigma_D(t.\vec{u}) = \sigma_D(t) \circledast \sigma_D(\vec{u})$ et comme $\sigma_D(x_i) = \{x_i\}$ que les substitutions descendantes respectent les torsions et le produit tensoriel.

La généralisation de σ_D à une forêt doit être menée avec précaution, en effet, en général, $\sigma_D(F \circledast [F_1, \dots F_n]) \neq \sigma_D(F) \circledast [\sigma_D(F_1), \dots \sigma_D(F_n)]$.

L'exemple suivant le montrera :

$\Sigma_o = \Delta_o = \{0, 1\}$, $\Sigma_1 = \{a\}$, $\Delta_2 = \{b\}$ et σ_D est définie par

$\sigma_D(0) = \{0\}$, $\sigma_D(1) = \{1\}$, $\sigma_D(a(x_1)) = \{b(x_1, x_1)\}$.

On a alors $\sigma_D(\{a(x_1)\} @ \{0, 1\}) = \sigma_D(\{\begin{matrix} a \\ | \\ 0 \end{matrix}, \begin{matrix} a \\ | \\ 1 \end{matrix}\}) = \sigma_D(\begin{matrix} a \\ | \\ 0 \end{matrix}) \cup \sigma_D(\begin{matrix} a \\ | \\ 1 \end{matrix}) = \{ \begin{matrix} & b & \\ / & & \backslash \\ 0 & & 0 \end{matrix} , \begin{matrix} & b & \\ / & & \backslash \\ 1 & & 1 \end{matrix} \}$.

Par contre $\sigma_D(a(x_1)) @ \sigma_D(\{0, 1\}) = \{ \begin{matrix} & b & \\ / & & \backslash \\ x_1 & & x_1 \end{matrix} \} @ \{0, 1\} = \{ \begin{matrix} & b & \\ / & & \backslash \\ 0 & & 0 \end{matrix} , \begin{matrix} & b & \\ / & & \backslash \\ 0 & & 1 \end{matrix} , \begin{matrix} & b & \\ / & & \backslash \\ 1 & & 0 \end{matrix} , \begin{matrix} & b & \\ / & & \backslash \\ 1 & & 1 \end{matrix} \}$.

Ceci est dû à la non linéarité de σ_D.

<u>Lemme 14</u> Pour toute substitution descendante σ_D et toutes forêts $F \in T(\Sigma)^m_n$, $F_1, \ldots F_n \in T(\Sigma)^1_p$ on a $\sigma_D(F @ [F_1, \ldots F_n]) \subseteq \sigma_D(F) @ [\sigma_D(F_1), \ldots \sigma_D(F_n)]$. Si σ_D est linéaire on a l'égalité.

<u>Preuve</u>

Si $\omega \in \sigma_D(F @ [F_1, \ldots F_n])$ il existe $\tilde{u}.\theta \in F$ avec $\tilde{u} \in \tilde{T}(\Sigma)^m_q$ et $\theta \in \Theta^q_n$ et $\vec{v} = [v_1, \ldots v_q]$ tels que pour tout $i \in [q]$, $v_i \in F_{\theta(i)}$ et on a

$\omega \in \sigma_D(\tilde{u}.\vec{v}) = \sigma_D(\tilde{u}) @ [\sigma_D(v_1), \ldots \sigma_D(v_q)]$ de plus

$[\sigma_D(v_1), \ldots \sigma_D(v_q)] \subset [\sigma_D(F_{\theta(1)}), \ldots \sigma_D(F_{\theta(q)})]$

$$\begin{aligned} \text{d'où } \sigma_D(\tilde{u}.\vec{v}) &\subset \sigma_D(\tilde{u}) @ \{\theta\} @ [\sigma_D(F_1), \ldots \sigma_D(F_n)] \\ &= \sigma_D(\tilde{u}.\theta) @ [\sigma_D(F_1), \ldots \sigma_D(F_n)] \\ &\subset \sigma_D(F) @ [\sigma_D(F_1), \ldots \sigma_D(F_n)] \end{aligned}$$

d'où $\sigma_D(F @ [F_1, \ldots F_n]) \subset \sigma_D(F) @ [\sigma_D(F_1), \ldots \sigma_D(F_n)]$.

Si $\omega \in \sigma_D(F) @ [\sigma_D(F_1), \ldots \sigma_D(F_n)]$ alors $\omega = \tilde{\nu}.[\omega_1, \ldots \omega_q]$ avec $\tilde{\nu} \in \tilde{T}(\Delta)^m_q$ et il existe $\theta \in \Theta^q_n$ tel que pour tout $i \in [q], \omega_i \in \sigma_D(F_{\theta(i)})$ et $\tilde{\nu}.\theta \in \sigma_D(F)$. Posons $u \in F$ tel que $\tilde{\nu}.\theta \in \sigma_D(n)$. En posant $u = \tilde{u}.\theta'$, si σ_D est linéaire, $\tilde{\nu} \in \sigma_D(\tilde{u})$, il suffit alors de prendre pour tout $i \in [q]$, $v_i \in F_i$ tel que $\omega_i \in \sigma_D(v_i)$. On a alors $\omega \in \sigma_D(\tilde{u}.[v_1, \ldots v_q])$ et $\tilde{u}.[v_1, \ldots v_q] \in F @ [F_1, \ldots F_n]$.

□

On trouvera ce dernier lemme exposé dans le cadre plus général des k-substitutions dans [D-te]. D'une façon générale, les problèmes liés aux substitutions sont **traités** dans le cadre des k-substitutions et de façon plus complète qu'ici dans [A-te], [D, te].

Une dernière difficulté est, qu'en général, le produit de deux substitutions n'est pas une substitution. Ce fait est illustré par les deux exemples qui suivent.

Posons $\Sigma_o = \Delta_o = \{\#\}$, $\Gamma_o = \{+, -\}$, $\Sigma_1 = \{a\}$, $\Delta_2 = \{b\}$, $\Gamma_4 = \{\gamma\}$ et les substitutions ascendantes σ_1 de $T(\Sigma)$ dans $P_A(T(\Delta))$ et σ_2 de $T(\Delta)$ dans $P_A(T(\Gamma))$ définies par $\sigma_1(\#) = \{\#\}$, $\sigma_1(a(x_1)) = \{b(x_1, x_1)\}$, $\sigma_2(\#) = \{+,-\}$, $\sigma_2(b(x_1,x_2))= \{\gamma(x_1, x_1, x_2, x_2)\}$. A tout arbre t de $T(\Sigma)$, σ_1 fait correspondre l'ensemble $\sigma_1(t) \subset T(\Delta)$, en appliquant σ_2 à cet ensemble on obtient $\sigma_2(\sigma_1(t)) \subset T(\Gamma)$.

Ainsi $\sigma_1(a(\#)) = \{b(\#, \#)\}$

et $\sigma_2(b(\#, \#)) = \{ \begin{array}{c}\gamma\\ //\ \backslash\backslash \\ +\ +\ +\ +\end{array}, \begin{array}{c}\gamma\\ //\ \backslash\backslash \\ +\ +\ -\ -\end{array}, \begin{array}{c}\gamma\\ //\ \backslash\backslash \\ -\ -\ +\ +\end{array}, \begin{array}{c}\gamma\\ //\ \backslash\backslash \\ -\ -\ -\ -\end{array} \} = \sigma_2(\sigma_1(\begin{array}{c}a\\ | \\ \#\end{array}))$

Il est clair que si $\forall\ t \in T(\Sigma)$, $\sigma_3(t) = \sigma_2(\sigma_1(t))$ il faut que $\sigma_3(\#) = \{+,-\}$ et $\sigma_3(a(x_1)) = \{\gamma(x_1, x_1, x_1, x_1)\}$ ce qui entraine que

$\sigma_3(\begin{array}{c}a\\ | \\ \#\end{array}) = \{ \begin{array}{c}\gamma\\ /\ /\ \backslash\ \backslash \\ a_1\ a_2\ a_3\ a_4\end{array} \mid \forall\ i \in [4],\ a_i \in \{+,-\}\} \neq \sigma_2(\sigma_1(\begin{array}{c}a\\ | \\ \#\end{array}))$.

Le second exemple mettra en jeu des substitutions descendantes.

$\Sigma_o = \{\#\}$, $\Delta_o = \{0, 1\}$, $\Gamma_o = \{+,-,*\}$, $\Sigma_1 = \Delta_1 = \{a\}$, $\Gamma_2 = \{b\}$.

$\sigma_1(a) = \{a\}$, $\sigma_1(\#) = \{0, 1\}$, $\sigma_2(a(x_1)) = \{b(x_1, x_1)\}$,

$\sigma_2(0) = \{+,-\}$ et $\sigma_2(1) = \{*\}$.

On a alors $\sigma_1(\begin{array}{c}a\\ | \\ \#\end{array}) = \{\begin{array}{c}a\\ | \\ 0\end{array}, \begin{array}{c}a\\ | \\ 1\end{array}\}$

$$\sigma_2\Big(\begin{array}{c}a\\|\\o\end{array}\Big) = \Big\{ \begin{array}{c}b\\ /\;\backslash\\ +\quad +\end{array} , \begin{array}{c}b\\ /\;\backslash\\ +\quad -\end{array} , \begin{array}{c}b\\ /\;\backslash\\ -\quad +\end{array} , \begin{array}{c}b\\ /\;\backslash\\ -\quad -\end{array} \Big\} \text{ et } \sigma_2\Big(\begin{array}{c}a\\|\\1\end{array}\Big) = \Big\{ \begin{array}{c}b\\ /\;\backslash\\ *\quad *\end{array} \Big\}$$

d'où $$\sigma_2\Big(\sigma_1\Big(\begin{array}{c}a\\|\\\#\end{array}\Big)\Big) = \Big\{ \begin{array}{c}b\\ /\;\backslash\\ +\quad +\end{array} , \begin{array}{c}b\\ /\;\backslash\\ +\quad -\end{array} , \begin{array}{c}b\\ /\;\backslash\\ -\quad +\end{array} , \begin{array}{c}b\\ /\;\backslash\\ -\quad -\end{array} , \begin{array}{c}b\\ /\;\backslash\\ *\quad *\end{array} \Big\}.$$

Il est clair que pour avoir tout $t \in T(\Sigma)$, $\sigma_3(t) = \sigma_2(\sigma_1(t))$, il faut avoir

$\sigma_3(\#) = \{+, -, *\}$ et $\sigma_3\Big(\begin{array}{c}a\\|\\x_1\end{array}\Big) = \Big\{ \begin{array}{c}b\\ /\;\backslash\\ x_1\quad x_1\end{array} \Big\}$ ce qui entraine

$$\sigma_3\Big(\begin{array}{c}a\\|\\\#\end{array}\Big) = \Big\{ \begin{array}{c}b\\ /\;\backslash\\ a_1\quad a_2\end{array} \;\Big|\; a_1, a_2 \in \{+, -, *\}\Big\} \neq \sigma_2\Big(\sigma_1\Big(\begin{array}{c}a\\|\\\#\end{array}\Big)\Big).$$

B - GRAMMAIRES ALGÉBRIQUES

1) DEFINITION DES GRAMMAIRES ET FORETS ALGEBRIQUES

Etant donnés un alphabet gradué Σ de symboles **terminaux**, un alphabet gradué V de symboles **non-terminaux**, disjoint de Σ, un symbole X_o de V_o appelé **axiome** et un ensemble fini R de **règles** de la forme $(X \rightarrow \tau)$ avec $X \in V$ et $\tau \in T(\Sigma \cup V)^1_{d(X)}$, on appelle **grammaire algébrique** G le quadruplet $<V, \Sigma, X_o, R>$.

On dit qu'un arbre $t \in T(\Sigma \cup V)^1_n$ se **dérive** en $t' \in T(\Sigma \cup V)^1_n$ par application de la règle $r = (X \rightarrow \tau) \in R$ et on note $t \Rightarrow t'$ ou $t \underset{G}{\Rightarrow} t'$ s'il est utile de rappeler la grammaire ou $t \underset{r}{\Rightarrow} t'$ s'il est utile de préciser la règle, si et seulement si t contient une occurrence de X qui est remplacée par τ pour former t', c'est à dire $t = \tilde{u}.[\vec{v}_1, X.\vec{w}, \vec{w}_2]$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)^1_p$; $[\vec{v}_1, X.\vec{w}, \vec{v}_2] \in T(\Sigma \cup V)^p_n$, $\vec{w} \in T(\Sigma \cup V)^{d(X)}$ et $t' = \tilde{u}.[\vec{v}_1, \tau.\vec{w}, \vec{v}_2]$. On dit alors que l'occurrence de X a été **réécrite** ou **dérivée**. Pour éviter certaines confusions on dit parfois que t se dérive **directement** en t'. Rappelons qu'il est essentiel que $\tilde{u}$ soit initial pour être certain de mettre en évidence une et une seule occurrence de X. Dans le cas où $t \in T(\Sigma \cup V)^1_o$, la description de l'application d'une règle $X \rightarrow \tau$ peut se simplifier sans ambiguité en posant $t = \tilde{u}.X.\vec{w} \Rightarrow t' = \tilde{u}.\tau.\vec{w}$.

On dit de plus que la dérivation est **descendante** et on note $t \overset{D}{\Rightarrow} t'$ si de plus $\tilde{u} \in \tilde{T}(\Sigma)^1_p$, c'est à dire si l'occurrence de X dérivée n'est située sous aucun arbre non-terminal. Par contre on dit que la dérivation est **ascendante** et on note $t \overset{A}{\Rightarrow} t'$ si $\vec{w} \in T(\Sigma)^{d(x)}$ c'est à dire si l'occurrence de X dérivée n'est située au dessus d'aucun autre non-terminal. Les dérivations descendantes sont parfois appelées OI-dérivations et les ascendantes IO-dérivations [En].

On dit que t se dérive en t' par une dérivation de **longueur** ℓ dans G (resp de façon descendante ou ascendante) et on note $t \overset{\ell}{\underset{G}{=>}} t'$ (resp. $t \overset{\ell}{\underset{G}{D=>}} t'$ ou $t \overset{\ell}{\underset{G}{A=>}} t'$) s'il existe des arbres $t_o, t_1, \dots t_\ell$ tels que $t_o = t$, $t_\ell = t'$ et pour tout $i \in [\ell]$, $t_{i-1} \underset{G}{=>} t_i$ (resp $t_{i-1} \underset{G}{D=>} t_i$ ou $t_{i-1} \underset{G}{A=>} t_i$). S'il n'est pas nécessaire de préciser ℓ on notera $t \overset{*}{\underset{G}{=>}} t'$ (resp. $t \overset{*}{\underset{G}{D=>}} t'$ ou $t \overset{*}{\underset{G}{A=>}} t'$) et pour tout arbre $t \in T(\Sigma \cup V)^1_n$ on a $t \overset{o}{\underset{G}{=>}} t$ (resp $t \overset{o}{\underset{G}{D=>}} t$ et $t \overset{o}{\underset{G}{A=>}} t$). Si l'on veut seulement préciser que ℓ n'est pas nul on notera $t \overset{+}{\underset{G}{=>}} t'$ (resp $t \overset{+}{\underset{G}{D=>}} t'$ ou $t \overset{+}{\underset{G}{A=>}} t'$).

On appelle **forêt algébrique engendrée** par G l'ensemble $F(G) = \{t \in T(\Sigma) \mid X_o \overset{*}{\underset{G}{=>}} t\}$. Quelques notations usuelles sont, pour tout $\tau \in T(\Sigma \cup V)$

$F(G, \tau) = \{t \in T(\Sigma) \mid \tau \overset{*}{\underset{G}{=>}} t\}$

$F_{IO}(G, \tau) = \{t \in T(\Sigma) \mid \tau \overset{*}{\underset{G}{A=>}} t\}$

$F_{OI}(G, \tau) = \{t \in T(\Sigma) \mid \tau \overset{*}{\underset{G}{D=>}} t\}$.

Avec ces notations on a $F(G) = F(G, X_o)$ et on pose $F_{IO}(G) = F_{IO}(G, X_o)$ la **forêt algébrique ascendante** (ou IO) engendrée par G et $F_{OI}(G) = F_{OI}(G, X_o)$ la **forêt algébrique descendante** (ou OI) engendrée par G. Puisque toute dérivation ascendante (resp descendante) est une dérivation particulière on a $F_{IO}(G, \tau) \subseteq F(G, \tau)$ (resp $F_{OI}(G, \tau) \subseteq F(G, \tau)$).

<u>Exemple G1</u> : $\Sigma_o = \{0\}$, $\Sigma_1 = \{S\}$, $\Sigma_2 = \{*\}$,

$V_o = \{X_o\}$, $V_1 = \{X\}$.

$R = \{(X_o \to X(0)), (X(x_1) \to X(S(x_1))), (X(x_1) \to *(x_1, x_1))\}$

que l'on écrit le plus souvent

$X_o \to X(0)$, $X(x_1) \to X(S(x_1)) + *(x_1, x_1)$

le signe + indiquant le choix entre plusieurs parties droites.

Il est alors clair que si $X_o \Rightarrow t$ alors $t = X(0)$.

Si $X(0) \overset{\ell}{\Rightarrow} t$ en appliquant la seule règle $X(x_1) \rightarrow X(S(x_1))$ alors $t = X\, S^{\ell}\, 0$.

Et $X\, S^{\ell}\, 0 \Rightarrow *(S^{\ell}\, 0,\ S^{\ell}\, 0)$ en appliquant la règle $X(x_1) \rightarrow *(x_1, x_1)$. Toute dérivation $X_o \overset{*}{\Rightarrow} t$ terminale c'est à dire telle que $t \in T(\Sigma)$ est de la forme qui vient d'être décrite, on remarque qu'elle est à la fois ascendante et descendante car aucune partie droite de règle n'a deux symboles non terminaux sur une même branche, on obtient donc

$F(G_1) = F_{IO}(G_1) = F_{OI}(G_1) = \{*(S^{\ell}\, 0,\ S^{\ell}\, 0) \mid \ell \in \mathbb{N}\}$.

Si l'on interprête $*$ comme la multiplication, S comme la fonction successeur dans $\mathbb{N}$ et 0 comme la constante zéro, l'ensemble des arbres de $F(G_1)$ calcule l'ensemble des carrés d'entiers.

<u>Exemple G2</u> : $\Sigma_o = \{0\}$, $\Sigma_1 = \{S\}$, $\Sigma_2 = \{*\}$,

$V_o = \{X_o\}$, $V_1 = \{X, Y\}$

$R = \{X_o \rightarrow Y(X(0)),\ Y(x_1) \rightarrow *(x_1, x_1),\ X(x_2) \rightarrow x_1 + X(S(x_1))\}$

Il est clair que $F(G_2) = F(G, Y(X(0)))$ et qu'il en est de même en IO et OI.

Si $Y(t) \overset{*}{\Rightarrow} \tau \in T(\Sigma)$ on a $\tau = *(t_1, t_2)$ avec $t_1, t_2 \in F(G_2, t)$ et il en est de même en OI.

On a de plus $F(G2, X(0)) = F(G1, X(0)) = \{S^{\ell}\, 0 \mid \ell \in \mathbb{N}\}$ et il en est de même en OI. On obtient alors

$F(G2) = \{*(t_1, t_2) \mid t_1, t_2 \in F(G_2, X(0))\} = \{*(S^{\ell_1}\, 0,\ S^{\ell_2}\, 0) \mid \ell_1, \ell_2 \in \mathbb{N}\}$

et $F_{OI}(G2) = F(G2)$. Par contre pour les dérivations ascendantes si $Y(t) \overset{*}{A\Rightarrow} \tau \in T(\Sigma)$ alors $Y(t) A\Rightarrow Y(t')$ avec $t' \in F_{IO}(G2, t)$ puis $Y(t') A\Rightarrow *(t', t')$. Comme on a $F_{IO}(G2) = F_{IO}(G2, Y(X(0)))$ et

$F_{IO}(G2, X(0)) = F(G2, X(0))$ on obtient

$F_{IO}(G2) = \{*(t', t') \mid t' \in F_{IO}(G2, X(0))\} = \{*(S^{\ell}0,\ S^{\ell}0) \mid \ell \in \mathbb{N}\} \subsetneq F(G2)$.

<u>Exemple G3</u> : $X_o \to X(0,\ 0),\ X(x_1,\ x_2) \to X(S(x_1),\ S(x_2)) + *(x_1,\ x_2)$
on a alors $F(G3) = F_{IO}(G3) = F_{OI}(G3) = F(G1)$.

<u>Exemple G4</u> : $X_o \to X(0,\ 0),\ X(x_1,\ x_2) \to X(S(x_1),\ x_2) + X(x_1,\ S(x_2)) + *(x_1,\ x_2)$
on a, dans ce cas, $F(G4) = F_{IO}(G4) = F_{OI}(G4) = F(G2)$.

L'exemple G5 met en lumière une application de grammaire algébrique d'arbre,utile pour définir et énumérer un ensemble d'arbres connu, celui des arbres AVL.

<u>Exemple G5</u> : $\Sigma_o = \{\square\},\ \Sigma_2 = \{*\},\ V_o = \{X_o\},\ V_2 = \{X,\ Y\}$.
Les règles sont : $X_o \to Y(\square,\ \square) + \square,\ Y(x_1,\ x_2) \to Y(X(x_1,\ x_2),\ x_1) + X(x_1,\ x_2)$
et $X(x_1,\ x_2) \to *(x_1,\ x_2) + *(x_1,\ x_1) + *(x_2,\ x_1)$
Notons $f_o,\ f_1,\ f_2$ ... les arbres de Fibbonacci $\square$, $\begin{array}{c} X \\ / \ \backslash \\ \square \quad \square \end{array}$, $\begin{array}{c} X \\ / \ \backslash \\ \begin{array}{c} X \\ / \ \backslash \\ \square \quad \square \end{array} \quad \square \end{array}$ ...

définis pour tout entier $i \geq 2$ par $f_i = \begin{array}{c} X \\ / \ \backslash \\ f_{i-1} \quad f_{i-2} \end{array}$

Il est alors clair que si $\begin{array}{c} Y \\ / \ \backslash \\ f_{i+1} \quad f_i \end{array}$ D=> t alors $t \in \{ \begin{array}{c} Y \\ / \ \backslash \\ f_{i+2} \quad f_{i+1} \end{array} ,\ f_{i+2}\}$

d'où $\{t \mid X_o \overset{*}{D\Rightarrow} t$ et racine $(t) = X\} = \{f_1,\ f_2,\ f_3,\ \dots\}$.
De plus on a X_o D=> $\square = f_o$.
On définit sur Σ l'ensemble des arbres AVL par :
$AVL_o = \{\square\},\ AVL_1 = \{ \begin{array}{c} * \\ / \ \backslash \\ \square \quad \square \end{array} \}$ et pour tout $i \geq 2$

$AVL_i = \{ \begin{array}{c} * \\ / \ \backslash \\ t_1 \quad t_2 \end{array} \mid (t_1 \text{ et } t_2 \in AVL_{i-1} \cup AFL_{i-2}) \text{ et } (t_1 \text{ ou } t_2 \in AVL_{i-1})\}$

$AVL = \bigcup_{i \in \mathbb{N}} AVL_i$.

Par récurrence sur i on montre alors que $AVL_i = F_{OI}(G, f_i)$ et par là que $F_{OI}(G5) = AVL$.

On a l'habitude de classer les grammaires algébriques selon certaines propriétés de leurs règles ou de leurs alphabets. Les forêts algébriques hériteront alors de ces propriétés en ce sens qu'on dira qu'une forêt est algébrique (resp ascendante ou descendante) et a la propriété P s'il existe une grammaire possédant la propriété P et qui l'engendre (par dérivations resp ascendantes ou descendantes).

Une grammaire est **régulière** si $V = V_o$, c'est à dire que tous les non terminaux sont d'arité 0. Un non terminal est donc toujours une feuille, ce qui entraine que sur chaque branche les dérivations fonctionnent de la même façon que les dérivations dans une grammaire de mots linéaire. Aucun non terminal ne peut se trouver au dessus d'un arbre, on a donc $F(G) = F_{IO}(G) = F_{OI}(G)$. Les forêts régulières, encore appelées rationnelles sont parmi les premières à avoir été étudiées à cause des propriétés de leurs feuillages. Si on appelle feuillage d'une forêt F le langage $\{feuillage(t) \mid t \in F\}$ alors la famille des feuillages des forêts rationnelles s'identifie à la famille des langages algébriques [Pa], [Ro2],[Th] alors que la famille des feuillages des forêts algébriques est la famille des langage indexés.

Une grammaire est **corégulière** si chaque règle est de l'une des formes $(X \to Y.\vec{v})$ avec $X, Y \in V$ et $\vec{v} \in T(\Sigma)$ ou $(Y \to T)$ avec $Y \in V$ et $t \in T(\Sigma)$. Tout arbre obtenu par dérivation depuis n'importe quel non terminal, ou bien est terminal ou bien possède une seule occurrence de non terminal qui est sa racine. Toute dérivation est alors à la fois ascendante et descendante et on a encore $F(G) = F_{IO}(G) = F_{OI}(G)$. Les grammaires G3 et G4 sont corégulières. Les forêts corégulières ont été étudiées par Arnold et Dauchet qui ont montré que $T(\Sigma)^1_o$ qui est une forêt rationnelle n'était pas corégulière [AD-c].

Une grammaire est **stricte** si toutes ses règles sont strictes, c'est à dire s'il n'existe pas de règle de la forme $(X \to x_i)$ avec $i \in d[X]$. Une telle règle non stricte est à rapprocher des règles effaçantes des grammaires de mots en ce sens que l'application d'une règle non stricte diminue la longueur de certaines branches. La grammaire G2 n'est pas stricte, G1, G3, G4 et G5 le sont.

Une grammaire est à **non strictitude monadique** si toute règle non stricte est de la forme $(X \to x_1)$ avec $X \in V_1$. Il s'ensuit que pour tout non terminal $X \notin V_1$ il n'existe pas $x_i \in F(G, X)$. Ceci est un cas très particulier de grammaire non stricte qui recouvre encore la non strictitude dans les langages où tous les symboles peuvent être considérés comme d'arité 1. Nous éprouvons le besoin de définir spécialement ce cas particulier car nous verrons que si la non-strictitude n'est pas réductible en général, elle le devient si elle est monadique.

Une grammaire est **Greibach-haut**, et parfois on précise stricte, si toutes les règles sont de la forme $(X \to \alpha.\vec{v})$ avec $\alpha \in \Sigma$. On dit aussi parfois seulement forme de Greibach pour forme de Greibach-haut, cette appellation recouvrant souvent même les grammaires de Greibach-haut non strictes, dans lesquelles on accepte les règles non strictes. La raison est que ces grammaires correspondent aux schémas de programmes les plus étudiés, dans lesquels aucune définition de procédure ne commence par un appel de procédure.

Une grammaire est **greibach-bas** si toutes ses règles sont de la forme $(X \to t.\vec{v})$ avec $t \in T(\Sigma \cup V)$, $\vec{v} = [t_1, \dots t_n] \in T(\Sigma)$ et pour tout $i \in [n]$ on a $|t_i| > 0$. Les grammaires G1 et G3 sont de Greibach-bas.

Une grammaire est **sans duplication** ou **linéaire** si toutes les règles peuvent s'écrire $(X \to \tilde{t}.\theta)$ avec $\tilde{t} \in \tilde{T}(\Sigma \cup V)^1_p$, $\theta \in \Theta^p_{d(X)}$ avec θ linéaire. Les grammaires G3 et G4 sont linéaires.

Une grammaire est **complète** si toutes les règles peuvent s'écrire $(X \to \tilde{t}.\theta)$ avec $\tilde{t} \in \tilde{T}(\Sigma \cup V)^1_p$, $\theta \in \Theta^p_{d(X)}$ avec θ complète. Dans les cinq exemples de grammaire qui précèdent, seule G5 n'est pas complète.

Une grammaire est **ordonnée** si toutes les règles peuvent s'écrire $(X \to \tilde{t}.\theta)$ avec $\tilde{t} \in \tilde{T}(\Sigma \cup V)^1_p$, $\theta \in \Theta^p_{d(X)}$ avec θ ordonnée.

Ces diverses propriétés permettent de définir un grand nombre de sous familles parmi les forêts algébriques. Il est connu que dans le cas des grammaires de mots, où les mots peuvent être considérés comme des arbres à une seule branche, ces propriétés permettraient seulement de distinguer la classe des rationnels, strictement incluse dans les algébriques. En théorie des arbres, les choses sont plus complexes et la suite de ce travail apportera certaines connaissances nouvelles sur ce point.

2) PROPRIETES DES GRAMMAIRES ALGEBRIQUES

Lemme 15 (de contexte libre) Pour toute dérivation $t \overset{*}{\Rightarrow} t'$ dans $G = \langle V, \Sigma, X_o, R\rangle$ et pour toute décomposition $t = u.[u_1, \dots u_p]$ on peut trouver une décomposition $t' = u'.[u'_1, \dots u'_n]$ et une torsion $\theta \in \Theta_p^n$ telles que pour tout $i \in [n]$, $u_{\theta(i)} \underset{G}{\overset{*}{\Rightarrow}} u'_i$ et $u \underset{G}{\overset{*}{\Rightarrow}} u'.\theta$. Réciproquement l'existence de ces deux décompositions entraine $t \underset{G}{\overset{*}{\Rightarrow}} t'$.

Preuve

Par récurrence sur la longueur des dérivations.

Le lemme est trivialement vrai si $t \overset{o}{\Rightarrow} t'$.

Hypothèse de récurrence : le lemme est vrai si $t \overset{n}{\Rightarrow} t'$ avec $n < \ell$.

Soit $t \overset{*}{\Rightarrow} t'' \Rightarrow t'$ une dérivation de longueur ℓ.

Par hypothèse de récurrence, pour toute décomposition $t = u.[u_1, \dots u_p]$ il existe une décomposition $t'' = u''.[u''_1, \dots u''_q]$ et une torsion $\theta'' \in \Theta_p^q$ telles que pour tout $i \in [q]$, $u_{\theta''(i)} \overset{*}{\Rightarrow} u''_i$ et $u \overset{*}{\Rightarrow} u''.\theta''$.

Posons $u'' = \tilde{u}''.\theta'$ avec $\tilde{u}'' \in \mathcal{T}(\Sigma \cup V)$ et $\theta' \in \Theta_q^n$ (proposition 2).

Pour la dérivation $t'' \Rightarrow t'$, deux cas sont à considérer selon que l'occurrence de non terminal réécrite est dans u'' ou pas :

ou bien $u'' \Rightarrow u'$ alors $t' = u'.[u''_1, \dots u''_q]$ et dans ce cas $u \overset{*}{\Rightarrow} u''.\theta'' \Rightarrow u'.\theta''$ et la décomposition de t' est trouvée.

Ou bien $t'' = \tilde{u}''.\theta'.[u''_1, \dots u''_q] = \tilde{u}''.[u''_{\theta'(1)}, \dots u''_{\theta'(n)}]$, $t' = u''.[u'_1, \dots u'_n]$ et il existe $i \in [n]$ tel que $u''_{\theta'(i)} \Rightarrow u'_i$ et pour tout $j \in [n]$ avec $j \neq i$, $u'_j \Rightarrow u''_{\theta'(j)}$. On a alors $\forall\, i \in [n]$, $u''_{\theta'(i)} \overset{*}{\Rightarrow} u'_i$.
Comme pour tout $j \in [q]$, $u_{\theta''(j)} \overset{*}{\Rightarrow} u''_j$, pour tout $i \in [n]$, $u_{\theta''(\theta'(i))} \overset{*}{\Rightarrow} u''_{\theta'(i)} \overset{*}{\Rightarrow} u'_i$. En posant $\theta = \theta'.\theta''$, on a $\forall\, i \in [n]$, $u_{\theta(i)} \overset{*}{\Rightarrow} u'_i$ et $u \overset{*}{\Rightarrow} u''.\theta'' = \tilde{u}''.\theta'.\theta'' = \tilde{u}''.\theta$, ce qui montre que la décomposition $t' = \tilde{u}''.[u'_1, \dots u'_n]$ a bien les propriétés requises.

La réciproque est une simple application de la définition des dérivations.
Si $t = u.[u_1, \dots u_p]$, $t' = u'.[u'_1, \dots u'_n]$ et s'il existe $\theta \in \Theta_p^n$ avec $u \overset{*}{=}> u'.\theta$ et pour tout $k \in [n]$, $u_{\theta(k)} \overset{*}{=}> u'_k$, il résulte directement de la définition de $\overset{*}{=}>$ que $t \overset{*}{=}> t'' = u'.\theta.[u_1, \dots u_p] = u'.[u_{\theta(1)}, \dots u_{\theta(n)}]$.
Posons $u' = \tilde{u}'.\theta'$ avec $\tilde{u}' \in \tilde{T}(\Sigma \cup V)$ et $\theta' \in \Theta_n^q$, on a alors
$t'' = \tilde{u}'.[u_{\theta(\theta'(1))}, \dots u_{\theta(\theta'(q))}]$ et $t'' \overset{*}{=}> t''_1 = \tilde{u}'.[u'_{\theta'(1)}, u_{\theta(\theta'(2))}, \dots u_{\theta(\theta'(q))}]$ car $u_{\theta(\theta'(1))} \overset{*}{=}> u'_{\theta'(1)}$.
De même $t''_1 \overset{*}{=}> t''_2 = \tilde{u}'.[u'_{\theta'(1)}, u'_{\theta'(2)}, u_{\theta(\theta'(3))}, \dots u_{\theta(\theta'(q))}] \dots$
$t''_{q-1} \overset{*}{=}> t''_q = \tilde{u}'.[u'_{\theta'(1)}, \dots u'_{\theta'(q)}] = \tilde{u}'.\theta'.[u'_1, \dots u'_n] = t'$.

□

Toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$ peut être vue comme un système d'équations, chaque règle étant une équation et tout non terminal une inconnue. Une solution est un ensemble de forêts $S_G(X)$ pour tout $X \in V$ qui satisfont ce système. En fait les non-terminaux représentant des ensembles d'arbres, il faudra interpréter les règles comme des équations dans $P_D(T(\Sigma))$ en remplaçant le produit de composition par la composition descendante θ, ou comme des équations dans $P_A(T(\Sigma))$ en remplaçant le produit de composition par la composition ascendante. On notera $S_G(X)$ les solutions dans $P_D(T(\Sigma))$ et $\bar{S}_G(X)$ les solutions dans $P_A(T(\Sigma))$. En définissant S_G comme la substitution descendante qui à tout $\alpha \in \Sigma$ associe $S_G(\alpha) = \alpha$ et à tout $X \in V$, $S_G(X)$, on aura $S_G(X) = \bigcup_{(X\to t)\in R} S_G(t)$. De même en définissant $\bar{S}_G$ comme la substitution ascendante qui à tout $\alpha \in \Sigma$ associe $\bar{S}_G(\alpha) = \alpha$ et à tout $X \in V$, $\bar{S}_G(X)$, on aura $\bar{S}_G(X) = \bigcup_{(X\to t)\in R} \bar{S}_G(t)$. En application du du théorème du point fixe, S_G et $\bar{S}_G$ pourront être construites comme limites de suites $S_G^o, S_G^1, \dots S_G^n \dots$ et $\bar{S}_G^o, \bar{S}_G^1, \dots \bar{S}_G^n$ que nous définissons par :
Pour tout $n \in \mathbb{N}$ et tout $\alpha \in \Sigma$, $S_G^n(\alpha) = \bar{S}_G^n(\alpha) = \{\alpha\}$.
Pour tout $X \in V$ $S_G^o(X) = \bar{S}_G^o(X) = \emptyset$.

Pour tout $n \in N^{+}$ et tout $X \in V$,

$S_G^n(X) = \{t \in T(\Sigma) \mid \exists\ (X \to \tau) \in R,\ t \in S_G^{n-1}(\tau)\}$

$\bar{S}_G^n(X) = \{t \in T(\Sigma) \mid \exists\ (X \to \tau \in R,\ t \in \bar{S}_G^{n-1}(\tau)\}$.

Par récurrence sur n on montre facilement que $S_G^n(t) \subset S_G^{n+1}(t)$ et que $\bar{S}_G^n(t) \subset \bar{S}_G^{n+1}(t)$.

<u>Exemple G6</u>

```
X  →    X      ;     X        →       Y          + a   + a   + a   + a   ;
 o     // \\       / /\ \             |            |     |     |     |
      1 2 3 4    x  x  x  x           X            x     x     x     x
                  1  2  3  4        / /\ \          1     2     3     4
                                   x  x  x  x
                                    1  2  3  4

Y  →    b      +    b
|      / \         / \
x     x   Y       x   x
 1     1  |        1   1
          x
           1
```

on a

```
 1           1                        1
S (X ) = ∅, S (X) = {a  | i  [4]},  S (Y) = {   b    },
 G  o        G       |               G         / \
                     x                        x   x
                      i                        1   1

 2                              2           1
S (X ) = {a | i ∈ [4]},  S (X) = S (X) ∪ {  b     | i, j ∈ [4]},
 G  o     |               G         G        / \
          i                                 a   a
                                            |   |
                                            x   x
                                             i   j

 2
S (Y) = {   b   ,    b      },
 G         / \      / \
          x   x    x   b
           1   1    1 / \
                     x   x
                      1   1

 3          2                                 3          2
S (X ) = S (X ) ∪ {  b    | i, j ∈ [4]},   S (Y) = S (Y) ∪ {   b   ,   b        }
 G  o       G  o    / \                     G          G      / \     / \
                   a   a                                     x   b   x   b
                   |   |                                      1 / \   1 / \
                   i   j                                       x   b   x   b
                                                                1 / \   1 / \
                                                                 x   x x   b
                                                                  1   1 1 / \
                                                                         x   x
                                                                          1   1
```

```
S³_G(X) = S²_G(X) ∪ {   b          | i, j, k ∈ [4]}
                       / \
                     x_i  b
                         / \
                       x_j  x_k
```

<u>Lemme 16</u> Si $\tau \overset{\ell}{\underset{G}{\Rightarrow}}$ avec $t \in T(\Sigma)$ alors il existe $n \leq \ell$ tel que $t \in S_G^n(\tau)$.

<u>Preuve</u>

Par induction sur ℓ en prenant comme hypothèse que le lemme est vrai si $\ell < L$, ce qui est trivialement vrai en prenant $L = 1$ c'est à dire $\ell = 0$ d'où $\tau \in T(\Sigma)$ et $S_G^o(\tau) = \{\tau\}$.

Soit $\tau \Rightarrow \tau' \overset{*}{\Rightarrow} t$ une dérivation de longueur L. Mettons en évidence l'occurrence de non terminal réécrite par la règle $(X \to t')$ utilisée dans la première dérivation :

$\tau = \tilde{u}.[u_1, \ldots u_{k-1}, X.[v_1, \ldots v_{p'}], u_{k+1}, \ldots u_p]$ et

$\tau' = \tilde{u}.[u_1, \ldots u_{k-1}, t'.[v_1, \ldots v_{p'}], u_{k+1}, \ldots u_p]$. D'après le lemme 15 $t = u'.[u'_1, \ldots u'_q]$, il existe une torsion θ et en posant $u_k = t'.[v_1, .. v_{p'}]$, on a $\tilde{u} \overset{*}{\Rightarrow} u'.\theta$ et pour tout $i \in [q]$, $u_{\theta(i)} \overset{*}{\Rightarrow} u'_i$. Toutes ces dérivations sont de longueurs inférieurs à L donc par hypothèse d'induction il existe $n < L$ tel que $\tilde{u}.\theta \in S_G^n(\tilde{u})$ et pour tout $i \in [q]$, $u'_i \in S_G^n(u_{\theta(i)})$. Pour tout $j \in \theta^{-1}(k)$ on a $X.[v_1, \ldots v_{p'}] \Rightarrow t'.[v_1, \ldots v_{p'}] \overset{*}{\Rightarrow} u'_j$, alors, d'après le lemme 15, $u'_j = t''.[v'_1, \ldots v'_{q'}]$ avec $t' \overset{*}{\Rightarrow} t''.\theta''$ et pour tout $i \in [q']$, $v_{\theta''(i)} \overset{*}{\Rightarrow} v'_i$. On a alors $t''.\theta'' \in S_G^n(t')$ et $v'_i \in S_G^n(v_{\theta''(i)})$ et par définition de S_G^{n+1}, puisque $(X \to t') \in R$, $t''.\theta'' \in S_G^{n+1}(X)$.

Comme de plus $S_G^n(t) \subset S_G^{n+1}(t)$ on peut écrire

$$t \in S_G^{n+1}(\tilde{u}) \circledcirc [S_G^{n+1}(u_1), \ldots S_G^{n+1}(u_{k-1}), S_G^{n+1}(X) \circledcirc [S_G^{n+1}(v_1), \ldots \ldots S_G^{n+1}(v_{p'})], S_G^{n+1}(u_{k+1}), \ldots S_G^{n+1}(u_p)]$$

donc que $t \in S_G^{n+1}(\tilde{u}.[u_1, \ldots u_{k-1}, X.[v_1, \ldots v_{p'}], u_{k+1}, .. u_p]) = S_G^{n+1}(\tau)$, car S_G^{n+1} est une substitution descendante. Puisque $n < L$, on a bien $n+1 \leq L$.

□

<u>Lemme 17</u> Si $t \in S^n_G(\tau)$ alors $\tau \overset{*}{\underset{G}{D\Rightarrow}} t$.

<u>Preuve</u>

Par récurrence sur n en prenant comme hypothèse que le lemme est vrai si $n < N$, ce qui est trivialement vrai si $N = 1$. On remarque de plus que si $\forall X \in V$, $t \in S^n_G(X)$ entraine $X \overset{*}{\underset{G}{D\Rightarrow}} t$ alors on a pour tout $\tau \in T(\Sigma \cup V)$, $t \in S^n_G(\tau)$ entraine $\tau \overset{*}{\underset{G}{D\Rightarrow}} t$. Posons $n = N$ et $X \in V$ et $t \in S^n_G(X)$. Par définition de $S^n_G(X)$, il existe $(X \to \tau) \in R$ telle que $t \in S^{n-1}_G(\tau)$. Il suffit alors de montrer par récurrence sur $|\tau|$ que $\tau \overset{*}{D\Rightarrow} t$.

Il est clair que si $|\tau| = 0$ alors $\tau = x_i$, $t = x_i$ et donc $\tau \overset{*}{D\Rightarrow} t$.

Hypothèse de récurrence : $|\tau| < p$ et $t \in S^{n-1}_G(\tau)$ implique $\tau \overset{*}{D\Rightarrow} t$.

Prenons τ avec $|\tau| = p$ et $t \in S^{n-1}_G(\tau)$. Deux cas se présentent

- Ou bien $\tau = \alpha(\tau_1, \dots \tau_q)$ avec $\alpha \in \Sigma_q$ et alors $t = \alpha(\tau'_1, \dots \tau'_q)$, $S^{n-1}_G(\tau) = \alpha[S^{n-1}_G(\tau_1), \dots S^{n-1}_G(\tau_q)]$ et pour tout $i \in [q]$, $\tau'_i \in S^{n-1}_G(\tau_i)$. Comme on a $|\tau_i| < |\tau| = p$, par hypothèse de récurrence sur p, $\tau_i \overset{*}{D\Rightarrow} \tau'_i$ et par définition des dérivations descendantes, $\tau \overset{*}{D\Rightarrow} t$.

- Ou bien $\tau = Y(\tau_1, \dots \tau_q)$ avec $Y \in V_q$. On a alors $t \in S^{n-1}_G(Y) \circledast [S^{n-1}_G(\tau_1), \dots S^{n-1}_G(\tau_q)]$ et il existe $\tilde{u}.\theta \in S^{n-1}_G(Y)$ avec $\tilde{u} \in \tilde{T}(\Sigma)$ et $\theta \in \Theta^r_q$ tels que pour tout $i \in [r]$, il existe $u_i \in S^{n-1}_G(\tau_{\theta(i)})$ et $t = \tilde{u}.[u_1, \dots u_r]$. Alors par hypothèse de récurrence sur n, $Y \overset{*}{D\Rightarrow} \tilde{u}.\theta$. Pour tout $i \in [r]$, $|\tau_{\theta(i)}| < |\tau| = p$ donc par hypothèse de récurrence sur p on a $\tau_{\theta(i)} \overset{*}{D\Rightarrow} u_i$, donc par le lemme 15, $\tau \overset{*}{\Rightarrow} t$ et puisque $\tilde{u} \in \tilde{T}(\Sigma)$, $\tau \overset{*}{D\Rightarrow} t$.

$\square$

<u>Lemme 18</u> Si $t \in T(\Sigma)$ alors $\tau \overset{*}{\underset{G}{\Rightarrow}} t$ équivaut à $t \in S_G(\tau)$.

Preuve

L'implication est évidente dans un sens par définition des dérivations descendantes comme cas particulier des dérivations. Les lemmes 16 et 17 nous donnent l'implication inverse.

□

Théorème 19 Pour toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$ on a pour tout $X \in V$, $F_{IO}(G, X) \subseteq F_{OI}(G, X) = F(G, X)$.

Preuve immédiate par les lemme 17 et 18.

□

Par la suite on utilisera indifféremment F_{OI} ou F.

Si l'on dit que deux sous arbres t_i et t_j de t sont **indépendants** quand aucune occurrence de symbole de t_i n'est ni une occurrence de symbole de t_j ni située au dessus ni au dessous d'une occurrence de symbole de t_j, on obtient le lemme suivant.

Lemme 20 Dans toute dérivation descendante terminale, l'ordre dans lequel sont menées les dérivations portant sur des sous arbres indépendants est sans influence sur le résultat.

Ceci résulte directement de la définition des dérivations descendantes, des lemmes 16 et 17.

Un problème important est de savoir si F(G, X) est vide ou non, ce qui revient en théorie des langages à savoir si le langage engendré depuis un non terminal est vide ou non. En théorie des arbres, il sera utile de se poser un problème plus fin : Etant donné un non terminal X et une partie e des variables de ce non terminal, existe-t-il dans F(G, X) un arbre t avec var(t) = e ? Le lemme et le corollaire suivant répondent à cette question. Une fois ce problème résolu il devient trivial de résoudre les problèmes classiques [Bo] de la vacuité de F(G, X) et de l'utilité d'une variable x_i de X, c'est à dire de l'existence de $t \in F(G, X)$ avec $x_i \in var(t)$. Enfin il peut être utile de savoir si une variable x_i de X est toujours conservée c'est à dire si $\forall\ t \in F(G, X)$, $x_i \in var(t)$, ou de savoir si pour deux variables x_i et x_j de X il existe $t \in F(G, X)$ avec $\{x_i, x_j\} \subset var(t)$. Le lemme et son corollaire permettront alors facilement de répondre à ces questions.

Pour toute forêt $F \subset T(\Sigma)^n_p$ et toute partie $e = \{x_{i_1}, x_{i_2}, ..x_{i_q}\} \subset X_p$ posons $F/e = \{u \in F \mid var(u) = e\}$. De cette façon on pourra poser $S^n_G(X) = \bigcup_{e \subset \{x_1,..x_{d(X)}\}} (S^n_G(X)/e)$ pour étudier les arbres de F(G, X) qui conservent ou non certaines variables de X.

On posera $D(V) = \sup_{X \in V}(d(X))$ et pour tous t et $t' \in T(\Sigma \cup V)^1_n$, $t \leq t'$ si et seulement si pour tout $x_i \in X_n$, le nombre d'occurrences de x_i dans t, $\nu_{x_i}(t)$ est inférieur ou égal à $\nu_{x_i}(t')$ ou si $\nu_{x_i}(t') \geq D(V)$. C'est à dire que $t \leq t'$ si et seulement si $\forall\ x_i$, $\min(D(V), \nu_{x_i}(t)) \leq \nu_{x_i}(t')$.

Cette relation de préordre sur les arbres de $T(\Sigma)^1_n$ induit la relation de préordre suivante sur les forêts de $T(\Sigma)^1_n$: si $F, F' \subset T(\Sigma)^1_n$, on aura $F \leq F'$ si $\forall\ t \in F$, $\exists\ t' \in F'$ avec $t \leq t'$. Si on a $t \leq t'$ et $t' \leq t$ on notera $t \simeq t'$, par contre si $t \leq t'$ et non $t' \leq t$ alors on pourra noter $t < t'$, on étendra ces autres notations aux forêts. Remarquons que $t \simeq t'$ implique que les variables apparaissant effectivement en feuillage des deux arbres sont les mêmes.

<u>Lemme 21</u> Pour toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$ on peut trouver un entier k tel que pour $X \in V$ et tout $e \subset var(X)$ on ait $F(G, X)/e \simeq S_G^k(X)/e$.

<u>Preuve</u>

Par construction, pour tous $n \in \mathbb{N}$, $X \in V$, $e \subset var(X)$ on a $S_G^{n-1}(X)/e \subset S_G^n(X)/e$ donc $S_G^{n-1}(X)/e \leq S_G^n(X)/e$. Il est clair que cette suite croissante est bornée par tout ensemble contenant un arbre avec $D(V)$ occurrences de chacune des variables de e. Définissons pour tout n, $S_n = \{S_G^n(X)/e \mid X \in V, e \subset var(X)\}$ et notons $S_i \leq S_j$ si et seulement si $\forall x \in V$, $\forall e \subset var(X)$, $S_G^i(X)/e \leq S_G^j(X)/e$. La suite S_n est composée des suites $S_G^n(X)/e$, leur nombre N est une constante de G. Dans chaque suite $S_G^n(X)/e$ on peut trouver au plus $D(V)^{Card(e)}$, donc à fonction au plus $D(V)^{D(V)}$, éléments distincts vis à vis de $\simeq$. Donc S_n contient au plus $k = N * D(V)^{D(V)}$ éléments distincts. Il est clair que pour tous n, $S_n \leq S_{n+1}$. Il suffit alors de montrer que si $S_n \simeq S_{n+1}$ alors $\forall p > n$ on a $S_n \simeq S_p$, pour avoir le résultat.

Il suffit pour cela de montrer que si $S_{n-1} \simeq S_n$ c'est à dire si pour tous $X \in V$, $e \subset var(X)$, $t_1 \in S_G^n(X)/e$ il existe $t_2 \in S_G^{n-1}(X)/e$ avec $t_1 \leq t_2$ alors pour tout $t \in S_G^{n+1}(X)/e$ il existe $t' \in S_G^n(X)/e$ avec $t \leq t'$.

Si $t \in S_G^{n+1}(X)/e$ alors $\exists\ (X \to \tau) \in R$ telle que $t \in S_G^n(\tau)$ par définition de S_G^{n+1}. Nous montrons alors par induction sur $|\tau|$ qu'il existe $t' \in S_G^n(X)/e$ avec $t \leq t'$. Si $|\tau| = 0$ alors $\tau = x_i$, $t = x_i \in S_G^o(x_i) \subset S_G^n(x_i)$ et $x_i \leq x_i$.

Hypothèse d'induction : si $|\tau| < p$ et $t \in S_G^{n+1}(\tau)$ alors $\exists\ t' \in S_G^n(\tau)$ avec $t \leq t'$.

Prenons τ de profondeur $p > 0$ on a

- Ou bien $\tau = \alpha(\tau_1, \dots \tau_q)$ avec $\alpha \in \Sigma_q$

alors $S_G^{n+1}(\tau) = \{\alpha\} @ [S_G^{n+1}(\tau_1), \dots S_G^{n+1}(\tau_q)]$

et $S_G^n(\tau) = \{\alpha\} @ [S_G^n(\tau_1), \dots S_G^n(\tau_q)]$.

$t \in S_G^{n+1}(\tau)$ entraine $t = \alpha(t_1, \dots t_q)$ et pour tout $i \in [q]$

on a $t_i \in S_G^{n+1}(\tau_i)$, $|\tau_i| < p$ donc par hypothèse d'induction $\exists\ t'_i \in S_G^n(\tau_i)$ avec $t_i \leq t'_i$ donc en posant $t' = \alpha[t'_1, \ldots t'_q]$ on a $t' \in S_G^n(\tau)$ et $t \leq t'$.

- Ou bien $\tau = Y(\tau_1, \ldots \tau_q)$ avec $Y \in V_q$, alors $t \in S_G^{n+1}(\tau) = S_G^{n+1}(Y) \circledast [S_G^{n+1}(\tau_1), \ldots S_G^{n+1}(\tau_q)]$. On a donc $t = \tilde{t}.[t_1, \ldots t_r]$, $\tilde{t} \in \tilde{T}(\Sigma)_r^1$ et $\exists\ \theta \in \Theta_q^r$ telle que $\tilde{t}.\theta \in S_G^{n+1}(Y)$ et pour tout $i \in [r]$, $t_i \in S_G^{n+1}(\tau_{\theta(i)})$. Comme $|\tau_{\theta(i)}| < p$, par hypothèse d'induction pour tout $i \in [r]$ il existe $t'_i \in S_G^n(\tau_{\theta(i)})$ avec $t_i \leq t'_i$. D'autre part $\tilde{t}.\theta \in S_G^{n+1}(Y)/var(\theta)$ donc il existe une règle $(Y \rightarrow \tau') \in R$ telle que $\tilde{t}.\theta \in S_G^n(\tau')$ et puisque $S_{n-1} \simeq S_n$ il existe $\tilde{t}'.\theta' \in S_G^{n-1}(\tau')$ avec $\tilde{t}.\theta \leq \tilde{t}'.\theta'$. On a de plus par déﬁniton de $S_G^n(Y)$, $\tilde{t}'.\theta' \in S_G^n(Y)$. Il est alors aisé de construire $t' = \tilde{t}.[t'_1, \ldots t'_{r'}] \in S_G^n(\tau)$ tel qu'à tout sous arbre t_i de t on puisse faire correspondre un sous arbre $t'_{i'}$ de t' avec $\theta(i) = \theta'(i')$ et $t_i \leq t'_{i'}$. En effet on peut diviser les variables de θ en deux parties $e_1(\theta)$ l'ensemble des variables x_i de θ qui se rencontrent au plus $D(V)$ fois dans θ et $e_2(\theta)$ celles qui re rencontrent plus de $D(V)$ fois. Toute variable de $e_1(\theta)$ se trouvera un nombre de fois au moins égal dans θ' donc tous les arbres qui leur correspondent dans t pourront se retrouver dans t'. Toute variable de $e_2(\theta)$ est dans $e_2(\theta')$, tout arbre de $S_G^n(X)$ a au plus $D(V)$ variables, donc à fortiori tout τ_i a au plus $D(V)$ variables, donc au plus $D(V)$ arbres de $S_G^n(\tau_i)$ sont nécessaires pour faire apparaître ces $D(V)$ variables, chacun avec au moins une variable donc si x_i apparait plus de $D(V)$ fois dans θ, il suffit qu'il apparaisse $D(V)$ fois dans θ'. Ainsi on aura $t \leq t'$.

$\square$

L'exemple G6 met en évidence l'utilité de $D(V)$, en effet il est clair que $F(G, X)/e$ est non vide quelque soit $e \subset var(X)$ et il faudra aller jusqu'à S_G^4 pour avoir $S_G^4(X)/e \neq \emptyset$ pour tout $e \subset var(X)$, car $S_G^3(Y)$ est le premier à contenir un arbre avec $D(V) = 4$ occurrences de x_1.

<u>Corollaire 22</u> Pour tout $X \in V$ et $e \subset var(X)$ on peut décider si $F(G, X)/e$ est vide.

<u>Preuve</u>

Il suffit de savoir si $S_G^k(X)/e$ est vide.

□

On a $F(G, X) = \bigcup_{e \subset var(X)} F(G, X)/e$ ce qui permet de décider de la vacuité de $F(G, X)$. De même $\bigcup_{\substack{e \subset var(X) \\ x_i \in e}} F(G, X)/e$ permet de décider s'il existe $t \in F(G, X)$ avec $x_i \in var(t)$.

Nous allons maintenant faire une étude similaire sur la forêt ascendante $F_{IO}(G)$ en utilisant les substitutions $\bar{S}_G^n$.

<u>Lemme 23</u> Si $\tau \overset{\ell}{\underset{G}{A\Rightarrow}} t$ avec $t \in T(\Sigma)$ alors il existe $n \leq \ell$ tel que $t \in \bar{S}_G^n(\tau)$.

<u>Preuve</u>

Par récurrence sur ℓ, en prenant comme hypothèse que le lemme est vrai si $\ell < L$ et en remarquant que si $L = 1$ alors $\ell = 0$ d'où $\tau \in T(\Sigma)$ et $\bar{S}_G^o(\tau) = \{\tau\}$.

Prenons une dérivation $\tau \; A\Rightarrow \; \tau' \; \overset{*}{A\Rightarrow} \; t$ de longueur $\ell = L$ et posons $r = (X \to t')$ la règle utilisée en première dérivation. $\tau = \tilde{u}.[u_1, \ldots u_{k-1}, X.\vec{v}, u_{k+1}, \ldots u_p]$ avec $\vec{v} \in T(\Sigma), \tau' = \tilde{u}.[u_1, \ldots u_{k-1}, t'.\vec{v}, u_{k+1}, \ldots u_p]$ par définition des dérivations ascendantes. Par hypothèse de récurrence on a

$t \in \overline{S_G^n}(\tilde{u}) \; \theta \; [\overline{S_G^n}(u_1), \ldots \overline{S_G^n}(u_{k-1}), \overline{S_G^n}(t'.v), \overline{S_G^n}(u_{k+1}), \ldots \overline{S_G^n}(u_p)]$.

En posant $t' = \tilde{t}'.\theta$ avec $\tilde{t}' \in \tilde{T}(\Sigma \cup V)$ et $\theta \in \Theta$ on a $\overline{S_G^n}(\tilde{t}'.\theta.\vec{v}) = \overline{S_G^n}(\tilde{t}') \; \theta \; \overline{S_G^n}(\theta.\vec{v})$ d'après le lemme 13, mais puisque $\vec{v} \in T(\Sigma)$ et que $\overline{S_G^n}$ est l'identité sur Σ, on a $\overline{S_G^n}(\theta.\vec{v}) = \{\theta.\vec{v}\}$ d'où $\overline{S_G^n}(t'.\vec{v}) = \overline{S_G^n}(t') \; \theta \; \overline{S_G^n}(\vec{v})$.

Par définition de $\bar{S}_G^{n+1}$ on a $\bar{S}_G^{n}(t') \subset \bar{S}_G^{n+1}(X)$ donc $\bar{S}_G^{n}(t'.\vec{v}) \subset \overline{S_G^{n+1}}(X.\vec{v})$ et comme pour tout $\rho \in T(\Sigma \cup V)$, $\bar{S}_G^{n}(\rho) \subset \bar{S}_G^{n+1}(\rho)$ on obtient $t \in \overline{S_G^{n+1}}(\tau)$.

Par hypothèse de récurrence $n \leq \ell-1$ d'où $n+1 \leq \ell$.

□

<u>Lemme 24</u> Si $t \in \overline{S_G^{n}}(\tau)$ alors $\tau \overset{*}{\underset{G}{A\!\!=\!\!>}} t$.

<u>Preuve</u>

Il est clair que c'est vrai pour $n = 0$ et $n = 1$ par définition de $\overline{S_G^{n}}(X)$. On remarque que si $\forall X \in V$, $t \in \overline{S_G^{n}}(X)$ entraine $X \overset{*}{\underset{G}{A\!\!=\!\!>}} t$ alors $\forall \tau \in T(\Sigma \cup V)$, $t \in \overline{S_G^{n}}(\tau)$ entraine $\tau \overset{*}{A\!\!=\!\!>} t$. Prenons comme hypothèse de récurrence que le lemme est vrai si $n < N$ et prenons $X \in V$, $t \in S_G^{n}(X)$ avec $n = N > 0$.

Par définition de $\overline{S_G^{n}}(X)$, il existe $r = (X \to \tau) \in R$ telle que $t \in \overline{S_G^{n-1}}(\tau)$.

Il suffit alors de montrer par récurrence sur $|\tau|$ que $\tau \overset{*}{A\!\!=\!\!>} t$, ce qui est évident si $|\tau| = 0$.

Posons comme hypothèse de récurrence sur $|\tau|$ que si $|\tau| < p$ alors $\tau \overset{*}{A\!\!=\!\!>} t$ et prenons τ de profondeur $p > 0$. Deux cas se présentent.

- Ou bien $\tau = \alpha(\tau_1, \ldots \tau_q)$ avec $\alpha \in \Sigma_q$ dans $t = \alpha(\tau'_1, \ldots \tau'_q)$
  $\overline{S_G^{n-1}}(\tau) = \{\alpha\} \circledast [\overline{S_G^{n-1}}(\tau_1), \ldots \overline{S_G^{n-1}}(\tau_q)]$ donc pour tout $i \in [q]$ on a $|\tau_i| < |\tau| = p$ et par hypothèse de récurrence sur $|\tau|$ et par définition des dérivations ascendantes $\tau \overset{*}{A\!\!=\!\!>} \tau'_i$ et $\tau \overset{*}{A\!\!=\!\!>} t$.

- Ou bien $\tau = Y(\tau_1, \ldots \tau_q)$ avec $Y \in V_q$. Dans ce cas $t = t'.[\tau'_1, \ldots \tau'_q]$ avec $t' \in \overline{S_G^{n-1}}(Y)$ donc par hypothèse de récurrence sur n, $Y \overset{*}{A\!\!=\!\!>} t'$, et pour tout $i \in [q]$, $\tau'_i \in \overline{S_G^{n-1}}(\tau_i)$ donc $\tau_i \overset{*}{A\!\!=\!\!>} \tau'_i$ comme dans le cas précédent et $\tau \overset{*}{A\!\!=\!\!>} t$.

□

<u>Corollaire 25</u> Si $t \in T(\Sigma)$, $t \in \bar{S}_G(\tau)$ équivaut à $\tau \underset{G}{\overset{*}{A\Rightarrow}} t$.

<u>Lemme 26</u> Pour toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$ on peut trouver une constante k' telle que pour tout $X \in V$ et tout $e \subset var(X)$ on ait $F_{IO}(G, X)/e = \emptyset \iff \bar{S}_G^{k'}(X)/e = \emptyset$

<u>Preuve</u>

Posons $\bar{S}_n = \{\bar{S}_G^n(X)/e \mid X \in V,\ e \subset var(X)\}$ et notons pour tout $n, \bar{S}_n < \bar{S}_{n+1}$ s'il existe $X \in V$ et $e \in var(X)$ tels que $\bar{S}_G^n(X)/e = \emptyset$ et $\bar{S}_G^{n+1}(X)/e \neq \emptyset$, étant entendu que l'on a toujours $\bar{S}_G^n(X)/e \subset \bar{S}_G^{n+1}(X)/e$ et qu'en ce sens $\bar{S}_n \leq \bar{S}_{n+1}$. Nous noterons alors $\bar{S}_n \simeq \bar{S}_{n+1}$ si l'on n'a pas $\bar{S}_n < \bar{S}_{n+1}$. Nous allons montrer que si $\bar{S}_{n-1} \simeq \bar{S}_n$ alors $\bar{S}_n \simeq \bar{S}_{n+1}$ et donc que pour tout $p > n$, $\bar{S}_n \simeq \bar{S}_p$. Comme de plus la suite contient un nombre borné d'éléments distincts (car V est borné et $\forall X \in V$, $d(X)$ est borné), on en tire l'existence de k' tel que $\bar{S}^{k'} \simeq \bar{S}^n$ pour tout $n \geq k'$. D'où pour tout $n \geq k'$, $\bar{S}_G^n(X)/e = \emptyset \iff \bar{S}_G^{k'}(X)/e = \emptyset$ et le résultat.

Posons $\bar{S}_{n-1} \simeq \bar{S}_n$. Si $t \in \bar{S}_G^{n+1}(X)$ alors $\exists\ (X \to \tau) \in R$ avec $t \in \bar{S}_G^n(\tau)$. Nous allons montrer par récurrence sur $|\tau|$ qu'il existe $t' \in \bar{S}_G^{n-1}(\tau)$ avec $var(t') = var(t)$. Alors, puisque $\bar{S}_G^{n-1}(\tau) \subset \bar{S}_G^n(\tau)$ on pourra en conclure l'équivalence $\bar{S}_G^n(X)/var(t) = \emptyset \iff \bar{S}_G^{n+1}(X)/var(t) = \emptyset$.

Hypothèse de récurrence : si $|\tau| < p$ alors il existe $t' \in \bar{S}_G^{n-1}(\tau)$ avec $var(t) = var(t')$.

Ce qui est vrai si $|\tau| = 0$ car alors $\tau = x_i$, $t = x_i$ et $x_i \in \bar{S}_G^q(x_i)$ quelque soit $q \in \mathbb{N}$.

Prenons τ de profondeur $p > 0$, alors $\tau = \alpha[\tau_1, \ldots \tau_q]$ avec $\alpha \in \Sigma_q \cup V_q$. Comme $t \in \bar{S}_G^n(\tau)$ on a $t = u.[t_1, \ldots t_q]$ avec $u \in \bar{S}_G^n(\alpha)$ et pour tout $i \in [q]$, $t_i \in \bar{S}_G^n(\tau_i)$. On a alors $|\tau_i| < |\tau| = p$ donc par hypothèse de récurrence il existe $t'_i \in \bar{S}_G^{n-1}(\tau_i)$ avec $var(t'_i) = var(t_i)$. Si $\alpha \in \Sigma$ on a alors $\bar{S}_G^n(\alpha) = \{\alpha\} = \bar{S}_G^{n-1}(\alpha)$ donc $u = \alpha$ et $t' = \alpha.[t'_, \ldots t'_q]$ a les mêmes variables que t. Si $\alpha \in V$ alors $\exists\, u' \in \bar{S}_G^{n-1}(\alpha)$ avec $var(u) = var(u')$ car $\bar{S}_{n-1} \simeq \bar{S}_n$ donc en posant $t' = u'.[t'_1, \ldots t'_q]$ où les t'_i sont ceux définis plus haut, on a bien $var(t) = var(t')$.

□

<u>Corollaire 27</u> Pour tout $X \in V$ et pour $e \subset var(X)$ on peut décider si $F_{IO}(G, X)/e$ est vide.

<u>Lemme 28</u> On peut toujours supposer que les règles linéaires et terminales seront appliquées après toutes les autres dans une dérivation.

<u>Preuve</u>

Donnons nous $G = \langle V, \Sigma, X_o, R\rangle$.

Il suffit de montrer que pour toute dérivation $t_1 \underset{r}{\Rightarrow} t_2 \underset{r'}{\Rightarrow} t_3$ avec r linéaire et terminale et r' quelconque, il existe une dérivation $t_1 \underset{r'}{\Rightarrow} t'_2 \overset{*}{\underset{r}{\Rightarrow}} t_3$.

Posons $r = (X \to \tilde{t}.\theta)$ avec $\tilde{t} \in \mathcal{T}(\Sigma)^1_n$ et θ une torsion linéaire de Θ^n_m, en posant $m = d(X)$.

$t_1 = \tilde{u}.[u_1, \ldots u_p]$ avec $\tilde{u}$ initial et il existe $k \in [p]$ tel que $u_k = X(v_1, \ldots v_m)$. On a alors $t_2 = \tilde{u}.[u'_1, \ldots u'_p]$ avec $u'_k = \tilde{t}.\theta.[v_1, \ldots v_m] = \tilde{t}.[v_{\theta(1)}, \ldots v_{\theta(n)}] = \tilde{t}.[\bar{v}_1, \ldots \bar{v}_n]$ et pour tout $j \neq k$, $u_j = u'_j$. La règle r' ne peut s'appliquer à $\tilde{t}$ qui est terminal, trois cas se présentent alors :

1) r' s'applique à $\tilde{u}$ donc $\tilde{u} \Rightarrow \tilde{u}'.\theta'$ avec $\tilde{u}' \in T(\Sigma \cup V)^1_{p'}$. D'où $t_3 = \tilde{u}'.\theta'.[u'_1, \ldots u'_p] = \tilde{u}'.[u'_{\theta'(1)}, \ldots u'_{\theta'(p')}]$. D'autre part $t_1 \underset{r'}{\Rightarrow} \tilde{u}'.[u_{\theta'(1)}, \ldots u_{\theta'(p')}] = t'_2$. Posons $I = \theta'^{-1}(k) = \{i_1, \ldots i_q\}$ l'ensemble des indices i tels que $\theta'(i) = k$. Alors $\forall\, j \in I$, $u_{\theta'(j)} \underset{r}{\Rightarrow} u'_{\theta'(j)}$ et $\forall\, j \notin I$, $u_{\theta'(j)} = u'_{\theta'(j)}$ donc $t'_2 \overset{q}{\underset{r}{\Rightarrow}} t_3$.

2) r' s'applique à u_j avec $j \neq k$. Le lemme 20 montre que dans ce cas $t_1 \underset{r'}{\Rightarrow} t'_2 \underset{r}{\Rightarrow} t_3$.

3) r' s'applique à $\bar{v}_j$. Posons $\bar{v}_j \underset{r'}{\Rightarrow} \bar{v}'_j$. Alors $t_3 = \tilde{u}.[u''_1, \ldots u''_p]$ avec $u''_k = \tilde{t}.[\bar{v}_1, \ldots \bar{v}_{j-1}, \bar{v}'_j, \bar{v}_{j+1}, \ldots \bar{v}_n]$ et pour tout $i \neq k$, $u''_i = u'_i = u_i$. Posons $\ell = \theta(j)$, alors $v_\ell = \bar{v}_j$ et $u_k \underset{r'}{\Rightarrow} X(v_1, \ldots v_{\ell-1}, \bar{v}'_j, v_{\ell+1}, \ldots v_m) \underset{r}{\Rightarrow} \tilde{t}.\theta.[v_1, \ldots v_{\ell-1}, \bar{v}'_j, v_{\ell+1}, \ldots v_m] = \tilde{t}.[v'_1,..v'_n]=\bar{u}_k$ avec $v'_h = \bar{v}'_j$ si $\theta(h) = \ell$ et $v'_h = v_h$ si $\theta(h) \neq \ell$. Comme la règle r est linéaire, θ est injective et j est le seul élément de $[n]$ tel que $\theta(j) = \ell$. Ceci entraine que $v'_h = v_{\theta(h)} = \bar{v}_h$ si $h \neq j$ et $v'_j = \bar{v}'_j$ donc $\bar{u}_k = u''_k$ et $t_1 \underset{r'}{\Rightarrow} t'_2 \underset{r}{\Rightarrow} t_3$.

$\square$

II - NOTION DE GRAMMAIRE REDUITE ET REDUCTIONS

1) INTRODUCTION

L'une des premières préoccupations des chercheurs en théorie des langages a été de chercher à simplifier les grammaires. Une première simplification consiste à éviter qu'elles ne soient encombrées d'éléments non nécessaires comme les non-terminaux inaccessibles depuis l'axiome, et les non terminaux engendrant un langage vide.

En théorie des arbres il est aussi simple de retirer les non terminaux inaccessibles depuis l'axiome, par contre pour retirer les non terminaux engendrant une forêt vide, on ne pourra pas se contenter de retirer les règles dans lesquelles ils apparaissent, comme c'est le cas en théorie des langages. La raison en est que, au cours des dérivations, un sous-arbre, et en particulier un sous arbre contenant des non terminaux, peut disparaître par application de règles non complètes. Ainsi si un arbre t contient un sous arbre τ engendrant une forêt vide, c'est à dire que $t = \tilde{u}.[\vec{v}_1, \tau, \vec{v}_2].\vec{w}$ avec $\tilde{u} \in \tilde{\mathcal{T}}(\Sigma \cup V)$ et $F(G, \tau) = \emptyset$, et si G n'est pas complète on n'a pas nécessairement $F(G, t) = \emptyset$. L'exemple suivant illustrera ce point.

Exemple G7 $\Sigma_o = \{0, 1\}$, $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, $V_o = \{X_o, Z\}$, $V_2 = \{X\}$

```
Xo →  X   +  X             X    →   b    + a
     / \    / \           / \      / \     |
    o   Z  1   1        x1   x2  x1   x2   x1
```

```
On a F(G7, Z) = Ø par contre F(G7,   X  ) = a
                                    / \     |
                                   o   Z    o
```

```
d'où F(G7) = {a, a,   b  }.
              |  |   / \
              o  1  1   1
```

Cependant $F_{IO}(G7, Z) = \emptyset$ entraine

```
F_IO(G7,  X  ) = F_IO(G7,   X    ) ⊕ F_IO(G7, Z) = ∅
         / \               / \
        o   Z             o   x1

d'où F_IO(G7) = F_IO(G7,   X  ) = {a,    b  }.
                          / \      |    / \
                         1   1     1   1   1
```

On dira qu'un symbole X_n est **accessible depuis** $X_1 \in V$ dans $G = \langle V, \Sigma, X_o, R\rangle$ s'il existe $X_2, X_3, \dots X_{n-1}$ dans V et pour tout $i \in \{2, \dots n\}$ il existe une règle $X_{i-1} \to t_{i-1}$ telle qu'il y ait une occurrence de symbole X_i dans t_{i-1}. On dira simplement que X est **accessible** s'il est accessible depuis X_o. Il est clair que si G est **sans non terminal inacessible,** pour tout $X \in V$ il existe une dérivation $X \overset{*}{\Rightarrow} t$ avec une occurrence de X dans t.

On dira que la grammaire G est **totale** si pour tout $X \in V$, $F(G, X) \neq \emptyset$ et qu'elle est **IO-totale** si pour tout $X \in V$, $F_{IO}(G, X) \neq \emptyset$. Comme $F_{IO}(G, X) \subset F(G, X)$ si une grammaire est IO-totale, elle est totale mais la réciproque n'est pas vraie.

```
Exemple G8 : Xo → Z +   X  ,    X    →    b    + a , Z →   X
                       / \     / \       / \     |        / \
                      1   1  x1   x2   x1   x2   x1       o   Z
```

on a $F_{IO}(G8, Z) = \emptyset$ car la seule dérivation ascendante possible pour Z est $Z \Rightarrow X(0, x_1).Z \Rightarrow (X(0, x_1))^2.Z \overset{*}{\Rightarrow} (X(0, x_1))^n.Z$ qui ne produit aucun arbre terminal.

```
Par contre F_OI(G8, Z) contient a entre autres.
                                |
                                o
```

Nous verrons qu'il est possible pour toute grammaire G de trouver une grammaire G' totale telle que F(G) = F(G') et une grammaire IO-totale G" telle que $F_{IO}(G) = F_{IO}(G'')$. Par contre en général on ne peut trouver G'' IO-totale telle qu'on ait à la fois $F_{IO}(G) = F_{IO}(G'')$ et $F(G) = F(G'')$, en particulier à $G7$ on ne pourra faire correspondre une telle grammaire.

Un autre type d'élément non nécessaire, propre aux grammaires d'arbres, est celui des variables inutiles dans un non terminal. On dira que x_i est une **variable utile** dans le non terminal X si $i \in [d(X)]$ et $\exists\ t \in F(G, X)$ avec $x_i \in var(t)$, le non-terminal X sera alors d'**arité utile** si toutes ses variables sont utiles.

<u>Exemple G9</u> :
```
X  →    X    ,    X    → a  , Y →   X
 o     / \       / \     |         / \
      o   Y    x1   x2   x1       o   Xo
```

la variable x_2 est inutile dans le non terminal X, il s'ensuit que toute la forêt F(G, Y) qui peut apparaître en cours de dérivation (non descendante) introduit une complication inutile.

On dira que x_i est une variable **IO-utile** dans le non terminal X si $i \in [d(X)]$ et $\exists\ t \in F_{IO}(G, X)$ avec $x_i \in var(t)$, le non-terminal X sera alors d'**arité Io-utile** si toutes ses variables sont IO-utiles.

Dans les grammaires interprétées comme schémas de programmes, un non terminal inaccessible correspond à une procédure qui ne sera jamais appelée, un non terminal engendrant une forêt vide correspond à une procédure qui ne donne jamais de résultat et une variable inutile à un paramètre qui n'est jamais utilisé dans une procédure. Supprimer ces éléments inutiles revient à simplifier le programme. Ces réductions ont été étudiées dans le cadre particulier des grammaires déterministes par Irène Guessarian [GU].

La réduction de la totalité a été étudiée dans le cas OI [AD, dupl] [A-te], [Bo] et [Co-2] et dans le cas IO [Bo]. La réduction des arités inutiles a été étudiée par Boudol [Bo]. L'étude de ces réductions est reprise ici avec la contrainte supplémentaire de conserver certaines propriétés intéressantes des grammaires.

Nous dirons qu'une grammaire est **réduite** si elle est totale, utile et sans non terminal inaccessible, de même nous dirons qu'elle est **IO-réduite** si elle est IO-totale, IO-utile et sans non-terminal inaccessible. Il est clair qu'une grammaire IO-réduite est aussi réduite.

2) REDUCTION DE LA NON-ACCESSIBILITE EN IO ET OI

Lemme II,1 Si $X \in V$ est inacessible dans $G = <V, \Sigma, X_o, R>$, en posant $\hat{X} = \{Y \in V \mid X$ est accessible depuis $Y\}$ et $G' = <V \setminus \hat{X}, \Sigma, X_o, R'>$ avec $R' = \{(X \to t) \in R \mid X \in V \setminus \hat{X}\}$ on a $F(G) = F(G')$ et $F_{IO}(G) = F_{IO}(G')$.

Preuve

V et R étant finis on peut construire pour tout $X \in V$ l'ensemble $\bar{X}$ des non terminaux accessibles depuis X, il suffit de remarquer que si Y est accessible depuis X, il existe une chaine $X = X_1, X_2, \ldots X_n = Y$ avec $n \leq \text{Card}(V)$ avec pour tout $i \in \{2, \ldots n\}$, $\exists(X_{i-1} \to t_{i-1}) \in R$ avec une occurrence de X_i dans t_{i-1}.
Il est alors immédiat de construire $\hat{X}$.
Il est clair que si un élément de $\hat{X}$ apparait dans une règle $(Y \to t) \in R$ c'est que $Y \in \hat{X}$ et qu'aucune règle où apparait un élément de $\hat{X}$ ne peut être utilisée dans une dérivation depuis X_o dans G.
Donc en posant $G'' = <V, \Sigma, X_o, R'>$ on a $F(G) = F(G'')$.
De plus comme aucun élément de $\hat{X}$ n'apparait dans R' on peut écrire $G' = <V \setminus \hat{X}, \Sigma, X_o, R'>$ et on a $F(G') = F(G'')$.

□

Corollaire II,2 Toute grammaire peut être supposée sans non terminal inaccessible.

Preuve

Il suffit d'itérer le procédé pour chaque non terminal inaccessible.

□

3) REDUCTION DE LA NON TOTALITE EN OI

La construction qui suit est due à Boudol [Bö] elle permet d'obtenir une grammaire totale engendrant la même forêt que toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$. L'idée de base de la construction vient de la remarque suivante :

Si $F(G, \tilde{u}.[t_1, \dots t_n]) \neq \emptyset$ et qu'il existe $i \in [n]$ avec $F(G, t_i) = \emptyset$, c'est qu'il existe $t \in F(G, \tilde{u})$ avec $x_i \notin var(t)$, tout élément engendrant une forêt vide finit par être abandonné au cours des dérivations, par application d'une règle non complète. On va donc essayer de ne pas le produire. Prenons une règle $r = \left(\begin{array}{c} X \\ | \\ x_1 \end{array} \to \begin{array}{c} Y \\ / \ \backslash \\ x_1 \quad Z \end{array}\right)$ où $X, Y, Z \in V$, $F(G, Z) = \emptyset$ et $F(G, Y(x_1, Z)) \neq \emptyset$.

C'est qu'il existe $t \in F(G, Y(x_1, x_2))$ avec $x_2 \notin var(t)$. Si l'on sait construire un nouveau non terminal Y' d'arité 1 et des règles associées telles Y' engendre exactement $\{t \mid t \in F(G, Y) \text{ et } x_2 \notin t\}$ alors la règle r pourra être remplacée par $\left(\begin{array}{c} X \\ | \\ x_1 \end{array} \to \begin{array}{c} Y' \\ | \\ x_1 \end{array}\right)$. On pourra étendre ce procédé de façon que tous les non terminaux engendrant une forêt vide deviennent inaccessibles, il suffira alors de les retirer.

Construction II,1

A toute partie finie $\alpha \subset \mathbb{N}^+$, contenant $d(\alpha)$ éléments notés par ordre numérique croissant $i_1, i_2, \dots i_{d(\alpha)}$, on associe d'une part l'application α injective, croissante de $[d(\alpha)]$ dans $[i_{d(\alpha)}]$, définie pour tout $j \in [d(\alpha)]$ par $\alpha(j) = i_j$, et d'autre part l'ensemble de variables $x(\alpha) = \{x_{i_1}, \dots x_{i_{d(\alpha)}}\} = \{x_{\alpha(j)} \mid j \in [d(\alpha)]\}$.

Etant données deux parties $\alpha \subset \beta \subset \mathbb{N}^+$ notons θ^α_β la torsion linéaire ordonnée de $\Theta^{d(\alpha)}_{d(\beta)}$ telle que pour tout $j \in [d(\alpha)]$, $\theta^\alpha_\beta(j) = k \in [d(\beta)]$ tel que $\beta(k) = \alpha(j)$. Il est alors immédiat que si $\alpha \subset \beta \subset \gamma \subset \mathbb{N}^+$ on a $\theta^\alpha_\gamma = \theta^\alpha_\beta . \theta^\beta_\gamma$

car pour tout $j \in [d(\alpha)]$, $\theta^{\alpha}_{\gamma}(j) = \theta^{\beta}_{\gamma}(\theta^{\alpha}_{\beta}(j))$.

Un cas particulier important est celui où $\alpha \subset \beta \subset \mathbb{N}^{+}$ avec $\beta = [n]$, $n \geq \alpha(d(\alpha))$.

Alors en posant $\alpha = \{i_1, \dots i_{d(\alpha)}\}$ on a $\theta^{\alpha}_{[n]} = \langle n \; ; x_{i_1}, \dots x_{i_{d(\alpha)}} \rangle$.

Soit $\theta = \langle p \; ; x_{i_1}, .. x_{i_n} \rangle$ une torsion quelconque (pas nécessairement linéaire ou croissante de θ^{n}_{p} avec $var(\theta) \subset x(\alpha)$ et $\alpha \subset [p]$. Posons $R_{\alpha}(\theta) = \langle d(\alpha) \; ; x_{j_1}, \dots x_{j_n} \rangle \in \Theta^{n}_{d(\alpha)}$ avec pour tout $k \in [n]$, $\theta^{\alpha}_{[p]}(j_k) = i_k$, c'est à dire que $R_{\alpha}(\theta)$ est la torsion obtenue depuis θ en renumérotant les variables par le rang de leur indice dans α. On a $R_{\alpha}(\theta).\theta^{\alpha}_{[p]} = \theta$ et si $\alpha \subset \beta \subset [p]$ on a $R_{\beta}(\theta) = R_{\alpha}(\theta).\theta^{\alpha}_{\beta}$.

Pour tout arbre $t = \tilde{t}.\theta$ on notera $R_{\alpha}(t) = \tilde{t}.R_{\alpha}(\theta)$.

Pour tout alphabet gradué Δ, toute forêt $F \subset T(\Delta)^{1}_{n}$ et tout $\alpha \subset [n]$ on pose $P_{\alpha}(F) = \{t \in F \mid var(t) \subset x(\alpha)\}$.

Donnons nous une grammaire algébrique $G = \langle V, \Sigma, X_o, R \rangle$.

Pour tout $X \in V$ posons

$\hat{X} = \{X_{\alpha} \mid \alpha \subset [d(X)]$ et $d(X_{\alpha}) = d(\alpha)\}$ et $\hat{V} = \bigcup_{X \in V} \hat{X}$. Toutefois nous noterons $\bar{X}_o$ plutôt que $X_{o\emptyset}$.

On pourra dire que l'arbre $X_{\alpha}.\theta^{\alpha}_{[d(X)]}$ représente le non terminal X amputé des variables dont les indices ne sont pas dans α, ou encore X limité à la partie $x(\alpha)$ de ses variables. Ces amputations pourront être représentées par la substitution linéaire descendante σ de $T(\Sigma \cup V)$ dans $T(\Sigma \cup \hat{V})$ qui à tout non terminal de V associe tous les non terminaux limités possibles et ne modifie pas les terminaux : $\forall \, \varepsilon \in \Sigma$, $\sigma(\varepsilon) = \{\varepsilon\}$ et $\forall \, X \in V$, $\sigma(X) = \{X_{\alpha}.\theta^{\alpha}_{[d(X)]} \mid \alpha \subset [d(X)]\}$.

On a alors évidemment $P_{\alpha}(\sigma(X)) = \{X_{\alpha}.\theta^{\alpha}_{[d(X)]}\}$ et $R_{\alpha}(P_{\alpha}(\sigma(X))) = \{X_{\alpha}\}$.

Il est alors assez naturel d'associer à toute règle $(X \to t) \in R$ les pseudo-règles $X_\alpha.\theta^\alpha_{[d(X)]} \to \tau$ avec $\tau \in \sigma(t)$ et $var(\tau) \subset var(X_\alpha.\theta^\alpha_{[d(X)]}) = x(\alpha)$ car on a $X_\alpha.\theta^\alpha_{[d(X)]} \in P_\alpha(\sigma(X))$ et $\tau \in P_\alpha(\sigma(t))$. On pourra en faire des règles véritables en renumérotant les variables par R_α : $R_\alpha(X_\alpha.\theta^\alpha_{[d(X)]}) = X_\alpha$, on obtient alors les règles $X_\alpha \to \tau'$ avec $\tau' = R_\alpha(\tau)$ donc $\tau' \in R_\alpha(P_\alpha(\sigma(t)))$.

Il est clair que $\sigma(X_o) = \{\bar{X}_o\}$, nous pouvons alors définir $G'' = \langle \hat{V}, \Sigma, \bar{X}_o, R'' \rangle$ avec $R'' = \{(X_\alpha \to \tau) \mid (X \to t) \in R,\ X_\alpha \in \hat{X},\ \bar{t} \in R_\alpha(P_\alpha(\sigma(t)))\}$.
D'après le corollaire 22 et sa remarque, nous pouvons déterminer $\bar{V} = \{X_\alpha \in \hat{V} \mid F(G'', X_\alpha) \neq \emptyset\}$.
Nous poserons $\mathring{V} = \hat{V} \setminus \bar{V}$, $\mathring{V}$ est l'ensemble des non terminaux engendrant une forêt vide dans G''.

Après avoir montré que $F(G) = F(G'')$, nous montrerons que toute règle $(Y \to t) \in R''$ avec une occurrence de symbole de $\mathring{V}$ dans t peut être retirée de R'' sans changer la forêt engendrée. Nous aboutirons ainsi à la grammaire $G_1 = \langle \hat{V}, \Sigma, \bar{X}_o, R_1 \rangle$ avec $R_1 = \{(X \to t) \in R'' \mid t \in T(\Sigma \cup \bar{V})\}$.
Il suffira alors de remarquer que tous les symboles de $\mathring{V}$ sont devenus inacessibles dans G1 pour voir que la réduction de la non accessibilité nous fournira une grammaire totale.

A tout arbre $t \in T(\Sigma \cup V)^1_p$ nous pouvons associer tous les arbres qui en sont des images amputées dans $T(\Sigma \cup \hat{V})$, c'est à dire $\sigma(t)$ et sélectionner ceux qui ont leurs variables dans une partie $x(\alpha)$ de $var(t)$, une telle partie est $P_\alpha(\sigma(t))$. De même nous pouvons isoler les arbres de $F(G, t)$ qui ont leurs variables dans $x(\alpha)$, nous notons $P_\alpha(F(G, t))$ cette partie.

Nous allons montrer que $P_\alpha(F(G'', \sigma(t))) = P_\alpha(F(G, t))$. Ceci sera du à deux propriétés. Tout d'abord le choix P_α peut commuter avec la dérivation dans G" : $P_\alpha(F(G'', \sigma(t))) = F(G'', P_\alpha(\sigma(t)))$, ensuite si $P_\alpha(F(G, t))$ n'est pas vide, il existe un arbre limité aux variables d'indices dans α : $\tau \in P_\alpha(\sigma(t))$, qui engendre exactement $P_\alpha(F(G, t))$ dans G", c'est à dire que $P_\alpha(F(G, t)) = F(G'', \tau)$.

Nous montrerons celà par induction sur n, en construisant F(G) et F(G") par les suites S_G^n et $S_{G''}^n$. Pour simplifier les preuves nous commencerons par nous limiter aux arbres initiaux.

<u>Lemme II,3</u> Si pour tout $X_\alpha \in \hat{V}$ on a $S_{G''}^n(X_\alpha . \theta^\alpha_{[d(X)]}) = P_\alpha(S_{G''}^n(\sigma(X))) = P_\alpha(S_G^n(X))$ alors pour tout $\tilde{t} \in \mathcal{T}(\Sigma \cup \tilde{V})_p^1$ et tout $\beta \subset [p]$ on a $S_{G''}^n(P_\beta(\sigma(\tilde{t}))) = P_\beta(S_{G''}^n(\sigma(\tilde{t}))) = P_\beta(S_G^n(\tilde{t}))$ de plus si $P_\beta(S_G^n(\tilde{t})) \neq \emptyset$ il existe $\tau \in P_\beta(\sigma(\tilde{t}))$ tel que $S_{G''}^n(\tau) = S_{G''}^n(P_\beta(\alpha(\tilde{t})))$.

<u>Preuve</u>

A) Remarquons d'abord que si $\tau' \in S_{G''}^n(\tau)$ on a $var(\tau') \subset var(\tau)$ d'où $S_{G''}^n(P_\alpha(\sigma(\tilde{t}))) \subset P_\alpha(S_{G''}^n(\sigma(\tilde{t})))$.

B) Si $\tilde{t} \in \mathcal{T}(\Sigma)_p^1$, $\sigma(\tilde{t}) = \{\tilde{t}\} = S_{G''}^n(\tilde{t}) = S_G^n(\tilde{t})$.

a) Si $var(\tilde{t}) \subset x(\alpha)$ alors $P_\alpha(\tilde{t}) = \{\tilde{t}\}$ et $S_{G''}^n(P_\alpha(\sigma(\tilde{t}))) = P_\alpha(S_{G''}^n(\alpha(\tilde{t}))) = P_\alpha(S_G^n(\tilde{t})) = \{\tilde{t}\}$

b) Si $var(\tilde{t}) \not\subset x(\alpha)$ alors $P_\alpha(\tilde{t}) = \emptyset$ et les trois membres de l'égalité sont vides, donc égaux.

C) si $|\tilde{t}| \leq 1$ alors le lemme est vrai. Ceci découle immédiatement de B et du fait que si $\tilde{t} = X(x_1, \ldots x_p) \in V$, pour tout $\alpha \subset [p]$ on a $P_\alpha(\sigma(X)) = \{X_\alpha . \theta^\alpha_{[p]}\}$.

D) Nous avons montré que le lemme était vrai si $|\tilde{t}| < 2$, supposons qu'il soit vrai si $|\tilde{t}| < \ell$ et prenons $\tilde{t}$ de profondeur $\ell \geq 2$. On peut écrire $\tilde{t} = X(t_1, \dots t_q) \in \tilde{T}(\Sigma \cup V)^1_p$. Compte tenu de A, il suffit de montrer que pour tout $\beta \subset [p]$, $P_\beta(S^n_{G''}(\sigma(\tilde{t}))) \subset P_\beta(S^n_G(\tilde{t}))$, ce sera le point a), et qu'il existe $\tau \in P_\beta(\sigma(\tilde{t}))$ tel que $P_\beta(S^n_G(\tilde{t})) \subset S^n_{G''}(\tilde{t})$, ce sera le point b).

a) $\sigma(\tilde{t}) = \sigma(X) @ [\sigma(t_1), \dots \sigma(t_q)]$ et d'après le lemme 14, $S^n_{G''}(\sigma(\tilde{t})) \subset S^n_{G''}(\sigma(X)) @ [S^n_{G''}(\sigma(t_1)), \dots S^n_{G''}(\sigma(t_q))]$. Par hypothèse, pour tout $\alpha \subset [d(X)]$ on a $P_\alpha(S^n_{G''}(\sigma(X))) = P_\alpha(S^n_G(X))$ donc $S^n_{G''}(\sigma(X)) = S^n_G(X)$, d'où $S^n_{G''}(\sigma(\tilde{t})) \subset S^n_G(X) @ [S^n_{G''}(\sigma(t_1)), \dots S^n_{G''}(\sigma(t_q))]$ et pour tout $\beta \subset [p]$ on a $P_\beta(S^n_{G''}(\sigma(t))) \subset S^n_G(X) @ [P_\beta(S^n_{G''}(\sigma(t_1))), \dots P_\beta(S^n_{G''}(\sigma(t_q)))]$ et $P_\beta(S^n_G(\tilde{t})) = S^n_G(X) @ [P_\beta(S^n_G(t_1)), \dots P_\beta(S^n_G(t_q))]$. Il nous suffit alors de montrer que pour tout $i \in [q]$ on a $P_\beta(S^n_{G''}(\sigma(t_i))) \subset P_\beta(S^n_G(t_i))$. Le seul point qui nous empêche d'utiliser l'hypothèse de récurrence est que t_i n'est pas initial. Cependant nous avons $t_i = \tilde{t}_i.\theta^{\alpha_i}_{[p]}$ avec $\alpha_i = \{p_{i-1}+1, p_{i-1}+2, \dots p_i\}$ car t_i est un sous arbre final de l'arbre initial $\tilde{t}$. On a $S^n_{G''}(\sigma(t_i)) = S^n_{G''}(\sigma(\tilde{t}_i)).\theta^{\alpha_i}_{[p]}$ et $S^n_G(t_i) = S^n_G(\tilde{t}_i).\theta^{\alpha_i}_{[p]}$. Comme $\alpha_i \subset \beta$, on a $P_\beta(S^n_{G''}(\sigma(t_i))) = P_{\alpha_i}(S^n_{G''}(\sigma(t_i)))$ et $P_\beta(S^n_G(t_i)) = P_{\alpha_i}(S^n_G(t_i))$, d'où $P_\beta(S^n_{G''}(\sigma(t_i))) = P_{\alpha_i}(S^n_{G''}(\sigma(\tilde{t}_i)).\theta^{\alpha_i}_{[p]})$ et $P_\beta(S^n_{G''}(S^n_G(t_i))) = P_{\alpha_i}(S^n_G(\tilde{t}_i).\theta^{\alpha_i}_{[p]})$.
Posons $\alpha'_i = \{j \in [d(\alpha_i)]$ avec $\theta^{\alpha_i}_{[p]}(j) \in \alpha_i\}$, alors $P_{\alpha_i}(S^n_{G''}(\sigma(\tilde{t}_i)).\theta^{\alpha_i}_{[p]}) = (P_{\alpha'_i}(S^n_{G''}(\sigma(\tilde{t}_i)))).\theta^{\alpha_i}_{[p]}$ et $P_{\alpha_i}(S^n_G(\tilde{t}_i).\theta^{\alpha_i}_{[p]}) = (P_{\alpha'_i}(S^n_G(\tilde{t}_i))).\theta^{\alpha_i}_{[p]}$. Par hypothèse de récurrence on a $P_{\alpha'_i}(S^n_{G''}(\sigma(\tilde{t}_i))) = P_{\alpha'_i}(S^n_G(\tilde{t}_i))$ car $|\tilde{t}_i| < \ell$. Donc $P_\beta(S^n_{G''}(\sigma(t_i))) = P_\beta(S^n_G(t_i))$ et $P_\beta(S^n_{G''}(\sigma(\tilde{t}))) \subset P_\beta(S^n_G(\tilde{t}))$.

b) Nous avons

$$\begin{aligned}P_\beta(S^n_G(\tilde{t})) &= P_\beta(S^n_G(X) @ [S^n_G(t_1), \dots S^n_G(t_q)]) \\ &= S^n_G(X) @ [P_\beta(S^n_G(t_1)), \dots P_\beta(S^n_G(t_q))].\end{aligned}$$

Montrons que pour tout $i \in [q]$ on a $P_\beta(S^n_G(t_i)) = S^n_{G''}(P_\beta(\sigma(t_i)))$.

Ici encore, c'est seulement parce que t_i n'est pas initial que l'on ne peut appliquer l'hypothèse de récurrence. Reprenons les notations de a) à savoir $\tilde{t}_i$, $\theta^{\alpha_i}_{[p]}$, α_i, α'_i. On a $S^n_{G''}(P_\beta(\sigma(t_i))) = S^n_{G''}(P_\beta(\sigma(\tilde{t}_i).\theta^{\alpha_i}_{[p]})) =$ $S^n_{G''}(P_{\alpha_i}(\sigma(\tilde{t}_i).\theta^{\alpha_i}_{[p]}))$ et puisque $P_{\alpha_i}(\sigma(\tilde{t}_i).\theta^{\alpha_i}_{[p]}) = (P_{\alpha'_i}(\sigma(\tilde{t}_i))).\theta^{\alpha_i}_{[p]}$, on obtient $S^n_{G''}(P_\beta(\sigma(t_i))) = S^n_{G''}(P_{\alpha'_i}(\sigma(\tilde{t}_i))).\theta^{\alpha_i}_{[p]}$. De même $P_\beta(S^n_G(t_i))) =$ $(P_{\alpha'_i}(S^n_G(\tilde{t}_i))).\theta^{\alpha_i}_{[p]}$ et en appliquant à $\tilde{t}_i$ l'hypothèse de récurrence : $S^n_{G''}(P_\beta(\sigma(t_i))) = P_\beta(S^n_G(t_i))$. Alors, par hypothèse de récurrence, si $P_\beta(S^n_G(t_i)) \neq \emptyset$, il existe $\tau_i \in P_\beta(\sigma(t_i))$ avec $S^n_{G''}(\tau_i) = S^n_{G''}(P_\beta(\sigma(t_i))) =$ $P_\beta(S^n_G(t_i))$. Posons $\alpha = \{i \in [q] \mid P_\beta(S^n_G(t_i)) \neq \emptyset\}$. On a alors $P_\beta(S^n_G(\tilde{t})) = P_\alpha(S^n_G(X)) \,\theta\, [P_\beta(S^n_G(t_1)), \dots P_\beta(S^n_G(t_q))]$, et par hypothèse $S^n_{G''}(X_\alpha.\theta^\alpha_{[q]}) = P_\alpha(S^n_G(X))$. Ecrivons $\alpha = \{i_1, i_2, \dots i_r\}$, nous obtenons

$$P_\beta(S^n_G(\tilde{t})) = S^n_{G''}(X_\alpha) \,\theta\, [P_\beta(S^n_G(t_{i_1})), \dots P_\beta(S^n_G(t_{i_r}))]$$
$$= S^n_{G''}(X_\alpha) \,\theta\, [S^n_{G''}(\tau_{i_1}), \dots S^n_{G''}(\tau_{i_r})]$$

Car pour tout $i_j \in \alpha$, il existe $\tau_{i_j} \in P_\beta(\sigma(t_{i_j}))$ avec $S^n_{G''}(\tau_{i_j}) = P_\beta(S^n_G(t_{i_j}))$. D'où $P_\beta(S^n_G(\tilde{t})) = S^n_{G''}(X_\alpha(\tau_{i_1}, \dots \tau_{i_r}))$ et $X_\alpha(\tau_{i_1}, \dots \tau_{i_r}) \in P_\beta(\sigma(\tilde{t}))$.

□

<u>Lemme II,4</u> Pour tout $X_\alpha \in \hat{V}$ et pour tout $n \in \mathbb{N}$ on a $S^n_{G''}(X_\alpha.\theta^\alpha_{[d(X)]}) =$ $P_\alpha(S^n_{G''}(\sigma(X))) = P_\alpha(S^n_G(X))$.

<u>Preuve</u> Par récurrence sur n.

Si n = 0, le lemme est trivialement vrai par vacuité. Supposons le vrai pour tout $n < N$ et prenons $n = N > 0$. On a toujours $\{X_\alpha.\theta^\alpha_{[d(X)]} = P_\alpha(\sigma(X))$ et si $t \in S^n_{G''}(X_\alpha.\theta^\alpha_{[a(X)]})$ alors $var(t) \subset x(\alpha)$ d'où $t \in P_\alpha(S^n_{G''}(\sigma(X))$ et $S^n_{G''}(X_\alpha.\theta^\alpha_{[d(X)]}) \subset P_\alpha(S^n_{G''}(\sigma(X)))$. Si $t \in S^n_{G''}(\sigma(X))$, $\exists\, X_\beta.\theta^\beta_{[d(x)]}$ avec

$t \in S^{n}_{G''}(X_\beta.\theta^{\beta}_{[d(X)]})$ donc $R_\beta(t) \in S^{n}_{G''}(X_\beta)$ et par définition de $S^{n}_{G''}$ il existe $(X_\beta \to \bar{\tau}) \in R''$ avec $R_\beta(t) \in S^{n-1}_{G''}(\bar{\tau})$ donc $t \in S^{n-1}_{G''}(\bar{\tau}.\theta^{\beta}_{[d(X)]})$. Par définition de R'', il existe $(X \to \tau) \in R$ avec $\bar{\tau}.\theta_{[d(x)]} \in \sigma(\tau)$ donc $t \in S^{n-1}_{G''}(\sigma(\tau))$. Posons $\tau = \tilde{\tau}.\theta$ alors $t \in S^{n-1}_{G''}(\sigma(\tilde{\tau})).\theta$ et par hypothèse de récurrence et par le lemme II,3 on a $S^{n-1}_{G''}(\sigma(\tilde{\tau})) = S^{n-1}_{G}(\tilde{\tau})$ donc $t \in S^{n-1}_{G}(\tau) \subset S^{n}_{G}(X)$. D'où $S^{n}_{G''}(\sigma(X)) \subset S^{n}_{G}(X)$. Si $t \in P_\alpha(S^{n}_{G}(X))$ alors il existe $(X \to \tau) \in R$ avec $t \in S^{n-1}_{G}(\tau)$ posons $\tau = \tilde{\tau}.\theta$ avec $\tilde{\tau} \in \tilde{T}(\Sigma \cup V)^{1}_{p}$, $t \in S^{n-1}_{G}(\tilde{\tau}).\theta$ posons $t = t'.\theta$ il existe $\alpha' \subset [p]$ tel que $var(t') = x(\alpha')$ et $t' \in P_{\alpha'}(S^{n-1}_{G}(\tilde{\tau}))$ donc $t' \in S^{n-1}_{G''}(P_{\alpha'}(\sigma(\tilde{\tau})))$ d'après le lemme II,3 et par hypothèse de récurrence. On a alors $t \in S^{n-1}_{G''}(P_{\alpha'}(\sigma(\tilde{\tau}))).\theta \subset S^{n-1}_{G''}(P_{\alpha}(\sigma(\tau)))$ et comme $P_{\alpha'}(S^{n-1}_{G}(\tilde{\tau})) \neq \emptyset$, car il contient t', il existe $\bar{\tau} \in P_{\alpha'}(\sigma(\tilde{\tau}))$ tel que $t' \in S^{n-1}_{G''}(\bar{\tau})$ d'où $t \in S^{n-1}_{G''}(\bar{\tau}.\theta)$ et $\bar{\tau}.\theta \in P_\alpha(\sigma(\tau))$. et la règle $X_\alpha \to R_\alpha(\bar{\tau}.\theta)$ est dans R'' et comme $\bar{\tau}.\theta = (R_\alpha(\bar{\tau}.\theta)).\theta^{\alpha}_{[d(X)]}$ on a $t \in S^{n}_{G''}(X_\alpha.\theta^{\alpha}_{[d(X)]})$. On en déduit que $P_\alpha(S^{n}_{G}(X)) \subset S^{n}_{G''}(X_\alpha.\theta^{\alpha}_{[d(X)]})$.

□

<u>Lemme II,5</u> Si $Z \in \mathring{V}$ et $r = (Y \to \tilde{u}.[\vec{v}_1, Z.\vec{v}, \vec{v}_2]) \in R''$, posons $G''_1 = \langle\hat{V}, \Sigma, \bar{X}_o, R'' \setminus r\rangle$, alors $\forall\, Y \in \hat{V}$, $F(G'', Y) = F(G''_1, Y)$.

<u>Preuve</u>

Du lemme II,4 on déduit que si $\alpha \subset \beta \subset [d(X)]$, pour tout n $S^{n}_{G''}(X_\alpha.\theta^{\alpha}_{[d(X)]}) = P_\alpha(S^{n}_{G''}(X_\beta.\theta^{\beta}_{[d(X)]}))$ donc $F(G'', X_\alpha.\theta^{\alpha}_{[d(X)]}) = P_\alpha(F(G'', X_\beta.\theta^{\beta}_{[d(X)]}))$ et

* $F(G'', X_\alpha.\theta^{\alpha}_{\beta}) = \{t \subset F(G'', X_\beta) \mid var(t) \subset \theta^{\alpha}_{\beta}(\alpha)\}$

Si la règle $r = (Y \to \tilde{u}.[\vec{v}_1, z.\vec{v}, \vec{v}_2])$ est dans R'', ou bien aucune occurrence de non terminal n'est au dessus de Z et alors $F(G'', \tilde{u}.[\vec{v}_1, z.\vec{v}, \vec{v}_2]) = \emptyset$ et la règle r peut être retirée sans changer $F(G'', Y)$, ou bien il y a une occurrence X_β de non terminal au dessous de Z,

$\tilde{u}.[\vec{v}_1, z.\vec{v}, \vec{v}_2] = \tilde{v}.[\vec{\omega}_1, X_\beta(\tau_1, \dots \tau_k, \dots \tau_p), \vec{\omega}_2]$ avec $\tau_k = \tilde{v}'.[\vec{\omega}'_1, z.v, \vec{\omega}'_2]$

et il n'y a aucune occurrence de non terminal au dessus de Z dans τ_k, c'est à dire que X_β est l'occurrence de non terminal la plus proche au-dessus de Z. Alors $F(G'', \tau_k) = \emptyset$. Posons $\alpha = \beta \setminus \theta^{\beta}_{d(X)}(k)$.
$F(G'', X_\beta(\tau_1, \ldots \tau_k, \ldots \tau_p)) = F(G'', X_\beta) \circledast [F(G'', \tau_1), \ldots F(G'', \tau_k) \ldots F(G'', \tau_p)] = \{t \in F(G'', X_\beta) \mid k \notin var(t)\} \circledast [F(G'', \tau_1), \ldots F(G'', \tau_k), \ldots F(G'', \tau_p)]$ et d'après * $= F(G'', X_\alpha.\theta^{\alpha}_{\beta}) \circledast [F(G'', \tau_1), \ldots F(G'', \tau_k), \ldots F(G'', \tau_p)] = F(G'', X_\alpha) \circledast [F(G'', \tau_1), \ldots F(G'', \tau_{k-1}), F(G'', \tau_{k+1}), \ldots F(G'', \tau_p)]$. On a alors $F(G'', \tilde{\nu}.[\vec{\omega}_1, X_\beta(\tau_1, \ldots \tau_k, \ldots \tau_p), \vec{\omega}_2]) = F(G'', \tilde{\nu}.[\vec{\omega}_1, X_\alpha(\tau_1, \ldots \tau_{k-1}, \tau_{k+1}, \ldots \tau_p), \vec{\omega}_2])$ et la règle $r' = (Y \to \tilde{\nu}.[\vec{\omega}_1, X_\alpha(\tau_1, \ldots \tau_{k-1}, \tau_{k+1}, \ldots \tau_p), \vec{\omega}_2])$ est dans R", on peut donc retirer r de R" sans changer F(G", Y). Il est alors évident que l'on ne change pas F(G", Y') pour tout $Y' \in \hat{V}$.

□

<u>Lemme II,6</u> Il existe G' totale avec F(G') = F(G).

<u>Preuve</u>

Du lemme II,4 on tire immédiatement $F(G) = P_\emptyset(F(G, X_o)) = P_\emptyset(F(G'',\bar{X}_o))=F(G'')$.
L'application du lemme II,5 à toute occurence de $Z \in \mathring{V}$ dans une partie droite de règle de G", conduit à G_1 donc F(G") = F(G1).
Tout symbole de $\mathring{V}$ est devenu inaccessible depuis $\bar{X}_o$ dans G_1. Il suffit alors d'appliquer le lemme II,1 pour obtenir G'. Puisque les non terminaux de G' sont dans $\bar{V} = \hat{V} \setminus \mathring{V}$, il est clair que G' est totale.

□

<u>Proposition II,9</u> Pour toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$ telle $F(G) \neq \emptyset$ on peut trouver une grammaire G' totale et sans non terminal inaccessible telle que F(G) = F(G') et telle que si G est linéaire ou stricte ou à non

strictitude monadique ou complète ou régulière ou corégulière ou Greibach haut ou Greibach bas ou Greibach double, G' a les mêmes propriétés.

Preuve

Si G n'est pas complète, la construction qui précède pourra être appliquée avec profit. En effet σ est linéaire, stricte, ce qui conservera dans G" les propriétés de linéarité, strictitude et non strictitude monadique. Comme de plus $\forall X \in V$, $\sigma(X) \subset T(\hat{V})$ et $\forall \alpha \in \Sigma$, $\sigma(\alpha) \subset T(\Sigma)$ les propriétés de régularité, corégularité et formes de Greibach seront conservées dans G". Comme les règles de G' sont un sous ensemble des règles de G", ces propriétés seront conservées dans G'.

Il est clair que σ n'est pas complète, on pourrait montrer que toutes les règles non-complètes introduites dans G' quand G est complète, peuvent être retirées,mais il est plus simple d'utiliser une transformation particulière de G, s'appuyant sur sa complétude qui est une propriété forte.

Il suffit de remarquer que si G est complète et si Y est un non terminal engendrant une forêt vide : $F(G, Y) = \emptyset$, alors toute règle de la forme $X \to t = \tilde{u}.[\vec{v}_1, Y.\vec{v}, \vec{v}_2]$ est inutile car pour tout $\tau \in F(G, \tilde{u})$ on a $var(\tau) = var(\tilde{u})$ car G est complète, on en déduit que $F(G, t) = \emptyset$ et donc que si la règle est utilisée dans une dérivation $X_o \overset{*}{\Rightarrow} \tilde{\nu}.[\vec{\nu}_1, t.\vec{\omega}, \vec{\nu}_2] = t'$, pour tout $\tau' \in F(G, \tilde{\nu})$ on a $var(\tau') = var(\tilde{\nu})$ donc t ne pourra être éliminé et $F(G, t') = \emptyset$. Il est évident qu'on ne change pas la forêt engendrée en retirant une règle qui ne peut être utilisée dans aucune dérivation terminale.

Posons $G'' = \langle V, \Sigma, X_o, R' \rangle$

avec $\bar{V} = \{X \in V \mid F(G, X) \neq \emptyset\}$,

$R' = \{(X \to t) \in R \mid X \in \bar{V} \text{ et } t \in T(\Sigma \cup \bar{V})\}$.

G" est la grammaire obtenue en retirant ces règles inutiles. On remarque qu'alors tout non terminal engendrant une forêt vide dans G est inaccessible dans G", il suffit alors de construire G' en rendant G" sans non terminal inaccessible. Toutes les règles de G' étant des règles de G, il est évident que les propriétés de G sont conservées dans G'.

□

4) REDUCTION DES ARITES INUTILES EN OI

L'idée de base de cette réduction tient à la remarque suivante. Si x_i est inutile dans X, c'est à dire que $\forall$ t $\in$ F(G, X), $x_i \notin$ var(t), alors pour toute règle ($X \to \tau$), pour tout t $\in$ F(G, τ), $x_i \notin$ var(t). Si par chance, $x_i \notin \tau$ pour aucune règle ($X \to \tau$),le problème est simple et il suffit de réduire X et de renuméroter convenablement les variables de τ. Généralement on trouve ($X \to \tau$) avec $x_i \in$ var(τ) alors dans τ chaque occurrence de x_i se trouve dans un sous arbre de τ qui est le $k^{ième}$ fils de non terminal Y dans lequel x_k est inutile (en supposant que la grammaire soit totale et sans non terminal inaccessible). Si l'on sait réduire Y on pourra le remplacer par sa forme réduite dans τ, ainsi le sous arbre contenant l'occurrence de x_i aura disparu de τ et l'on sera ramené au cas où τ ne contient aucune occurrence de x_i. Il est clair que les réductions de X et Y peuvent dépendre mutuellement l'une de l'autre, aussi leurs réductions ne pourront être faites une à une. Pour cette raison on commencera par déterminer la forme de chaque non terminal réduit, c'est à dire son arité et la façon de renuméroter ses variables puis on fera le remplacement partout dans la grammaire.

Le problème des réductions d'arités inutiles a été résolu par Irène Guessarian [Gu] dans le cadre des schémas de programmes déterministes qui sont à peu de choses près des grammaires déterministes. Dans ce cadre la difficulté tient essentiellement à ce que l'on prend en compte des arbres infinis alors qu'ici elle réside dans le non déterminisme. Boudol appelle réduction forte la réduction des arités inutiles conjointe à celle de la totalité qu'il appelle réduction faible. Il propose une construction passant par une transformation de la grammaire du langage de branches associé à la forêt. Sa construction nécessite une première transformation de la grammaire d'arbres, visant à éviter qu'un non terminal n'engendre que des arbres de hauteur nulle. Nous avons remarqué sur l'exemple suivant :

$X_o \to ZX(0, 1)$; $X(x_1, x_2) \to x_1 + x_2$; $Z(x_1) \to Zb(x_1, x_1) + b(x_1, x_1)$;

que cette première transformation conduisait à la grammaire

$X_o \to Z\bar{X}$; $\bar{X} \to 0{+}1$; $Z(x_1) \to Zb(x_1, x_1) + b(x_1, x_1)$, qui n'engendre pas la même forêt. Les arguments qu'il donne dans sa preuve comptent parmi ceux qu'il utilise pour prouver la validité de sa construction réalisant la réduction faible, aussi avons nous jugé utile d'en donner une autre preuve. Il est probable que sa construction réalisant la réduction forte nécessite peu de modification, cependant cette construction se prêterait mal à notre propos qui est, outre la réalisation des réductions, d'obtenir la conservation de certaines propriétés.

<u>Construction II,2</u> Soit $G = \langle V, \Sigma, X_o, R\rangle$ une grammaire algébrique totale et sans non terminal inaccessible.

Posons pour tout $X \in V$, $E(X) = \{x_i \in \mathrm{var}(X) \mid \exists\, t \in F(G, X)$ avec $x_i \in \mathrm{var}(t)\}$, le degré de X étant fini, le corollaire 22 nous montre que E(X) peut éffectivement être construit.

Si $d(X) = n$ et $E(X) = \{x_{i_1}, x_{i_2}, \dots x_{i_p}\}$ avec pour tous $j, k \in [p]$, $j < k$ entraine $i_j < i_k$ posons $\theta_X = \langle n ; x_{i_1}, \dots x_{i_p}\rangle \in \Theta_n^p$ et ρ le démarquage linéaire ordonné strict injectif défini par

$\forall\, a \in \Sigma$, $\rho(a) = a$ et $\forall\, X \in V$, $\rho(X) = \bar{X}.\theta_X$ où $\bar{X}$ est un symbole d'arité égale à Card(E(X)) et n'est ni dans V ni dans Σ.

$G' = \langle \bar{V}, \Sigma, \bar{X}_o, \bar{R}\rangle$ avec $\bar{V} = \{\bar{X} \mid X \in V\}$

et $\bar{R} = \{(\bar{X} \to \tau) \mid (X \to t) \in R,\ \rho(X) = \bar{X}.\theta_X,\ \rho(t) = \tau.\theta_X\}$.

X_o étant d'arité 0 on a $\rho(X_o) = \bar{X}_o$.

G" est obtenue depuis G' en supprimant les non-terminaux devenus inaccessibles.

Nous allons montrer que G" est réduite et F(G) = F(G").

__Lemme II,10__ Pour toute torsion linéaire $\theta \in \Theta^p_q$ et tout arbre $t \in T(\Sigma \cup V)^1_q$ avec $var(t) \subset var(\theta)$ il existe un et un seul arbre $t' \in T(\Sigma \cup V)^1_p$ tel que $t = t'.\theta$.

__Preuve__

θ étant linéaire on peut trouver $\theta' = \langle p \,;\, x_{i_1}, \ldots x_{i_q} \rangle \in \Theta^q_p$ telle que pour tout $x_j \in var(\theta)$, $i_j = \theta^{-1}(j)$ on a alors $t.\theta' = t'.\theta.\theta' = t'$ et comme $var(t) \subset var(\theta)$ t' ne dépend pas du choix fait dans θ' pour i_k avec $x_k \notin var(\theta)$.

□

__Corollaire II,11__ Pour toute règle $(X \to t) \in R$ il existe une et une seule règle $(\bar{X} \to \tau) \in \bar{R}$ avec $\rho(t) = \tau.\theta_X$.

Ceci découle directement du fait que θ_X est linéaire et montre que $\bar{R}$ est fini.

__Lemme II,12__ Si pour tout $X \in V$, $S^n_{G'}(\rho(X)) = S^n_G(X)$ alors pour tout $t \in T(\Sigma \cup V)$, $S^n_{G'}(\rho(t)) = S^n_G(t)$.

__Preuve__

Par récurrence sur la profondeur de t,

Si $|t| = 0$ alors $t = x_i$, $\rho(x_i) = x_i$, $S^n_{G'}(x_i) = S^n_G(x_i) = \{x_i\}$.

Si $|t| > 0$ alors $t = X(t_1, \ldots t_p)$ avec $\forall\, i \in [p]$, $|t_i| < |t|$ et

si $X \in \Sigma$, $S^n_G(t) = \{X\} \circledast [S^n_G(t_1), \ldots S^n_G(t_p)]$

$S^n_{G'}(\rho(t)) = \{X\} \circledast [S^n_{G'}(\rho(t_1)), \ldots S^n_{G'}(\rho(t_p))] = S^n_G(t)$ par hypothèse de récurrence.

Si $X \in V$ $\quad S_G^n(t) = S_G^n(X) \circledast [S_G^n(t_1), S_G^n(t_p)]$

$S_{G'}^n(\rho(t)) = S_{G'}^n(\rho(X)) \circledast [S_{G'}^n(\rho(t_1)), \ldots S_{G'}^n(\rho(t_p))] = S_G^n(t)$ par hypothèses de récurrence et car $S_{G'}^n(\rho(X)) = S_G^n(X)$ par hypothèse. Ceci est du au fait que le composé d'un homomorphisme par une substitution descendante est une substitution descendante [AE].

□

<u>Lemme II,13</u> Pour tout $t \in T(\Sigma \cup V)$ on a $F(G, t) = F(G', \rho(t))$.

<u>Preuve</u>

Il suffit d'après le lemme 18 de montrer par récurrence sur n que pour tout $n \in \mathbb{N}$ et tout $t \in T(\Sigma \cup V)$ on a $S_G^n(t) = S_{G'}^n(\rho(t))$. Ce qui est évident si $n = 0$ car alors si $t \notin T(\Sigma)$ on a $S_G^o(t) = S_{G'}^o(\rho(t)) = \emptyset$ et si $t \in T(\Sigma)$ on a $S_G^o(t) = S_{G'}^o(\rho(t)) = \{t\}$. Si on a pour tout $t \in T(\Sigma \cup V)$, $S_G^{n-1}(t) = S_{G'}^{n-1}(\rho(t))$ alors par définiton de S_G^n on a pour tout $X \in V$,

$$\star \quad S_G^n(X) = \bigcup_{(X\to\tau)\in R} S_G^{n-1}(\tau)$$

$$S_{G'}^n(\rho(X)) = S_{G'}^n(\bar{X}.\theta_X) = \bigcup_{(\bar{X}\to\bar{\tau})\in\bar{R}} S_{G'}^{n-1}(\bar{\tau}.\theta_X)$$

$$= \bigcup_{(X\to t)\in R} S_{G'}^{n-1}(\rho(t)) \text{ d'après le corollaire II,11}$$

$$= S_G^n(X) \text{ par hypothèse de récurrence et } \star.$$

D'après le lemme II,13 on a $S_{G'}^n(\rho(t)) = S_G^n(t)$.

□

<u>Lemme II,14</u> G" est réduite et engendre la même forêt que G.

<u>Preuve</u>

D'aprés le lemme II,13 on a $F(G, X_o) = F(G', \rho(X_o)) = F(G', \bar{X}_o)$, de plus si $F(G', \bar{X}) = \emptyset$ alors $F(G', \bar{X}.\theta_X) = \emptyset$ et $F(G, X) = \emptyset$ donc si G est totale, G' aussi.

Il est clair que la méthode de réduction des non terminaux inaccessibles qui n'introduit aucune règle nouvelle et aucun non terminal nouveau conserve la totalité et l'utilité de G'.

□

<u>Théorème II,15</u> Pour toute grammaire G telle que $F(G) \neq \emptyset$ il existe une grammaire G' réduite telle que $F(G) = F(G')$ et que si G est linéaire ou stricte ou à non strictitude monadique ou complète ou règulière ou de Greibach haut, bas ou double, G' a les mêmes propriétés.

<u>Preuve</u>

Le lemme II,6 nous dit qu'on peut supposer G totale et sans terminaux inaccessibles.

L'homomorphisme ρ de la construction II,2 étant linéaire, strict, les propriétés de linéarité, strictitude et non strictitude monadique de G seront conservées dans G". Comme de plus $\forall X \in V$, $\rho(X) \subset T(\bar{V})$ et $\forall \alpha \in \Sigma$, $\rho(\alpha) \in T(\Sigma)$, les propriétés de régularité, corégularité et formes de Greibach seront conservées dans G".

Le but de la construction II,2 étant de retirer les arités inutiles des non terminaux, ρ n'est pas complet mais il suffit de remarquer que si G est complète et totale, la construction II,2 est sans objet car alors G est aussi sans arité inutile. En effet si G est complète on a pour tout non terminal X et toute règle $X \to t$, $var(X) = var(t)$. On en déduit par induction sur $|\tau|$ que si $\tau \Rightarrow t$ alors $var(\tau) = var(t)$ et par induction sur la longueur des dérivations que si $\tau \overset{*}{\Rightarrow} t$ alors $var(\tau) = var(t)$. D'où si $t \in F(G, X)$, c'est à dire $X \overset{*}{\Rightarrow} t \in T(\Sigma)$ alors $var(X) = var(t)$ donc toute variable de X est utile si $F(G, X) \neq \emptyset$, ce qui est le cas si G est totale. La suppression des non terminaux inutiles ne pose encore aucun problème.

□

5) INTERET DES GRAMMAIRES REDUITES

Il est souvent commode dans de nombreuses démonstrations de s'appuyer sur certaines propriétés que doivent vérifier tous les arbres pour pouvoir se dériver en un arbre terminal, par exemple, un symbole d'une certaine classe ne peut se trouver qu'immédiatement sous un symbole d'une autre classe. Il est clair que si dans un arbre peuvent se glisser des sous arbres qui soit disparaîtront au cours de toute dérivation terminale, soit interdiront toute dérivation terminale, la prise en compte de ces éléments inutiles gênera considérablement le raisonnement. De tels éléments inutiles ne peuvent se rencontrer lors de dérivations dans une grammaire réduite, c'est ce que montrent les résultats énoncés maintenant.

Lemme II,16 Si $G = \langle V, \Sigma, X_o, R\rangle$ est totale et sans arité inutile, pour tout $u \in T(\Sigma \cup V)^1_n$ et tout $x_i \in var(u)$, il existe $u' \in F(G, u)$ avec $x_i \in var(u')$.

Preuve immédiate par induction sur la profondeur de u.

□

Corollaire II,17 Si G est totale et sans arité inutile, pour tout arbre $t = \tilde{u}.[\vec{v}_1 \otimes v \otimes \vec{v}_2].\vec{w}$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)$ il existe $t' \in F(G, t)$ et $v' \in F(G, v)$ tel que v' soit un sous arbre de t'.

Preuve

Il suffit d'appliquer le lemme II,16 à $\tilde{u}$ et d'utiliser le lemme 15 pour décomposer t'.

□

Corollaire II,18 Si G est réduite, pour toute règle $X \to t$ de G, pour tout sous arbre v de t, il existe une dérivation terminale $X_o \overset{*}{\Rightarrow} \tilde{u}.v.\vec{w} \overset{*}{\Rightarrow} u'.v'.\vec{w}' \in T(\Sigma)$ avec $\tilde{u} \in \tilde{T}(\Sigma \in V)_1^1$, $\tilde{u} \overset{*}{\Rightarrow} u'$, $v \overset{*}{\Rightarrow} v'$, $\vec{w} \overset{*}{\Rightarrow} \vec{w}'$ et $x_1 \in var(u')$, c'est à dire où v est effectivement utilisé.

Preuve

G étant sans non terminal inaccessible, il existe une dérivation $X_o \overset{*}{\Rightarrow} \tilde{u}_1.X.\vec{w}_1$ d'où en appliquant la règle $(X \to t)$ on obtient l'existence de la dérivation $X_o \overset{*}{\Rightarrow} \tilde{u}.v.\vec{w}$. Il suffit alors d'appliquer le corollaire II,17.

□

6) EFFET DES REDUCTIONS EN IO

En appliquant la construction II,1 à la grammaire G7, on obtient G7R :

$$\bar{X}_o \to X_{<1>}(0) + X_{<1>}(1) + X_{<1,\ 2>}(1,\ 1)\ ;$$

$$X_{<1>}(x_1) \to a(x_1)\ ; \quad X_{<1,\ 2>}(x_1,\ x_2) \to b(x_1,\ x_2) + a(x_1)$$

On vérifie que F(G7R) = F(G7) et que G7R est totale, cependant $F_{IO}(G7R) = F(G7) \neq F_{IO}(G7)$, la construction II,1 ne conserve donc pas les forêts IO-engendrées. Il serait d'ailleurs vain de chercher une transformation conservant les deux forêts et assurant la IO-totalité. Il suffit de prendre par exemple G9 qui est réduite :

$$X_o \to X(o,\ X_o)\ , \quad X(x_1,\ X_2) \to b(x_1,\ x_2) + a(x_1)\ ,$$ et pour laquelle on a $F_{IO}(G9) = \emptyset$

donc G9 n'est pas IO-totale. Il est bien évident que pour toute grammaire $G = \langle V, \Sigma, \bar{X}_o, R\rangle$ avec $F(G) = F(G9) \neq \emptyset$ on ne pourra avoir à la fois G IO-totale et $F_{IO}(G) = F_{IO}(G9) = \emptyset$. Par contre si une grammaire est totale et IO-totale la construction II,2 conserve les forêts OI et IO et la grammaire obtenue est à la fois utile et IO-utile, il suffit essentiellement pour celà de démontrer l'équivalent en IO du lemme II,12.

On pourrait alors rechercher une construction spécifique visant à réduire le IO-totalité, le chapitre suivant nous montrera que celà n'apporterait rien de nouveau car cette réduction découlera directement de la construction rendant la grammaire complète.

III - REDUCTIONS PARTICULIERES EN IO

1) INTRODUCTION

Nous allons montrer comment on peut transformer une grammaire de façon à la rendre complète, stricte et IO-réduite en conservant la forêt IO-engendrée. Nous commencerons par la rendre complète car cette propriété qui est très forte permettra d'obtenir facilement la strictitude et la IO-réduction. Les phénomènes de duplication peuvent empêcher de rendre la grammaire ordonnée, nous en donnons un exemple qui pourrait être démontré en utilisant le théorème de duplication [ADd]. Ces phénomènes de duplication n'intervenant pas dans le cas de règles linéaires, nous obtiendrons un résultat plus fort pour les grammaires linéaires. Un résultat équivalent sera obtenu pour les forêts OI-engendrées par une grammaire linéaire en profitant du fait que pour une grammaire G linéaire IO-totale on a $F_{IO}(G) = F(G)$.

La réduction de la non complétude est basée sur la remarque suivante : Si dans une dérivation ascendante, une règle $X \to t$ non complète, c'est à dire qu'il existe $x_i \in var(X)$ tel que $x_i \notin var(t)$, est utilisée, on pourrait imaginer d'ajouter à la grammaire une règle complète $\bar{X} \to t.\theta$ avec $d(\bar{X}) = card(var(t))$ et θ servant à renuméroter les variables dans t. Observons alors la règle ρ qui a produit l'occurrence de X dans la dérivation et substituons $\bar{X}.\theta$ à X dans cette règle. De cette façon nous aurons anticipé l'abandon des sous arbres provoqué par l'application de $X \to t$, de façon à ce que cet abandon se produise au moment de la production de X. Il se peut alors que la règle construite depuis ρ soit elle-même non complète, on pourra alors reprendre le procédé en remontant dans la dérivation jusqu'à l'axiome. L'axiome est d'arité 0 donc toutes les règles qui lui sont appliquées sont complètes, on aura alors substitué à la dérivation ascendante de départ une autre dérivation ascendante donnant le même résultat et utilisant uniquement des règles complètes. Nous généraliserons cette transformation en une transformation uniforme opérant sur la grammaire. On notera que la transformation apportée à la règle ρ, qui consiste à y reporter une partie de l'effet de l'application de la règle $(X \rightarrow t)$ n'est possible que parce que, les dérivations étant ascendantes, on est assuré que le sous arbre ayant pour racine l'occurrence de X ne sera pas dupliqué avant d'avoir été dérivé en un arbre terminal. Autrement dit, on peut choisir quelles dérivations appliquer à X avant de dupliquer puisque le choix sera le même pour toutes les occurrences. En notant T ces transformations appliquées aux noeuds des arbres et T(t) le résultat de toutes ces transformations sur un arbre t on aura $T(\tilde{u}.\vec{v}) = t(\tilde{u}).T(\vec{v})$. Ayant $F_{IO}(G, \tilde{u}.\vec{v}) = F_{IO}(G, \tilde{u}) \mathbin{\textcircled{$\uparrow$}} F_{IO}(G, \vec{v})$, en notant G' la grammaire transformée, il faudra avoir entre autres $F_{IO}(G', T(\tilde{u}.\vec{v})) = F_{IO}(G, T(\tilde{u})) \mathbin{\textcircled{$\uparrow$}} F_{IO}(G, T(\vec{v}))$ pour que la forêt engendrée en IO soit conservée, pour cela il est nécessaire que $T(\tilde{u})$ soit linéaire donc T linéaire et ce sera le cas. Par contre pour avoir $F_{OI}(G', T(\tilde{u}.\vec{v})) =$ $F_{OI}(G', T(\tilde{u})) \mathbin{\textcircled{$\uparrow$}} F_{OI}(G', T(\vec{v}))$ il faut que $F_{OI}(G')$ soit linéaire et ce ne sera le cas que si G' est linéaire.

2) REDUCTION DE LA NON COMPLETUDE EN IO

Construction III,1

Soit $G = \langle V, \Sigma, X_o, R\rangle$ la grammaire à transformer.

A toute partie $\alpha \subset \mathbb{N}^+$ contenant les $d(\alpha)$ éléments $i_1, i_2, .. i_{d(\alpha)}$ et à tout entier $n \geq i_{d(\alpha)}$, on fait correspondre la torsion $\theta_{n\alpha} = \langle n; x_{i_1}, .. x_{i_{d(\alpha)}}\rangle \in \Theta_n^{d(\alpha)}$ linéaire ordonnée.

Pour tout non terminal $X \in V$ on définit $\hat{X} = \{X_\alpha \mid \alpha \in P([d(X)]$ et $d(X_\alpha) = d(\alpha)\}$ et on pose $\hat{V} = \bigcup_{X \in V} \hat{X}$. Si X est d'arité 0, $\hat{X}$ contient un seul élément. On notera $\bar{X}_o$ le seul élément de $\hat{X}_o$.

Définissons la substitution σ de $T(\Sigma \epsilon V)$ dans $P_D(T(\Sigma \cup \hat{V}))$ en posant $\forall\ \varepsilon \in \Sigma$, $\sigma(\varepsilon) = \{\varepsilon\}$ et $\forall\ X \in V$, $\sigma(X) = \{X_\alpha . \theta_{d(X)\alpha} \mid \alpha \in P([d(X)])\}$. σ est linéaire, on la considérera comme descendante.

On pose $G' = \langle \hat{V}, \Sigma, \bar{X}_o, R'\rangle$ où $\bar{X}_o$ est l'unique élément de $\sigma(X_o)$ et $R' = \{X_\alpha \to \tau \mid (X\ t) \in R, \tau . \theta_{d(X)\alpha} \in \sigma(t)$ et $var(\tau) = var(X_\alpha)\}$ c'est à dire que G' est très semblable à la grammaire G" de la construction II,1, la seule différence est que R' est l'ensemble des règles complètes de G", G' est donc complète.

Nous allons montrer que $F_{IO}(G) = F_{IO}(G')$.

<u>Lemme III,1</u> Si $\forall\ X \in V$ on a $\bar{S}^n_G(X) = \bar{S}^n_{G'}(\sigma(X))$ alors pour tout $t \in T(\Sigma \cup V)$ on a $\bar{S}^n_G = \bar{S}^n_{G'}(\sigma(t))$.

<u>Preuve</u> : Par récurrence sur la profondeur de t.

Si $|t| = 0$ alors $t = x_i$, $\sigma(x_i) = \{x_i\}$ et $\bar{S}^n_G(x_i) = \bar{S}^n_{G'}(x_i) = \{x_i\}$.

Si $|t| > 0$ alors $t = X(t_1, .. t_p)$ avec $\forall\ i \in [p]$, $|t_i| < |t|$ et si $X \in \Sigma$ alors $\bar{S}^n_G(t) = \{X\} \circledast [\bar{S}^n_G(t_1), .. \bar{S}^n_G(t_p)]$ et $\sigma(t) = \{X\} \circledast [\sigma(t_1), .. \sigma(t_p)]$ d'où $\bar{S}^n_{G'}(\sigma(t)) = \{X\} \circledast [\bar{S}^n_{G'}(\sigma(t_1)), .. \bar{S}^n_{G'}(\sigma(t_p))] = \bar{S}^n_G(t)$ par hypothèse de récurrence.

Si $X \in V$ alors $\bar{S}^n_G(t) = \bar{S}^n_G(X) \circledast [\bar{S}^n_G(t_1), .. \bar{S}^n_G(t_p)]$ et

$\sigma(t) = \sigma(X) \circledast [\sigma(t_1), .. \sigma(t_p)]$ comme σ est linéaire, $\sigma(X)$ est linéaire et d'après le lemme 12 on a $\bar{S}^n_{G'}(\sigma(t)) = \bar{S}^n_{G'}(\sigma(X)) \theta [\bar{S}^n_{G'}(\sigma(t_1)), \ldots \bar{S}^n_{G'}(\sigma(t_p))]$ de l'hypothèse $\bar{S}^n_G(X) = \bar{S}^n_{G'}(\sigma(X))$ et de l'hypothèse de récurrence sur $|t|$ on tire alors $\bar{S}^n_{G'}(\sigma(t)) = \bar{S}^n_G(t)$.

□

<u>Lemme III,2</u> $\forall n \in \mathbb{N}$, $\forall X \in V$ on a $\bar{S}^n_G(X) = \bar{S}^n_{G'}(\sigma(X)$.

<u>Preuve</u>

Par récurrence sur n.

Par définition de $\bar{S}^n_G$ on a $\bar{S}^o_G(X) = \bar{S}^o_{G'}(X) = \emptyset$.

Supposons le lemme vrai si $n < m$.

Par définition de S^n_G on a :

Si $t \in \bar{S}^m_G(X)$ alors $\exists (X \to \tau) \in R$ et $t \in \bar{S}^{m-1}_G(\tau)$ alors l'hypothèse de récurrence permet d'appliquer le lemme III,1 et on a $t \in \bar{S}^{m-1}_{G'}(\sigma(\tau))$, il existe donc $\tau' \in \sigma(\tau)$ avec $t \in \bar{S}^{m-1}_{G'}(\tau')$ et il existe $\theta_{d(X)\alpha}$ avec $var(\theta_{d(X)\alpha}) = var(\tau')$. Posons τ'' l'arbre de $T(\Sigma \cup \bar{V})^1_{d(\alpha)}$ tel que $\tau' = \tau''.\theta_{d(X)\alpha}$, par définition de G' la règle $X_\alpha \to \tau''$ est dans R' donc $t \in \bar{S}^m_{G'}(X_\alpha) \subset \bar{S}^m_{G'}(\sigma(X))$, d'où $\bar{S}^m_G(X) \subset \bar{S}^m_{G'}(\sigma(X))$.

Réciproquement si $t \in \bar{S}^m_{G'}(\sigma(X))$ il existe $X_\alpha.\theta_{d(X)\alpha} \in \sigma(X)$ tel que $t \in \bar{S}^m_{G'}(X_\alpha.\theta_{d(X)\alpha})$ donc $t' \in \bar{S}^m_{G'}(X_\alpha)$ avec $t = t'.\theta_{d(X)\alpha}$. Par définition de $\bar{S}^m_{G'}$ il existe une règle $(X_\alpha \to \tau) \in R'$ avec $t' \in \bar{S}^{m-1}_{G'}(\tau)$ et par définition de G' il existe une règle $(X \to \tau') \in R$ avec $\tau.\theta_{d(X)\alpha} \in \sigma(\tau')$ par hypothèse de récurrence et par le lemme III,1 on a $\bar{S}^{m-1}_G(\tau') = \bar{S}^{m-1}_{G'}(\sigma(\tau'))$ donc $t \in \bar{S}^{m-1}_G(\tau')$ et par définition de $\bar{S}^m_G$ on a $t \in \bar{S}^m_G(X)$.

□

<u>Proposition III,3</u> Pour toute grammaire $G = \langle V, \Sigma, X_o, R\rangle$ avec $F_{IO}(G) \neq \emptyset$, on peut trouver une grammaire G' complète IO-réduite conservant les propriétés (linéarité, strictitude, non strictitude monadique, complètude, régularité,

corégularité, formes de Greibach) de G et telle que $F_{IO}(G) = F_{IO}(G')$.

Preuve

Si G n'est pas complète on pourra lui substituer G' de la construction III,1 qui est complète. Du lemme III,2 on tire immédiatement $\bar{S}^n_G(X_o) = \bar{S}^n_{G'}(\bar{X}_o)$ pour tout entier n d'où $F_{IO}(G) = F_{IO}(G')$. Puisque R' de la construction III,1 est une partie de R" de la construction II,1, les propriétés de G sont conservées dans G' comme dans G". Il est clair que si G est complète on prendra G' = G. Si G' est complète, on la rendra IO-totale par suppression des mêmes règles inutiles supprimées pour la rendre totale dans la proposition II,9. En fait on pourrait montrer qu'une grammaire complète et totale est IO-totale. Nous avons vu dans le théorème II.15 qu'une grammaire complète et totale est d'arité utile . De la même façon on obtient qu'une grammaire complète et IO-totale est d'arités IO-utiles. La suppression des non terminaux inaccessibles ne pose aucun problème particulier et permet d'obtenir le résultat qui découle du fait qu'une grammaire complète et réduite est IO-réduite.

□

3) REDUCTION DE LA NON STRICTITUDE EN IO

Construction III,2

Soit $G = \langle V, \Sigma, X_o, R\rangle$ complète et IO-réduite. Si elle est non stricte, elle est à non strictitude monadique :

Si $(X \to x_i) \in R$ elle doit être complète donc $X \in V_1$ et $x_i = x_1$.

De même si $X \overset{*}{A\Rightarrow} x_i$ on a $X \in V_1$ et $x_i = x_1$.

Posons $N = \{X \in V \mid X \overset{*}{a\Rightarrow} x_1\}$

Le fait d'ajouter à R toutes les règles $(X \to x_1)$ avec $X \in N$ ne change pas $F_{IO}(G)$, nous supposerons donc que R contient toutes ces règles.

Donnons nous la substitution linéaire σ de $T(\Sigma \cup V)$ dans $T(\Sigma \cup V)$ définie par : $\forall X \in N \quad \sigma(X) = \{X, x_1\}$, $\forall X \in \Sigma \cup V \setminus N$, $\sigma(X) = \{X\}$ et que nous considérerons comme ascendante.

Posons alors $G' = \langle V, \Sigma, X_o, R'\rangle$

avec $R' = \{(X \to t) \mid (X \to \tau) \in R \text{ et } t \in \sigma(\tau) \setminus \{x_1\}\}$.

Si N peut effectivement être construit, la construction de G' est effective.

Lemme III,4 N peut effectivement être construit.

Preuve

Posons $N_1 = \{X \in V \mid (X \to x_1) \in R\}$ et pour tout $i \in \mathbb{N}^+$

$N_{i+1} = \{X \in V \mid \exists X_1, X_2, .. X_p \in N_i \text{ et } (X \to X_1 \; X_2 .. X_p \, x_1) \in R\}$

la suite N_i est croissante et bornée par V_1 et si $N_i = N_{i+1}$ alors

$\forall p \in \mathbb{N}$, $N_i = N_{i+p}$ donc sa limite $\hat{N}$ peut effectivement être construite.

Il est clair que N est la limite de la suite

$\{X \in V \mid x_1 \in \bar{S}^n_G(X)\}$. Montrons par récurrence sur n que

$N_n = \{X \in V \mid x_1 \in \bar{S}^n_G(X)\}$, ce qui est évident pour n = 1 puisque

$\bar{S}^1_G(X) = \{\tau \in T(\Sigma) \mid (X \to \tau) \in R\}$. Supposons que ce soit vrai pour tout $n < m$.

Si $X \in N_m$, $\exists\, X_1, \ldots X_p \in N_{m-1}$ et $(X \to X_1 X_2 .. X_p) \in R$ d'où par hypothèse de récurrence : $\forall\, i \in [p]$, $x_1 \in \bar{S}_G^{m-1}(X_i)$ donc $x_i \in \bar{S}_G^{m-1}(X_1 X_2 .. X_p)$ et $x_1 \in \bar{S}_G^{m}(X)$. Réciproquement si $x_1 \in \bar{S}_G^{m}(X)$ il existe $(X \to \tau) \in R$ avec $x_1 \in \bar{S}_G^{m-1}(\tau)$. Par récurrence sur $|\tau|$ on en déduit que τ est composé uniquement de non terminaux d'arité 1, $X_1, X_2 \ldots X_p$ donc $\tau = X_1, X_2 .. X_p$ et pour tout $i \in [p]$, $x_1 \in \bar{S}_G^{m-1}(X_i)$ donc par hypothèse de récurrence $X_i \in N_{m-1}$et $X \in N_m$. On obtient donc $\hat{N} = N$ et $\hat{N}$ peut être construit.

□

<u>Lemme III,5</u> Si pour tout $X \in V$, $\bar{S}_G^n(X) = \bar{S}_G^n(\sigma(X))$ alors pour tout $t \in T(\Sigma \cup V)$, $\bar{S}_G^n(t) = \bar{S}_{G'}^n(\sigma(t))$.

<u>Preuve</u>

Par récurrence sur $|t|$. Le lemme est vrai si $|t| = 0$ car $\bar{S}_G^n(x_i) = \bar{S}_{G'}^n(x_i) = \sigma(x_i) = \{x_i\}$. Si $|t| > 0$ alors $t = X(t_1, \ldots t_p)$.
Si $X \in \Sigma$, $\bar{S}_G^n(t) = \{X\} \circledast [\bar{S}_G^n(t_1), \ldots \bar{S}_G^n(t_p)]$, $\sigma(t) = \{X\} \circledast [\sigma(t_1),..\sigma(t_p)]$ et $\bar{S}_{G'}^n(\sigma(t)) = \{X\} \circledast [\bar{S}_{G'}^n(\sigma(t_1)), \ldots, \bar{S}_{G'}^n(\sigma(t_p))]$ et comme pour tout $i \in [p]$, $|t_i| < |t|$ par hypothèse de récurrence on a $\bar{S}_{G'}^n(\sigma(t_i)) = \bar{S}_G^n(t_i)$ donc $\bar{S}_G^n(t) = \bar{S}_{G'}^n(t)$. Si $X \in V$, $\bar{S}_G^n(t) = \bar{S}_G^n(X) \circledast [\bar{S}_G^n(t_1), .. \bar{S}_G^n(t_p)]$, $\sigma(t) = \sigma(X) \circledast [\sigma(t_1),.. \sigma(t_p)]$ et σ étant linéaire on a $\bar{S}_{G'}^n(\sigma(t)) = \bar{S}_{G'}^n(\sigma(X)) \circledast [\bar{S}_{G'}^n(\sigma(t_1)), .. \bar{S}_{G'}^n(\sigma(t_p))]$ d'après le lemme 12. Par hypothèse $\bar{S}_{G'}^n(\sigma(X)) = \bar{S}_G^n(X)$ donc par hypothèse de récurrence on obtient $\bar{S}_G^n(t) = \bar{S}_{G'}^n(\sigma(t))$.

□

<u>Lemme III,6</u> $F_{IO}(G) = F_{IO}(G')$.

Preuve

Puisque $\sigma(X_o) = \{X_o\}$ il suffit de montrer par récurrence sur n que pour tout $n \in \mathbb{N}$ et tout $X \in V$ on a $\bar{S}^n_G(X) = \bar{S}^n_{G'}(\sigma(X))$. Ajoutons $\bar{S}^n_G(X) \setminus \{x_1\} = \bar{S}^n_{G'}(X)$ pour obtenir l'hypothèse de récurrence. L'hypothèse est vraie si $n = 0$.

Supposons que ce soit vrai pour tout $i < n$.

On a $\bar{S}^n_G(X) = \{t \in \bar{S}^{n-1}_G(\tau) \mid (X \to \tau) \in R\}$

Si $X \in V \setminus N$ on a $\sigma(X) = \{X\}$, $\bar{S}^n_{G'}(X) = \{t \in \bar{S}^{n-1}_{G'}(\tau) \mid (X \to \tau) \in R'\}$

d'où par définition de R', $\bar{S}^n_{G'}(X) = \{t \in \bar{S}^{n-1}_{G'}(\sigma(\tau)) \mid (X \to \tau) \in R'\}$

du lemme III,5 on tire alors $\bar{S}^n_G(X) = \bar{S}^n_{G'}(X) = \bar{S}^n_{G'}(\sigma(X))$.

Si $X \in N$ on a $\sigma(X) = \{X, x_1\}$ d'où $\bar{S}^n_{G'}(\sigma(X)) = \bar{S}^n_{G'}(X) \cup \{x_1\}$.

$\bar{S}^n_G(X) = \{t \in \bar{S}^{n-1}_G(\tau) \mid (X \to \tau) \in R\}$

$\bar{S}^n_G(X) \setminus \{x_1\} = \{t \in \bar{S}^{n-1}_G(\tau) \setminus \{x_1\} \mid (X \to \tau) \in R\}$

$$\begin{aligned}
\bar{S}^n_{G'}(X) &= \{t \in \bar{S}^{n-1}_{G'}(\tau) \mid (X \to \tau) \in R'\} \\
&= \{t \in \bar{S}^{n-1}_{G'}(\sigma(\tau) \setminus \{x_1\}) \mid (X \to \tau) \in R\} \text{ par définition de R'} \\
&= \{t \in \bar{S}^{n-1}_{G'}(\sigma(\tau)) \setminus \{x_1\} \mid (X \to \tau) \in R\} \text{ car G' est stricte.} \\
&= \{t \in \bar{S}^{n-1}_{G}(\tau) \setminus \{x_1\} \mid (X \to \tau) \in R\} = \bar{S}^n_G(X) \setminus \{x_1\}
\end{aligned}$$

et donc $\bar{S}^n_{G'}(\sigma(X)) = \bar{S}^n_G(X)$.

□

Théorème III,7 Pour toute grammaire G avec $F_{IO}(G) \neq \emptyset$ on peut trouver G' complète stricte et réduite telle que $F_{IO}(G') = F_{IO}(G)$ et que les propriétés de linéarité, régularité, corégularité et formes de Greibach de G soient conservées dans G'.

Preuve

La proportion III,3 montre qu'on peut supposer G complète et réduite. La construction III,2 s'applique alors. Il est clair que G' est stricte et comme σ est complète, G' est complète. σ étant linéaire, la linéarité de G se retrouve dans G' et comme de plus pour tout $\alpha \in \Sigma$, $\sigma(\alpha) = \{\alpha\}$ les autres propriétés sont conservées.

G' étant complète il est facile de la rendre réduite sans changer ses propriétés par la même méthode que dans la proposition III,3.

□

La forêt $c(a^i \#, a^j \#, a^i \#, a^j \#)$ qui est corégulière permettrait de montrer qu'on ne peut pas ajouter au théorème III,7 la condition supplémantaire que G' soit ordonnée.

4) LE CAS DES GRAMMAIRES LINEAIRES

Construction III,3 Soit $G = \langle V, \Sigma, X_o, R\rangle$ linéaire complète stricte réduite. Pour tout $n \in \mathbb{N}$ posons $P(n)$ l'ensemble des $n!$ bijections de $[n]$ dans $[n]$ que l'on assimilera aux torsions linéaires complètes de Θ_n^n. Pour tout $X \in \mathbb{N}$ posons $\hat{X} = \{\overset{\theta}{X} \mid \theta \in P(d(X))$ et $d(\overset{\theta}{X}) = d(X)\}$ et $\hat{V}$ la réunion des $\hat{X}$ pour $X \in V$. σ est une substitution ascendante linéaire complète stricte définie par $\forall\, \alpha \in \Sigma,\ \sigma(\alpha) = \{\alpha\}$ et $\forall\, X \in V,\ \sigma(X) = \{\overset{\theta}{X}.\theta \mid \theta \in P(d(X))\}$. On définit alors $G' = \langle\hat{V}, \Sigma, \bar{X}_o, R'\rangle$ avec $\bar{X}_o$ l'unique élément de $\sigma(X_o)$ et $R' = \{\overset{\theta}{X} \to \tilde{t} \mid (X \to \tau) \in R,\ \tilde{t}.\theta \in \sigma(\tau)$ et $\tilde{t} \in \mathcal{T}(\Sigma \cup \hat{V})\}$. Il est facile de voir que G' est linéaire complète ordonnée, elle est stricte car G l'est. On pourra la réduire facilement par simple suppression de non terminaux et de règles. On montre que $F_{IO}(G) = F_{IO}(G')$ à peu près de la même façon que dans la construction III,1. La propriété essentielle est que pour tout $X \in V$ et toute dérivation $X \overset{*}{\underset{G}{\Rightarrow}} t$, pour tout $x_i \in var(X)$ le nombre d'occurrences de x_i dans t est exactement 1, ce qui permet d'écrire $t = \tilde{t}.\theta$ avec $\tilde{t} \in \mathcal{T}(\Sigma \cup V)^1_{d(X)}$ et $\theta \in P(d(X))$.

Théorème III,8 Si G est une grammaire linéaire et si $F(G) \neq \emptyset$ (resp. $F_{IO}(G) \neq \emptyset$) on peut trouver G' (resp G'') linéaire complète stricte ordonnée et réduite telle que $F(G)=F(G')$ (resp $F_{IO}(G)=F_{IO}(G''))$ et que les propriétés de régularité, corégularité et formes de Greibach de G soient conservées dans G' (resp G'').

Preuve

Si G est linéaire, on peut trouver G_1 totale conservant les propriétés de G et telle que $F(G) = F(G_1)$ par la construction II,1, si elle n'est pas IO-totale on pourra la rendre trivialement IO-totale en ajoutant pour tout non terminal X tel que $F_{IO}(G_1, X) = \emptyset$ une règle $(X \to t)$ avec $t \in R(G_1, X)$, on obtiendra alors G2 IO-totale conservant les propriétés de G et telle que F(G) = F(G2). G2 étant linéaire et IO-totale on a $F_{IO}(G2) = F_{OI}(G2)$, on peut alors trouver G'_1 linéaire complète stricte et réduite (théorème III,7) telle que $F_{IO}(G'1) = F_{IO}(G2)$ et comme G' est linéaire IO-totale, $F_{IO}(G'1) = F(G'1) = F(G2) = F(G)$. D'autre part le théorème III,7 permet de trouver G"1 linéaire complète stricte et réduite avec $F_{IO}(G) = F_{IO}(G''_1)$ et conservant les propriétés de G.

Le seul problème qui reste est de rendre G'1 et G"1 ordonnées.

Ceci sera réalisé par la construction III,3.

□

IV - PROBLEME DE LA STRICTITUDE ET DE LA COMPLETUDE EN OI

INTRODUCTION

Nous avons vu au chapitre III que l'on pouvait toujours trouver une grammaire complète, stricte et réduite engendrant $F_{IO}(G)$, quelque soit la grammaire G. L'intérêt des grammaires ayant ces propriétés vient du fait que les dérivations y sont croissantes, c'est à dire que si t => t' par une règle croissante on a $|t| \leq |t'|$ et aussi la largeur de t est inférieure ou égale à la longueur de t'. La hauteur croissante est liée à la strictitude et la largeur croissante à la complètude.

Nous allons ici étudier le problème de la réduction de la non complétude et de la non strictitude avec conservation de la forêt OI-engendrée. Nous verrons que ces réductions sont généralement impossibles en dehors de certaines grammaires particulières comme les linéaires. L'expérience de la théorie des langages fait que l'on a généralement tendance à s'intéresser d'abord au problème de la strictitude. En fait la non strictitude rencontrée en théorie des langages est, par force, toujours monadique et nous verrons que la non strictitude monodique est toujours réductible. Les autres cas de non strictitude s'assortissent aussi de non complétude : soit $(X \to x_i)$ une règle non stricte, si $d(X) \neq 1$ la règle est nécessairement non complète.

Nous verrons que la non complétude n'est en général pas réductible et la non strictitude quelconque non plus. Ceci permet de penser que ce qui n'est pas réductible dans la non strictitude a trait à la non complétude qui lui est liée. En fait nous verrons que dans certains cas ces deux propriétés peuvent s'échanger. Ainsi pour la forêt BF_3 nous donnerons une grammaire complète et une grammaire stricte, par contre nous montrerons qu'il n'existe pas de grammaire G à la fois complète et stricte telle que $F(G) = BF_3$.

Nous verrons qu'il existe des forêts algébriques qui ne peuvent être engendrées par aucune grammaire complète, telle BF_2 et des grammaires algébriques qui ne peuvent être engendrées par aucune grammaire stricte, telle BF_1. Il nous suffira de remarquer que toute grammaire complète est nécessairement à non strictitude monadique pour conclure que BF_1 ne peut être engendrée par une grammaire complète. En effet cette grammaire complète serait alors à non strictitude monadique et nous avons vu que celle-ci est réductible. On en conclura que la complétude est une propriété plus forte que la strictitude.

Pour prouver que BF_1, BF_2, BF_3 sont bien les contre exemples recherchés, nous utiliserons un lemme qui mettra en évidence certaines contraintes sur les dérivations produisant des arbres arbitrairement larges. Ces contraintes sont assez difficiles à exprimer, c'est pourquoi nous nous limiterons à des arbres construits autour d'arbres balancés arbitrairement larges. Le fait que ces arbres soient balancés n'est pas essentiel mais permet, outre une expression plus commode, des preuves rendues plus aisées pour certaines propriétés remarquables des arbres balancés qui ont déjà été mises à profit dans certaines preuves [AD-ϕ^{-1}].

Afin d'allèger la preuve du lemme il nous sera utile de supposer que les grammaires que nous utiliserons, seront réduites au sens du chapitre II et que de plus, dans toute dérivation descendante, l'apparition d'une règle non complète ou non stricte sera aussi tardive que possible. Nous dirons que les dérivations initiales y sont complètes et strictes, en donnant à l'appellation de dérivation initiale la même signification que dans [AD-d].

Enfin nous montrerons, en nous appuyant encore sur des arbres balancés arbitrairement larges, que par homomorphisme linéaire inverse on peut obtenir une forêt non algébrique depuis une forêt algébrique complète et stricte. Ce résultat est un peu plus fort que celui de [AD-ϕ^{-1}] et permet d'avancer un peu dans la recherche des classes fermées par bimorphismes linéaires.

1) REDUCTION DE LA NON STRICTITUDE MONADIQUE

Nous allons commencer par mettre la grammaire sous une forme plus simple, en particulier pour tout non terminal X tel que $X \overset{*}{\Rightarrow} x_1$ nous allons construire un nouveau non terminal $\hat{X}$ tel que $F(G, \hat{X}) = F(G, X) \setminus x_1$. De cette façon on pourra remplacer toutes les règles applicables à X par $X \to \hat{X}$ et $X \to x_1$. Il est alors évident que si le problème de duplication ne se posait pas on pourrait rendre la grammaire stricte en anticipant l'application des règles $X \to \hat{X} + x_1$ dans la grammaire, comme en théorie des langages. Les difficultés liées aux duplications nous conduiront à une transformation plus complexe.

Pour que les constructions soient effectives nous aurons besoin de pouvoir décider si $X \overset{*}{\Rightarrow} x_1$, le lemme suivant montre que l'on peut décider si $X \overset{*}{\Rightarrow} x_i$.

<u>Lemme IV,1</u> Etant donnée une grammaire $G = \langle V, \Sigma, X_o, R\rangle$ on peut construire pour tout $X \in V$, $z(X) = \{x_i \in var(X) \mid X \underset{G}{\overset{*}{\Rightarrow}} x_i\}$.

<u>Construction</u> Posons σ_o une substitution linéaire descendante définie par $\forall \alpha \in \Sigma$, $\sigma_o(\alpha) = \emptyset$ et $\forall X \in V$, $\sigma_o(X) = \{x_j \mid (X \to x_j) \in R\}$.
Et pour tout $i > 0$, définissons la substitution linéaire descendante σ_i par :
$\forall \alpha \in \Sigma$, $\sigma_i(\alpha) = \emptyset$ et $\forall X \in V$
$\sigma_i(X) = \sigma_{i-1}(X) \cup \{x_j \mid \exists (X \to t) \in R$ tel que $x_j \in \sigma_{i-1}(t)\}$.
Si $\tau \in \sigma_i(t)$ on dira que $t \overset{*}{\Rightarrow} \tau$ de profondeur $i+1$.

<u>Preuve</u>

Par induction sur i on montre

P : $x_j \in \sigma_i(X) \iff (X \overset{*}{\Rightarrow} x_j$ par une dérivation de longueur au plus $i+1)$.

En effet pour $i = 0$, P est vrai par définition de σ_o.

Si P est vrai pour i et si $x_j \in \sigma_{i+1}(X)$ alors il existe $(X \to t) \in R$ et $x_j \in \sigma_i(t)$. Explicitons la branche de t joignant la racine à l'occurrence de x_j conservée :

$$t = X_1(u_{1,1}, \dots u_{1,k_1-1}, X_2(u_{2,1},\dots u_{2,k_2-1}, X_3(\dots X_p(u_{p,1},\dots u_{p,k_p-1}, x_j, u_{p,k_{p+1}} \dots) \dots) \dots) \dots)$$

avec pour tout $\ell \in [p]$, $x_{k_\ell} \in \sigma_i(X_\ell)$ donc par hypothèse de récurrence $X_\ell \overset{*}{=>} x_{k_\ell}$ de profondeur au plus i+1 donc $X => t \overset{*}{=>} x_j$ de profondeur au plus i+2. Réciproquement si $X \overset{*}{=>} x_j$ de profondeur i+2, on a $(X \to t) \in R$ avec $t \overset{*}{=>} x_j$ de profondeur au plus i+1 donc $x_j \in \sigma_i(t)$ et $x_j \in \sigma_{i+1}(X)$.

La propriété P est donc prouvée.

Si on pose $\sigma_i(V) = \{\sigma_i(X) \mid X \in V\}$, il est immédiat que cette suite est croissante vis à vis de la relation d'ordre $\sigma_i(V) \leq \sigma_j(V) <=> \forall X \in V$ $\sigma_i(X) \subseteq \sigma_j(X)$. Chaque suite $\sigma(X)$ étant bornée par $\{x_1, x_2, \dots x_{d(X)}\}$ la suite $\sigma_i(V)$ est bornée, elle devient donc stationnaire à partir d'un indice i_o au plus égal à la somme des degrés des éléments de V.

D'après P, $z(X) = \sigma_{i_o}(X)$.

□

<u>Corollaire IV,2</u> Pour toute grammaire $G = <V, \Sigma, X_o, R>$ on peut effectivement construire $E_G = \{X \in V \mid \exists\, t \in F(G, X) \text{ avec } |t| = 0\}$.

<u>Preuve</u>

Si $t \in F(G, X)$ avec $|t| = 0$ alors $t = x_i$ avec $x_i \in var(X)$ d'où $x_i \in z(X)$ et $z(X)$ peut être construit.

□

Notation

E_G est l'ensemble des non terminaux de V qui peuvent s'effacer, c'est à dire que si $X \in E_G$ et $t = \tilde{u}.[\vec{v}_1, X(t_1, \dots t_p), \vec{v}_2]$ il existe au moins un $i \in [p]$ tel que $t \overset{*}{\Rightarrow} \tilde{u}.[\vec{v}_1, t_i, \vec{v}_2]$. On voit que si $d(X) > 1$, la non strictitude s'accompage de non complétude, par contre la non strictitude monadique est en quelque sorte la non strictitude pure, très voisine de la non strictitude rencontrée en théorie des langages.

Il serait gênant de pouvoir rencontrer des dérivations du genre $X \overset{*}{\Rightarrow} U \overset{*}{\Rightarrow} X$ avec $Y \neq X$. Pour éviter celà on fera en sorte que la relation de préordre $X \overset{*}{\Rightarrow} Y$ soit une relation d'ordre.

Lemme IV,3 On peut transformer G de façon que dans V la relation $X \overset{*}{\Rightarrow} Y$ soit une relation d'ordre.

Preuve

Si on a $X \overset{*}{\Rightarrow} Y$ et $Y \overset{*}{\Rightarrow} X$, puisque $F(G, X) = F(G, Y)$, on peut remplacer Y par X partout dans la grammaire pour obtenir G'. Cette transformation se fait par homomorphisme linéaire et on montre facilement que $F(G) = F(G')$ comme pour la construction II,2.

□

Le lemme suivant indique que si l'on ne peut pas en général transformer les règles d'une grammaire par application anticipée d'autres règles comme c'est l'usage en théorie des langages, par contre on peut faire l'opération inverse.

Lemme IV,4 Dans toute grammaire G réduite on peut remplacer toute règle $r = (Y \to \tilde{u}.[\tilde{u}_1 \otimes \tilde{u}_2 \otimes \dots \otimes \tilde{u}_p].\theta.\vec{w})$ par les règles $r' = (Y \to \tilde{u}.[\tilde{u}_1 \otimes \dots \otimes \tilde{u}_{i-1} \otimes X \otimes \tilde{u}_{i+1} \otimes \dots \otimes \tilde{u}_p].\theta.\vec{w})$ et $r'' = (X \to \tilde{u}_i)$ où X est un nouveau non terminal avec $var(X) = var(\tilde{u}_i)$. La nouvelle grammaire G' ainsi obtenue est encore réduite et engendre la même forêt que G.

Preuve

Puisque X apparaît seulement dans la règle r', pour avoir F(G) = F(G') il suffit d'avoir

$F(G, \tilde{u}.[\tilde{u}_1 \otimes \tilde{u}_2 \otimes \dots \otimes \tilde{u}_p.\theta.\vec{w}) = F(G', \tilde{u}.[\tilde{u}_1 \otimes \dots \otimes \tilde{u}_{i-1} \otimes X \otimes u_{i+1} \otimes \dots \otimes \tilde{u}_p].\theta.\vec{w})$, ce qui est immédiat car $F(G', X) = F(G, \tilde{u}_i)$.

G étant réduite, $F(G, \tilde{u}_i) \neq \emptyset$ donc $F(G', X) \neq \emptyset$, d'où G' est totale. Pour tout $x_j \in var(\tilde{u}_i)$, $\exists$ $t \in F(G, \tilde{u}_i)$ avec $x_j \in var(t)$ car G est sans arité inutile. Comme $var(X) = var(\tilde{u}_i)$ et $F(G', X) = F(G, \tilde{u}_i)$, X est sans arité inutile. Comme Y est accessible depuis l'axiome dans G, il est accessible dans G' et comme G' est sans arité inutile X est accessible dans G'.

□

Lemme IV,5 Pour toute grammaire G réduite à non strictitude monadique on peut trouver G' réduite à non strictitude monadique telle que pour toute règle $r = (Y \to t)$ de G' on ait $|t| \leq 2$.

Preuve

Cette transformation est apparentée à celle qui met une grammaire sous forme normale [AD-d], [Col], [Ma].

Transformation A :

Si dans G existe une règle $r = (Y \to t)$ avec $|t| > 2$, on peut écrire $t = \alpha.[\tilde{u}_1 \otimes \tilde{u}_2 \otimes \dots \otimes \tilde{u}_p].\theta$ avec $\alpha \in \Sigma \cup V$. Pour tout $\tilde{u}_i$ avec $|\tilde{u}_i| > 1$

introduisons le nouveau non terminal X_i avec $var(X_i) = var(\tilde{u}_i)$ et la règle $X_i \to \tilde{u}_i$. Pour tout $i \in [p]$ posons $u'_i = \tilde{u}_i$ si $|\tilde{u}_i| \leq 1$ et $u'_i = X_i$ sinon. Posons $t' = \alpha.[u'_1 \otimes \ldots \otimes u'_p].\theta$ et remplaçons r par $r' = (Y \to t')$. Le lemme IV,4 nous montre que G' est encore réduite et F(G) = F(G').

Il faut montrer que la non strictitude demeure monadique. Supposons qu'il existe X_i avec $x_k \in F(G', X_i) = F(G, \tilde{u}_i)$, alors G étant à non strictitude monadique, $\tilde{u}_i = Z_1 Z_2 Z_q$ avec $Z_1, Z_2, \ldots Z_q \in E_G \subset V_1$ et comme $\tilde{u}_i$ est initial on a $var(\tilde{u}_i) = \{x_1\}$ d'où $var(X_i) = \{x_1\}$ et $x_k = x_1$. Donc si $X_i \in E_{G'}$ alors $X_i \in V_1$ et G' est à non strictitude monadique.

Il est clair que $|t'| \leq 2$ et pour tout $i \in [p]$, $|\tilde{u}_i| < |t|$.

Soit h le plus grand entier tel qu'il existe $(Y \to t)$ dans G avec $|t| = h$, c'est à dire que h est la plus grande hauteur de règle dans G. Il suffit d'appliquer la transformation A à toutes les règles de hauteur h pour obtenir G' avec h-1 comme plus grande hauteur de règle. La transformation A est applicable si h > 2, il suffira de l'appliquer jusqu'à obtenir h = 2.

□

<u>Corollaire IV,6</u> On peut supposer que dans G réduite, à non strictitude monadique, tout $X \in E_G$ n'apparaisse en partie gauche que dans des règles de l'un des cinq types suivants :

a) $X \to x_1$

b) $X \to A.\vec{w}$ avec $A \in \Sigma \cup V \setminus E_G$

c) $X \to Y$ avec $Y \in E_G$

d) $X \to Y.u$ avec $Y \in E_G$ et $u \in T(\Sigma \cup V \setminus E_G) \setminus \{x_1\}$

e) $X \to Y.Z$ avec $Y, Z \in E_G$

et que de plus $X \overset{*}{\Rightarrow} Y$ soit une relation d'ordre dans E_G.

Preuve

Il suffit d'appliquer la transformation du lemme IV,5, alors $X \to u$ avec $|u| \leq 2$ est nécessairement de l'un des cinq types proposés. Il est clair que la transformation du lemme IV,3 ne change pas la hauteur des parties droites des règles, il suffit alors de l'appliquer à G.

$\square$

Lemme IV,7 On peut transformer G à non strictitude monadique de façon que pour tout $X \in E_G$ on ait la propriété P :

Les seules règles s'appliquant à X sont $X \to x_1$ et $X \to \hat{X}$ avec $\hat{X} \in V \setminus E_G$ donc $F(G, \hat{X}) = F(G, X) \setminus \{x_1\}$.

Preuve

On suppose que G a les propriétés du corollaire IV,6.

A) Montrons que la transformation est possible pour X tel que $\forall\, Y \in E_G$ avec $X \overset{*}{\Rightarrow} Y$, Y a la propriété P.

a) Puisque $X \overset{*}{\Rightarrow} x_1$, $X \in E_G$ et on peut ajouter la règle $X \to x_1$ si elle n'existe pas déjà. Introduisons le nouveau non terminal $\hat{X}$ d'arité 1 et la règle $X \to \hat{X}$.

b) Toute règle $X \to A.\vec{w}$ avec $A \in \Sigma \cup V \setminus E_G$ est remplacée par $\hat{X} \to A.\vec{w}$ ce qui ne change pas $F(G, X)$.

c) Toute règle $X \to Y$ avec $Y \in E_G$ est remplacée par $\hat{X} \to \hat{Y}$ où $\hat{Y}$ est défini car Y a la propriété P. On a toujours $X \Rightarrow x_1$, $X \Rightarrow \hat{X} \Rightarrow \hat{Y}$, $F(G, \hat{Y}) = F(G, Y) \setminus \{x_1\}$ donc $F(G, X)$ ne change pas.

d) Toute règle $X \to Y.u$ avec $Y \in E_G$ et $u \in T(\Sigma \cup V \setminus E_G) \setminus \{x_1\}$ est remplacée par $\hat{X} \to Y.u$. Puisque $|u| > 0$, $x_i \notin F(G, Y.u)$.

e) Toute règle $X \to Y.Z$ avec $Y, Z \in E_G$ est remplacée par les règles $\hat{X} \to \hat{Y}.Z$, $\hat{X} \to \hat{Z}$. On a $X \overset{*}{\Rightarrow} Y$ et $X \overset{*}{\Rightarrow} Z$ donc Y et Z ont la propriété P par hypothèse. On a alors $F(G, Y.Z) = F(G, \hat{Y}.Z) \cup F(G, \hat{Z}) \cup \{x_1\}$ et puisque $x_1 \in F(G, X)$, $F(G, X)$ n'est pas changé.

Après toutes ces transformations les seules règles qui s'appliquent à X sont $X \to x_1$ et $X \to \hat{X}$ et puisque $F(G, X)$ n'a pas changé et que $x_1 \notin F(G, \hat{X})$ on a $F(G, \hat{X}) = F(G, X) \setminus \{x_1\}$.

On notera que ces transformations conservent les propriétés du corollaire IV,6, on peut donc répéter leur application.

B) Il est clair que pour tout $X \in E_G$ tel qu'il existe aucun $Y \in E_G \setminus X$ avec $X \overset{*}{\Rightarrow} Y$, l'hypothèse de A est vérifiée et donc que la transformation est possible pour X. E_G est fini donc par récurrence sur E_G, s'appuyant sur la relation d'ordre $\overset{*}{\Rightarrow}$, la transformation A pourra être appliquée à E_G tout entier.

□

<u>Lemme IV,8</u> On peut transformer G de sorte que dans toute partie droite de règle, aucun élément de E_G ne soit au dessus d'un autre symbole.

<u>Preuve</u>

A la suite du lemme IV,7, pour tout $X \in E_G$, les seules règles s'appliquant à X sont $X \to x_1$ et $X \to \hat{X}$ avec $\hat{X} \notin E_G$. A la suite du lemme IV,5, toute partie droite de règle est de hauteur au plus 2, donc si $X \in E_G$ est au dessus d'un autre symbole, il en est le sommet. La règle s'écrit alors $Y \to X.\alpha$ avec $\alpha \in \Sigma \cup V \setminus E_G$ (en effet si $\alpha \in E_G$ alors $Y \overset{*}{\Rightarrow} x_1$ donc $Y \in E_G$ ce qui implique que $Y \to X.\alpha$ n'est pas une règle de G).

Il suffit de remplacer la règle $Y \to X.\alpha$ par les deux règles $Y \to \hat{X}.\alpha$ et $Y \to \alpha$, ceci ne change pas F(G) car on a $F(G, X) = F(G, \hat{X}) \cup \{x_1\}$ donc $F(G, X\alpha) = F(G, \hat{X}.\alpha) \cup F(G, \alpha)$. Cette transformation préserve les propriétés des lemmes IV,5 et IV,7 ce qui permet de l'appliquer à toutes les règles de G de la forme $Y \to X.\alpha$.

□

Nous allons maintenant introduire par des exemples les points principaux de la construction réduisant la non-strictitude monadique. Le passage de GIV6 à GIV7 constituera l'essentiel de cette transformation.

<u>Exemple</u> GIV1 qui est à non strictitude monadique :

```
Xo →  Y   ;   Y   →   Y     +    Z    ; Z  →   Z    +    b    ;
     / \     / \     / \         |      |      |        / \
    #   #   x1  x2  X   α        β      x1     b      x1   x1
                    |   |       / \           / \
                    x1  x2    x1   x2       x1   x1

      ^        ^
X  → x1 + X  ; X  → a
|         |    |    |
x1        x1   x1   x1
```

Elle a les propriétés des lemmes IV,7 et IV,8 et engendre la forêt $\{\tilde{b}^p.[\beta(a^{i_1}\, \#, \alpha^n\, \#), \beta(a^{i_2}\, \#, \alpha^n\, \#), \ldots \beta(a^{i_q}\, \#, \alpha^n\, \#)]$ avec $p \in \mathbb{N}^+$, $n \in \mathbb{N}$, $q = 2^p$ et pour tout $j \in [q]$, $i_j \leq n\}$.

Puisque F(GIV1, Y) n'est pas linéaire, la transformation qui consisterait à anticiper les dérivations applicables à X, ce qui donnerait GIV2 :

```
Xo →  Y   ;   Y   →   Y    +   Y    +   Z   , Z  →   Z    +    b   ,
     / \     / \     / \      ^/ \      |     |      |        / \
    #   #   x1  x2  x1  α    X   α      β     x1     b      x1   x1
                        |    |   |     / \          / \
                        x2   x1  x2  x1   x2      x1   x2

^
X  → a , serait une erreur car la forêt engendrée serait alors
|    |
x1   x1
```

$\{b^p.\beta(a^i\, \#, \alpha^n\, \#) \mid p \in \mathbb{N}^+, n \in \mathbb{N}, i \leq n\} \subsetneq F(GIV1)$.

Les transformations que l'on devra utiliser pour obtenir une grammaire stricte seront moins simples, elles consisteront en fait à échanger la non strictitude contre la non complétude. Dans le cas de la grammaire GIV1 toute dérivation descendante doit être de la forme :

```
X  D=>  Y   D*=>    Y      D=>    Z     D*=> b^p .   β
 o     / \         / \            |                 / \
      #   #     X^n   α^n         β              X^n   α^n
                 |     |         / \              |     |
                 #     #      X^n   α^n           #     #
                               |     |
                               #     #
```

Puis chacune des 2^p occurrences de $\beta(X^n\ \#,\ \alpha^n\ \#)$ sera dérivée séparément et une solution triviale pour échanger la non strictitude par une non complétude est dans ce cas, GIV3 :

```
X  →  Y   ;    Y    →   Y    +    Z    ; Z  →   Z    +    b    ;
 o   / \      / \      / \        |      |      |        / \
    #   #   x1   x2   X   α       β      x1     b      x1   x1
                      |   |      / \           / \
                      x1  x2   x1   x2       x1   x1

X  → X̂  + # ; X̂  → a .
|    |        |    |
x1   x1       x1   x1
```

Bien entendu ce procédé ne pourrait plus être utilisé si # était remplacé par quelque chose de plus compliqué comme dans GIV4, obtenue en remplaçant

```
X  →  Y   par X  →  Y   +  Y    dans GIV1.
 o   / \       o   / \    / \
    #   #         0   #  1   #
```

Ici il faudra conserver en paramètre ce qui suit la chaine des X qui peuvent s'effacer. On pourra construire GIV5 :

```
X  →  Y   +  Y   ;   Y      →     Y       +    Z     ; Z  →   Z    +    b
 o   /|\    /|\     /|\         / | \          |       |      |        / \
    0 # 0  1 # 1  x1 x2 x3     X  α   x3       B       x1     b      x1   x1
                              / \ |           /|\            / \
                            x1 x3 x2        x1 x2 x3       x1   x1
```

```
  X    → X̂  + X̂  ; X̂  → a  ;   B     →   β    +   β
 / \     |    |     |    |     /|\        / \       / \
x1  x2   x1   x2    x1   x1   x1 x2 x3   x1   x2   x3   x2
```

on vérifiera que F(GIV5) = F(GIV4).

En fait ici les effacements sont tous regroupés en bout de branche et réalisés par non complétude. Malheureusement dans l'exemple suivant ce ne sera pas possible.

Grammaire GIV6 :

```
Xo →  Y   ;   Y   →   Y    +   Y    +   Z    ; Z →    Z    +   b    ;
     / \     / \     / \      / \       |      |      |       / \
    #   #   x1  x2  A   a    B   α      β      x1     b      x1  x1
                    |   |    |   |     / \           / \
                    x1  x2   x1  x2   x1  x2        x1  x1

A  → x1 + Â  ; B  → x1 + B̂  ; Â  → a  ; B̂  → α  ;
|         |    |         |    |    |    |    |
x1        x1   x1        x1   x1   x1   x1   x1
```

Si l'on pose pour tout $w \in \{a, \alpha\}^*$,

$D(w) = \{u = u_1u_2 \ldots u_p \mid \exists\, v_o, v_1, \ldots v_p$ tels que $v_o\, u_1\, v_1\, u_2 \ldots u_p\, v_p = w\}$,

on a

$F(GIV6) = \{\tilde{b}^p.[\beta(\omega_1\#, w\#), \ldots \beta(\omega_q\#, w\#)] \mid p \in \mathbb{N}^+,\ q = 2^p,\ w \in \{a, \alpha\}^*$ et $\omega_1, \ldots \omega_p \in D(w)\}$.

Posons $\bar{w}$ l'arbre obtenu en substituant A à a et B à α dans $w \in \{a, \alpha\}^*$, toute dérivation descendante dans GIV6 commence par X_o D=> $Z.\beta(\bar{w}\#, w\#)$ et on a $F(GIV6, \bar{w}) = D(w)$. Une solution consisterait à trouver et à construire un arbre t, image de $\bar{w}$, depuis lequel on pourrait obtenir D(w) en appliquant uniquement des règles strictes. Ceci n'est pas possible car $x_1 \in D(w)$ et si $|w| > 0$ alors $|t| > 0$. Par contre nous pouvons trouver 3 arbres t_o, t_A, t_B tels que $t_o = x_1$, t_A engendre tous les arbres $aw' \in D(w)$ et t_B ceux de la forme $\alpha\, w'' \in D(w)$. Ceci est évident si $|w| = 0$. Supposons que ce soit vrai pour $|w| = \ell$ et posons $\omega = a.w$. Si nous disposons d'un non terminal $\bar{A}$ d'arité 3 tel que $\bar{A}(x_1, x_2, x_3)$ engendre {a , a , a }, il est facile de

```
                                            {a  , a  , a },
                                             |    |    |
                                             x1   x2   x3
```

voir que $\bar{A}(t_o, t_A, t_B)$ engendre tous les arbres $a.w' \in D(a.w)$ donc à ω on pourra faire correspondre $t'_o = t_o$, $t'_A = \bar{A}(t_o, t_A, t_B)$ et $t'_B = t_B$. De même si $\omega = \alpha.w$ il suffira d'introduire le non terminal $\bar{B}$ pour obtenir $t'_o = t_o$, $t'_A = t_A$ et $t'_B = \bar{B}(t_o, t_A, t_B)$. On pourra alors remplacer $\beta(\bar{w}\#, w\#)$ par $C(t_o\#, t_A\#, t_B\#, w\#)$ avec $C(x_1, x_2, x_3, x_4)$ engendrant l'un des arbres

```
  β   ,   β   ,   β   .
 / \     / \     / \
x1  x4  x2  x4  x3  x4
```

Ces remarques conduisent à la grammaire GIV7 :

```
Xo →   Y    ;       Y        →        Y          +        Y              +       Z
     // \\        // \\              /  / \  \           / / \   \                 |
    # # # #     x1 x2 x3 x4        x1   Ā  x3  a       x1 x2  B̄     α             C
                                       /|\     |              /|\    |           // \\
                                     x1 x2 x3  x4           x1 x2 x3 x4       x1 x2 x3 x4

Z  →   Z    +    b    ;      C        →    β    +    β    +    β    ;
|      |        / \        // \\          / \       / \       / \
x1     b      x1   x1    x1 x2 x3 x4    x1   x4   x2   x4   x3   x4
      / \
    x1   x1

                ^     ^     ^                       ^    ^    ^     ^        ^
    Ā      →    A  +  A  +  A  ;     B̄        →     B  + B  + B  ;  A  → a ;  B  → α
   /|\          |     |     |       /|\             |    |    |     |    |    |    |
 x1 x2 x3       x1    x2    x3    x1 x2 x3          x1   x2   x3    x1   x1   x1   x1
```

Cette grammaire est stricte et engendre la même forêt que GIV6. En fait au lieu de construire $\beta(\bar{w}\#, w\#)$ on obtient ici $X_o \overset{*}{\underset{D}{\Rightarrow}} Z.C(t_o\#, t_A\#, t_B\#, w\#)$ et les branches de $C(t_o\#, t_A\#, t_B\#, x_1)$ sont l'image de l'ensemble $\{\beta(v\#, x_1 \mid v \in D(w)\}$. Nous avons conservé dans GIV7 les non terminaux $\hat{A}$ et $\hat{B}$ pour bien montrer que la construction demeure valable, que les forêts qu'ils engendrent soient linéaires ou non.

La construction qui suit est une formalisation de ce qui vient d'être vu. La grammaire qu'elle engendrera sera un peu plus compliquée car, alors que dans GIV7 on n'a multiplié les variables que quand c'était nécessaire, la construction multipliera systèmatiquement toutes les arités par $\mathrm{Card}(E_G)+1$.

Ceci peut être évité en reconnaissant d'abord les variables de chaque terminal ou non terminal auquel pourront être substitués des arbres ayant pour racine des éléments de E_G. Dans l'état actuel de la construction il sera généralement aisé de simplifier son résultat.

Construction IV,1

Etant donnée la grammaire $G = \langle V, \Sigma, X_o, R\rangle$ à non strictitude monadique et ayant les propriétés des lemmes IV7 et IV8, posons $z = \mathrm{Card}(E_G)$ et introduisons un nouveau symbole λ qui sera utilisé seulement pour certaines notations. On écrira alors $\bar{E}_G = E_G \cup \{\lambda\}$.

Donnons nous une numérotation $\bar{\phi}$ de $\bar{E}_G$ telle que $\bar{\phi}(0) = \lambda$ et que $\bar{\phi}$ soit une bijection de $[z]$ dans E_G.
Notons f la bijection de $\mathbb{N}^+ \times \bar{E}_G$ dans $\mathbb{N}^+$ définie par $f(j, X) = (j-1)(z+1) + 1 + \bar{\phi}^{-1}(X) = i$. On pourra noter ϕ et Ψ les fonctions projection de l'image inverse de i de sorte que $i = f(\Psi(i), \phi(i))$. Le rôle de ces fonctions est d'indiquer les relations existant entre les nouvelles variables qui seront ajoutées aux symboles, les éléments de $\bar{E}_G$ et les anciennes variables.

Pour tout $Y \in \Sigma \cup V \setminus E_G$, d'arité n, introduisons le nouveau non terminal $\bar{Y}$ d'arité $m = n(z+1)$. La restriction f_Y de f à $[n] \times \bar{E}_G$ est une bijection dans $[m]$ qui associe à toute variable x_i de $\bar{Y}$ une variable x_j de Y et une marque X de $\bar{E}_G$ tels que $i = f(j, X)$, on a donc $j = \Psi(i)$ et $X = \phi(i)$. C'est à dire que pour toute variable de Y, tout choix de marque détermine le choix d'une variable de $\bar{Y}$. Ces combinaisons de choix pourront être exprimées par l'ensemble des torsions linéaires ordonnées
$\Theta_Y = \{\langle m ; x_{f(1,X_1)}, \ldots x_{f(n, X_{n_o})}\rangle \mid X_1, \ldots, X_n \in \bar{E}_G\} \subset \Theta^n_m$.
Posons $W = \{\bar{Y} \mid Y \in \Sigma \cup V \setminus E_G\}$ et $\bar{T}(\Sigma \cup V) = \{t \in T(\Sigma \cup V) \mid \mathrm{racine}(t) \notin E_G\}$

Définissons la substitution descendante σ de $T(W \cup \Sigma)$ dans $T(\Sigma \cup V \setminus E_G)$ par : $\forall \bar{X} \in W$, $\sigma(\bar{X}) = \{X.\theta \mid \theta \in \Theta_X\}$ et

$\forall \alpha \in \Sigma$, $\sigma(\alpha) = \{\alpha\}$.

On appellera $R_E = \{(X \to t) \in R \mid X \in E_G\}$ et on exprimera l'effet de l'application de ces règles à l'aide de la substitution descendante σ_E de $T(\Sigma \cup V)$ dans $T(\Sigma \cup V \setminus E_G)$ définie par :

$\forall Y \in \Sigma \cup V \setminus E_G$, $\sigma(Y) = \{Y\}$ et $\forall X \in E_G$, $\sigma(X) = \{x_1, \hat{X}\}$.

A tout arbre $t \in T(\Sigma \cup V)^1_n$ nous allons associer z+1 arbres de $T(W)^1_n$ notés $H_{\bar{\phi}(0)}(t)$, $H_{\bar{\phi}(1)}(t)$, ... $H_{\bar{\phi}(z)}(t)$ et définis récursivement par :

$\forall x_i \in X$ et $\forall X \in \bar{E}_G$, $H_X(x_i) = x_{f(i,X)}$

Si $t = Y(t')$ avec $Y \in E_G$ alors $H_Y(Y.t') = \hat{\bar{Y}}(H_{\bar{\phi}(0)}(t'), .. H_{\bar{\phi}(z)}(t'))$ et pour tout $Z \in \bar{E}_G \setminus Y$, $H_Z(Y.t') = H_Z(t')$.

Si $t = X(t_1, .., t_p)$ avec $X \notin E_G$ et $d(X) = p$ alors pour tout $Z \in \bar{E}_G$ $H_Z(t) = H(t) = \bar{X}(\tau_1, .. \tau_q)$ avec $q = (z+1)p$ et pour tout $i \in [q]$, $\tau_i = H_{\phi(i)}(t_{\psi(i)})$.

Pour tout $t \in \mathring{T}(\Sigma \cup V)^1_n$ les z+1 arbres sont identiques et seront notés par $H(t)$. En particulier pour tout élément α de $\Sigma \cup V \setminus E_G$ on a $H(\alpha) = \bar{\alpha}$.

Nous venons en fait de définir une application H qui à tout arbre t de $T(\Sigma \cup V)^1_n$ associe un vecteur de $T(W)^{z+1}_{n(z+1)}$, c'est un (z+1)-morphisme de $T(\Sigma \cup V)$ dans $T(W)$ selon la définition donnée dans [AD-m1].

Posons alors $G' = \langle W', \bar{\Sigma}, \bar{X}_0, R' \rangle$ avec

$\bar{\Sigma} = \{\bar{\alpha} \mid \alpha \in \Sigma\} \subset W$, $W' = \{\bar{X} \mid X \in V \setminus E_G\} = W \setminus \bar{\Sigma}$ et

$R' = \{\bar{X} \to H(t) \mid (X \to t) \in R$ et $X \in V \setminus E\}$ ce qui est valide car d'après le lemme IV8, si $X \in V \setminus E$ et $(X \to t) \in R$ alors $t \in \mathring{T}(\Sigma \cup V)$.

Il est clair que G' est stricte. Afin d'obtenir le même alphabet terminal que pour G, définissons $G'' = \langle W, \Sigma, \bar{X}_0, R' \cup R'' \rangle$ avec

$R'' = \{(\bar{\alpha} \to \alpha.\theta \mid \bar{\alpha} \in \bar{\Sigma}$ et $\theta \in \Theta_\alpha\}$.

G'' est encore stricte, nous allons montrer que $F(G) = F(G'')$.

<u>**Lemme IV,9**</u> Posons $t = \alpha_1.\alpha_2. \dots .\alpha_r.\tau$ avec $\alpha_1, \alpha_2, .. \alpha_r \in E_G$ et $\tau \in \overset{\circ}{T}(\Sigma \cup V)^1$.

P : Pour tout $t_i = \alpha_{i+1}.\alpha_{i+2}. \dots .\alpha_r.\tau$ et pour tout $\alpha \in \bar{E}_G$ on a $\sigma(S^n_{G'}(H_\alpha(t_i))) \subset \sigma(S^n_{G'}(H_\alpha(t))) \cup \sigma(S^n_{G'}(H_\lambda(t)))$.

<u>Preuve</u>

Montrons le par récurrence sur r.

Si $r = 0$ on a $H_\alpha(t_i) = H_\alpha(t) = H_\alpha(\tau) = H_\lambda(t)$.

Supposons la propriété P vraie si $r < N$ et prenons $r = N > 0$. $t = \alpha_1.t_1$.

Si $\alpha \neq \alpha_1$ $H_\alpha(t) = H_\alpha(\alpha_1.t_1) = H_\alpha(t_1)$ donc P est vrai pour α et t_1

Si $\alpha = \alpha_1$: ou bien il n'existe pas $i \in [r]$ avec racine $(t_i) = \alpha$ et alors $H_\alpha(t_1) = H_\alpha(\tau) = H_\lambda(t)$, et P est vraie, ou bien posons $i \in [r]$ le plus petit tel que racine $(t_i) = \alpha$.

$$\sigma(S^n_{G'}(H_\alpha(t))) = \sigma(S^n_{G'}(\bar{\hat{\alpha}}(H_\lambda(t_1), H_{\bar{\phi}(1)}(t_1), \dots H_\alpha(t_1), \dots H_{\bar{\phi}(z)}(t_1))))$$

et puisque σ est linéaire

$$\sigma(S^n_{G'}(H_\alpha(t))) = \sigma(S^n_{G'}(\bar{\hat{\alpha}})) \circledast [\sigma(S^n_{G'}(H_\lambda(t_1))), \sigma(S^n_{G'}(H_{\bar{\phi}(1)}(t_1))), \dots \sigma(S^n_{G'}(H_\alpha(t_1))), \dots \sigma(S^n_{G'}(H_{\bar{\phi}(z)}(t_1)))].$$

on a $H_\alpha(t_1) = H_\alpha(t_i)$, $t_i = \alpha.t_{i+1}$

$$\sigma(S^n_{G'}(H_\alpha(t_1))) = \sigma(S^n_{G'}(\bar{\hat{\alpha}})) \circledast [\sigma(S^n_{G'}(H_\lambda(t_{i+1}))), \sigma(S^n_{G'}(H_{\bar{\phi}(1)}(t_{i+1}))), \dots \sigma(S^n_{G'}(H_\alpha(t_{i+1}))), \dots \sigma(S^n_{G'}(H_{\bar{\phi}(z)}(t_{i+1})))].$$

En fait si l'on pose $\theta = \langle 1 \,;\, x_1, \dots x_1\rangle \in \Theta_1^{z+1}$

on a $\sigma(S^n_{G'}(\bar{\hat{\alpha}})) = \{\sigma(S^n_{G'}(\bar{\hat{\alpha}})).\theta\} \circledast \{x_1, \dots x_{z+1}\}$.

$H_\lambda(t_1) = H_\lambda(t_{i+1}) = H_\lambda(t)$ et par hypothèse de récurrence pour tout $X \in \bar{E}_G$, $\sigma(S^n_{G'}(H_X(t_{i+1}))) \subset \sigma(S^n_{G'}(H_X(t_1))) \cup \sigma(S^n_{G'}(H_\lambda(t_1)))$ donc s'il existe $i \in [r]$ avec racine $(t_i) = \alpha$ on a $\sigma(S^n_{G'}(H_\alpha(t_1))) \subset \sigma(S^n_{G'}(H_\alpha(t)))$ et P est vraie.

$\square$

Pour la grammaire G définie ici, donnons nous la substitution descendante $\hat{S}^n_G$ légèrement différente de S^n_G, définie par :

$\forall\ n \in \mathbb{N},\ \forall\ \alpha \in \Sigma,\ \hat{S}^n_G(\alpha) = S^n_G(\alpha) = \{\alpha\}$

$\forall\ X \in V \setminus E_G,\ \hat{S}^o_G(X) = S^o_G(X) = \emptyset$ et si $X \in E_G,\ \hat{S}^o_G(X) = \{x_1\}$

$\forall\ n \in \mathbb{N}^+$, si $X \in V \setminus E_G$ alors $\hat{S}^n_G(X) = \{\tau \in T(\Sigma) \mid \exists\ (X \to t) \in R$ et $\tau \in \hat{S}^{n-1}_G(t)\}$

Si $X \in E_G$ alors $\hat{S}^n_G(X) = \{x_1\} \cup \hat{S}^n_G(\hat{X})$

On peut dire que si on appelle l'indice n de S^n_G la profondeur de dérivation, l'indice n de $\hat{S}^n_G$ ne compte pas l'application de règles de R_E dans cette nouvelle indication de profondeur. De la même façon que pour S^n_G on montrerait que $\hat{S}^n_G(X)$ a pour limite $F(G, X)$.

<u>Lemme IV,10</u> Pour tout $\alpha \in E_G$ et tout $t \in T(\Sigma \cup V)$ on a

$$\bigcup_{X\in\bar{E}_G} \sigma(S^n_{G'}(H_X(\alpha.t))) = \Big(\bigcup_{X\in\bar{E}_G} \sigma(S^n_{G'}(H_X(t)))\Big) \cup \sigma(S^n_{G'}(H_\alpha(\alpha.t))).$$

<u>Preuve</u>

Par définition de H_X on a

$\bigcup_{X\in\bar{E}_G} H_X(\alpha.t) = \Big(\bigcup_{X\in\bar{E}_G\setminus\alpha} H_X(t)\Big) \cup \{H_\alpha(\alpha.t)\}$, ce qui nous donne l'inclusion dans un sens. D'autre part le lemme IV,9 nous montre que

$\sigma(S^n_{G'}(H_\alpha(t))) \subset \sigma(S^n_{G'}(H_\alpha(\alpha.t))) \cup \sigma(S^n_{G'}(H_\lambda(t)))$ et $\lambda \in \bar{E}_G \setminus \alpha$.

$\square$

Pour tout $t \in T(\Sigma)$ posons $\sigma_H(t) = \bigcup_{X\in\bar{E}_G} \sigma(H_X(t))$.

$\sigma_H([t_1,\dots t_p]) = [\sigma_H(t_1),\dots \sigma_H(t_p)] = \{[t'_1,\dots t'_p] \mid \forall\ i\in[p],\ t'_i \in \sigma_H(t_i)\}$

et $\sigma_H(x_i) = \{x_{f(i,\bar{\phi}(o))},\dots x_{f(i,\bar{\phi}(z))}\} = \{x_j,\dots x_{j+z} \mid j = (i-1)(z+1)+1\}$

<u>Lemme IV,11</u> $\sigma_H(t.u) = \{t\} \circledS \sigma_H(u)$.

<u>Preuve</u>

Par induction sur $|t|$. Posons $t \in T(\Sigma)^1_p$ et $u = [u_1, .. u_p] \in T(\Sigma)^p_q$. Si $|t| = 0$, $t = x_i$, $t.u = u_i$ donc $\sigma_H(t.u) = \sigma_H(u_i)$ et $\{t\} \circledS \sigma_H([u_1, .. u_p]) =$ $\{x_i\} \circledS [\sigma_H(u_1), ... \sigma_H(u_p)] = \sigma_H(u_i)$. Le lemme est donc vrai si $|t| = 0$; supposons le vrai si $|t| < \ell$ et prenons $t = \alpha(t_1,.. t_q)$ avec $\alpha \in \Sigma_q$ et $\forall\, i \in [q]$, $t_i \in T(\Sigma)^1_p$. $\sigma_H(t.u) = \sigma_H(\alpha(t_1.u,.. t_q.u))$

$$= \bigcup_{X \in \bar{E}_G} \sigma(H_X(\alpha(t_1.u,.. t_q.u)))$$

Pour tout $X \in \bar{E}_G$

$H_X(\alpha(t_1.u,.. t_q.u)) = \bar{\alpha}(H_{\bar{\phi}(0)}(t_1.u), .. H_{\bar{\phi}(z)}(t_1.u), H_{\bar{\phi}(0)}(t_2.u),..H_{\bar{\phi}(z)}(t_q.u))$

$\sigma(H_X(\alpha(t_1.u, .. t_q.u))) = \{\alpha\} \circledS [\sigma_H(x_1),.. \sigma_H(x_q)] \circledS [\sigma(H_{\bar{\phi}o}(t_1.u)),...$
$\sigma(H_{\bar{\phi}(z)}(t_1.u)), \sigma(H_{\bar{\phi}(0)}(t_2.u)), ... \sigma(H_{\phi(z)}(t_q.u))]$ de
$\sigma_H(x_i) = \{x_j, ... x_{j+z} \mid j = (i-1)(z+1) +1\}$ on déduit
$\sigma(H_X(\alpha(t_1.u,.. t_q.u))) = \{\alpha\} \circledS [\sigma_H(t_1.u),.. \sigma_H(t_q.)]$ comme
$\forall\, X, Y \in \bar{E}_G$, $H_X(\alpha(t_1.u, .. t_q.u)) = H_Y(\alpha(t_1.u,.. t_q.u))$ on a
$\sigma_H(t) = \{\alpha\} \circledS [\sigma_H(t_1.u), .. \sigma_H(t_q.u)]$ par hypothèse de récurrence
$\sigma_H(t_i.u) = \{t_i\} \circledS \sigma_H(u)$ donc
$\sigma_H(t) = \{\alpha\} \circledS [\{t_1\} \circledS \sigma_H(u),.. \{t_q\} \circledS \sigma_H(u)]$
$= \{\alpha(t_1,.. t_q)\} \circledS \sigma_H(u) = \{t\} \circledS \sigma_H(u)$.

□

<u>Lemme IV,12</u> Pour tout $Y \in V \setminus E_G$ et tout $n \in \mathbb{N}$ on a $\sigma_H(\hat{S}^n_G(Y))=\sigma(S^n_{G'}(\bar{Y}))$.

<u>Preuve</u>

Par récurrence sur n.

Le lemme est trivialement vrai par vacuité quand n = 0. Supposons le vrai pour tout n < N et prenons n = N > 0. Il suffit de montrer que pour toute règle $(Y \to t) \in R$ avec $Y \in V \setminus E_G$ on a $\sigma_H(\hat{S}_G^{n-1}(t)) = \sigma(S_{G'}^{n-1}(H(t)))$.

D'après les lemmes IV,5 et IV,8 on a $t = \alpha(t_1,.. t_q) \in T(\Sigma \cup V)_p^1$ avec $\alpha \in \Sigma \cup V \setminus E_G$ et pour tout $i \in [q]$, $|t_i| \leq 1$

$H(t) = \bar{\alpha}(H_{\bar{\phi}(0)}(t_1),.. H_{\bar{\phi}(z)}(t_1), H_{\bar{\phi}(0)}(t_2),.. H_{\bar{\phi}(z)}(t_q))$

et en posant $r = q(z+1)$

$H(t) = \bar{\alpha}(H_{\phi(1)}(t_{\Psi(1)}), \ldots H_{\phi(r)}(t_{\Psi(r)}))$.

$S_{G'}^{n-1}(H(t)) = S_{G'}^{n-1}(\bar{\alpha}) \circledast [S_{G'}^{n-1}(H_{\phi(1)}(t_{\Psi(1)})), .. S_{G'}^{n-1}(H_{\phi(r)}(t_{\Psi(r)}))]$.

σ est linéaire donc d'après le lemme 14

$\sigma(S_{G'}^{n-1}(H(t))) = \sigma(S_{G'}^{n-1}(\bar{\alpha})) \circledast [\sigma(S_{G'}^{n-1}(H_{\phi(1)}(t_{\ (1)}))),.. \sigma(S_{G'}^{n-1}(H_{\phi(r)}(t_{\Psi(r)})))]$.

Et par hypothèse de récurrence sur n, $\sigma(S_{G'}^{n-1}(\bar{\alpha})) = \sigma_H(\hat{S}_G^n(\alpha))$. On a alors $\tau \in \sigma(S_{G'}^{n-1}(H(t)))$ ssi $\tau = \tilde{\nu}.[\tau_1,.. \tau_m]$ avec $\tilde{\nu}.\theta \in \hat{S}_G^{n-1}(\alpha)$ et $\tilde{\nu}.\theta' \in \sigma_H(\hat{S}_G^{n-1}(\alpha))$ avec $\theta' \in \sigma_H(\theta)$ d'après le lemme IV,11.

On a alors pour tout $i \in [m]$, $x_i.\theta' \in \{x_{f(\theta(i),\ X)} \mid X \in \bar{E}_G\}$ d'où

$\tau_i \in \bigcup_{X \in \bar{E}_G} \sigma(S_{G'}^{n-1}(H_X(t_{\theta(i)})))$.

$\sigma_H(\hat{S}_G^{n-1}(t)) = \sigma_H(\hat{S}_G^{n-1}(\alpha) \circledast [\hat{S}_G^{n-1}(t_1), .. \hat{S}_G^{n-1}(t_q)])$ donc

$\tau \in \sigma_H(\hat{S}_G^{n-1}(t))$ ssi $\tau = \tilde{\nu}.[\tau_1,\ldots \tau_m]$ avec $\tilde{\nu}.\theta \in \hat{S}_G^{n-1}(\alpha)$ et pour tout $i \in [m]$, $\tau_i \in \sigma_H(\hat{S}_G^{n-1}(t_{\theta(i)}))$. Donc on aura $\sigma_H(\hat{S}_G^{n-1}(t)) = \sigma(S_{G'}^{n-1}(H(t))$ si pour tout $i \in [q]$ on a $\sigma_H(\hat{S}_G^{n-1}(t_i)) = \bigcup_{X \in \bar{E}_G} \sigma(S_{G'}^{n-1}(H_X(t_i)))$. Puisque $|t_i| \leq 1$ nous distinguerons 3 cas :

A) $t_i = x_j$ alors $\sigma_H(\hat{S}_G^{n-1}(x_j)) = \sigma_H(x_j) = \bigcup_{X \in \bar{E}_G} H_X(x_j)$ et $\bigcup_{X \in \bar{E}_G} \sigma(S_{G'}^{n-1}(H_X(x_j))) = \bigcup_{X \in \bar{E}_G} H_X(x_j)$. Donc si $t = \alpha.\theta$, avec $\theta \in \Theta_{d(Y)}^{d(\alpha)}$ et $\alpha \in \Sigma \cup V \setminus E_G$ on a $\sigma_H(\hat{S}_G^{n-1}(t)) = \sigma(S_{G'}^{n-1}(H(t))$.

B) $t_i = \gamma.\theta$ avec $\gamma \in \Sigma \cup V \setminus E_G$ et $\theta \in \Theta_{d(Y)}^{d(\gamma)}$, cf. résultat de A.

C) $t_i = Z(x_j)$ avec $Z \in E_G$. D'après le lemme IV.10,

$$\bigcup_{X\in\bar{E}_G} \sigma(S^{n-1}_{G'}(H_X(Z(x_j))))=(\bigcup_{X\in\bar{E}_G} \sigma(S^{n-1}_{G'}(H_X(x_j)))) \cup \sigma(S^{n-1}_{G'}(H_Z(Z(x_j)))).$$

$$= \sigma_H(x_j) \cup \sigma(S^{n-1}_{G'}(H_Z(Z(x_j)))).$$

On a $H_Z(Z(x_j)) = H_Z(\hat{Z}(x_j))$ donc

$$\bigcup_{X\in\bar{E}_G} \sigma(S^{n-1}_{G'}(H_X(Z(x_j)))) = \sigma_H(x_j) \cup \sigma(S^{n-1}_{G'}(H_Z(\hat{Z}(x_j))))$$

$$= \sigma_H(x_j) \cup (\bigcup_{X\in\bar{E}_G} \sigma(S^{n-1}_{G'}(H_X(\hat{Z}(x_j)))))$$

$$= \sigma_H(x_j) \cup \sigma_H(\hat{S}^{n-1}_G(\hat{Z}(x_j)))$$

$$= \sigma_H(\{x_j\} \cup \hat{S}^{n-1}_G(\hat{Z}(x_j)))$$

$$= \sigma_H(\hat{S}^{n-1}_G(Z(x_j))) \text{ par définition de } \hat{S}.$$

□

<u>Corollaire IV,13</u> $F(G) = \sigma(F(G'))$

<u>Preuve</u>

$\sigma_H(\hat{S}^n_G(X_o)) = \hat{S}^n_G(X_o)$ car $var(X_o) = \emptyset$ et par définition de $\hat{S}^n_G$ on a $\lim_{n\to\infty} \hat{S}^n_G(X_o) = F(G, X_o)$, d'autre part $F(G') = \lim_{n\to\infty} S^n_{G'}(\bar{X}_o)$ et le lemme IV,12 nous montre que pour tout n, $\hat{S}^n_G(X_o) = \sigma(S^n_{G'}(\bar{X}_o))$.

□

<u>Lemme IV,14</u> $F(G'') = \sigma(F(G'))$.

<u>Preuve</u>

Les règles de R" sont toutes linéaires et terminales. Donc d'après le lemme 28 on peut supposer que toutes les règles de R" seront appliquées après celles de R'. Toute dérivation terminale dans G" s'écrit alors $\bar{X}_o \overset{*}{\underset{R'}{\Rightarrow}} t' \overset{*}{\underset{R''}{\Rightarrow}} t''$, avec $t' \in F(G') \subset T(\bar{\Sigma})$ et $t'' \in T(\Sigma)$. Par définition de R" on a $S^n_{G''}(\bar{\alpha}) = \sigma(\bar{\alpha})$

$\forall\ \bar{\alpha} \in \bar{\Sigma}$ et $n \geq 1$, donc $F(G'', t') = \sigma(t')$. On a alors $F(G'') = \sigma(F(G'))$.

□

Corollaire IV,15 $F(G'') = F(G)$.

Preuve

Par le lemme IV,14, $F(G'') = \sigma(F(G'))$ et par le corollaire IV,13, $F(G) = \sigma(F(G'))$.

Proposition IV,16 La non strictitude monadique est réductible et pour toute grammaire complète G, il existe une grammaire stricte G' telle que $F(G) = F(G')$.

Preuve

Le corollaire IV,15 montre que la construction IV,1 réduit la non strictitude monadique. Si G est complète et non stricte alors toute règle non stricte est de la forme $X \rightarrow x_1$ avec $d(X) = 1$, donc G est à non strictitude monadique.

□

Remarque

En général G' n'est plus complète et nous montrerons qu'en général on ne peut trouver G' à la fois complète et stricte.

2) REDUCTION DE LA NON COMPLETUDE ET DE LA NON STRICTITUDE DANS LES DERIVATIONS INITIALES

Dans de nombreuses démonstrations, on est amené à examiner dans le détail certaines dérivations descendantes, et en particulier le début de ces dérivations depuis l'axiome. Les arbres obtenus successivement ont alors pour racine un non terminal auxquel s'appliquent les règles. L'utilisation de règles non strictes ou non complètes complique considérablement l'examen de ces dérivations, en ce sens que l'on doit tenir compte du développement de sous arbres qui, disparaissant au cours des dérivations suivantes, ne sont d'aucune utilité. Arnold et Dauchet avaient déjà évité ces inconvénients en montrant dans le cas des grammaires sous forme normale que pour toute dérivation initiale produisant un arbre t, on pouvait trouver une dérivation initiale produisant le même arbre t sans utiliser de règle non stricte [AD-ϕ-1]. Nous allons montrer que l'on peut transformer toute grammaire en conservant la forêt engendrée et certaines propriétés, de telle façon que toute dérivation initiale depuis l'axiome utilise des règles complètes et strictes, nous verrons aussi que l'on peut séparer en deux classes disjointes les non terminaux pouvant apparaitre en sommet dans une dérivation et ceux pouvant apparaître ailleurs.

On dira que la dérivation $X \underset{G}{D\Rightarrow} t_1 \underset{G}{D\Rightarrow} \ldots \underset{G}{D\Rightarrow} t_n \underset{G}{D\Rightarrow} t$ est **initiale** si pour tout $i \in [n]$, racine (t_i) est non terminal, elle sera **initiale maximale** si de plus racine (t) est un symbole terminal, elle est alors maximale en ce sens qu'elle peut se prolonger en une dérivation initiale plus longue. Dans la majorité des cas X sera l'axiome de la grammaire.

Si $X.\vec{w}$ D=> $\tilde{u}.[u_1, \ldots u_i, \ldots u_p]$. $\vec{w}$ $D\overset{*}{=}>$ $u_i.\vec{w}$ avec $|u_i| > 0$, c'est à dire que $\tilde{u}$ $D\overset{*}{=}>$ x_i, nous noterons $X.\vec{w}$ >=> $u_i.\vec{w}$. Cette notation permettra d'exhiber les moments importants dans une dérivation sans avoir à se perdre dans les détails de dérivation qui n'ont pour résultat que d'effacer le sommet : $\tilde{u}$ $D\overset{*}{=}>$ x_i. Si on a t_o >=> t_1 >=> t_2 >=> ... >=> t_n on écrira $t \overset{n}{>=>} t_n$ ou $t \overset{*}{>=>} t_n$ et comme pour les autres dérivations on écrira $t_o \overset{*}{>=>} t_o$.

Lemme IV,17 Toute dérivation initiale est la composition de dérivations >=>.

Preuve

Soit X_o $D\overset{*}{=}>$ t une dérivation initiale de longueur ℓ dans G = <V, Σ, X_o, R> On a $|t| > 0$ et la dérivation s'écrit X_o D=> $\tilde{u}.[u_1, ..u_i..u_p]$ $D\overset{*}{=}>$ t avec $\tilde{u}$ $D\overset{*}{=}>$ x_i, $u_i = X.\vec{w}$, X $D\overset{*}{=}>$ u avec $|u| > 0$ et $t = u.\vec{w}$. Donc X_o >=> $X.\vec{w}$ $D\overset{*}{=}>$ $u.\vec{w}$ et X $D\overset{*}{=}>$ u est une dérivation initiale de longueur inférieure à ℓ. De plus pour toute dérivation de longueur nulle t $D\overset{*}{=}>$ t on a $t \overset{*}{>=>} t$ devra par induction sur ℓ, $X_o \overset{*}{>=>} t$.

□

Lemme IV,18 Pour toute grammaire G = <V, Σ, X_o, R> on peut construire l'ensemble $\hat{V}$ des non terminaux pouvant se dériver dans une dérivation initiale depuis l'axiome.

$\hat{V} = \{X \in V \mid \exists\, \vec{w} \in T(\Sigma \cup V)$ tel que $X_o \underset{G}{D\overset{*}{=}>} X.\vec{w}\}$.

Construction

Posons $\hat{V}_o = \{X_o\}$ et pour tout $i > 0$

$\hat{V}_i = \hat{V}_{i-1} \cup \{X \in V \mid \exists\, Y \in \hat{V}_{i-1}$ et $\exists\, \vec{w} \in T(\Sigma \cup V)$ tels que $Y >=> X.\vec{w}\}$

la construction s'arrête car $\hat{V}_i \subset V$ et si $\hat{V}_i = \hat{V}_{i+1}$ alors $\forall\, k > 0$, $\hat{V}_i = \hat{V}_{i+k}$.

<u>Preuve</u> : Il faut montrer que la construction de $\hat{V}_i$ connaissant $\hat{V}_{i-1}$ est effective et que la limite de la suite $\hat{V}_i$, qui est atteinte au plus tard en Card(V) étapes est $\hat{V}$. V est fini donc $\hat{V}_i$ est fini et il suffit de déterminer si $Y >=> X.\vec{w}$ pour au moins un w. On développe $Y >=> X.\vec{w}$ en $Y \; D=> t = \tilde{u}.[u_1,.. \; u_i,.. \; u_p] \; D\overset{*}{=}>u_i = X.\vec{w}$. Alors $(Y \to t) \in R$ et R est fini, de plus t est de profondeur finie, on peut donc le découper en $\tilde{u}.[u_1,.. \; u_i,.. \; u_p]$ d'un nombre fini de façons différentes et connaissant E_G et z(X) pour tout $X \in E_G$ (lemme IV,1 et corollaire IV,2), on peut déterminer si $\tilde{u}$, qui est de profondeur finie, se dérive en x_i, on peut donc déterminer tous les u_i tels que $t \; D\overset{*}{=}> u_i$, pour tous les t tels que $(Y \to t) \in R$ pour tous les $Y \in \hat{V}_{i-1}$, la construction de $\hat{V}_i$ est donc effective.
Si $X_o \; D\overset{*}{\underset{G}{=}}> X.\vec{w}$ avec $X \in V$ la dérivation est initiale donc $X_o \overset{k}{>=>} X.\vec{w}$ pour k fini, donc $X \in \hat{V}_k \subseteq \hat{V}_{Card(V)}$ donc $\hat{V} \subset \hat{V}_{Card(V)}$.
Réciproquement par définition de $\hat{V}_i$ on a $\hat{V}_i \subset \hat{V}$ donc $\hat{V} = \hat{V}_{Card(V)}$.

□

<u>**Lemme IV,19**</u> Pour toute grammaire $G = <V, \Sigma, X_o, R>$, on peut trouver une grammaire $\bar{G} = <\bar{V}, \Sigma \cup V, \bar{X}_o, \bar{R}>$ avec $\bar{V} \cap (V \cup \Sigma) = \emptyset$ telle que $F(\bar{G}) = \{t \in T(\Sigma \cup V) \mid \exists \; (X_o \; D\overset{*}{\underset{G}{=}}> t)$ dérivation initiale maximale$\}$.

Construction

On pose $\bar{V} = \{\bar{X} \mid X \in \hat{V}$ et $d(\bar{X}) = d(X)\}$ et tous les symboles de $\bar{V}$ sont de nouveaux symboles.

Pour tout $t \in T(\Sigma \cup V)$ avec $t = X.\vec{w}$ on pose

si $X \notin \hat{V}$ alors marq(t) = t

si $X \in \hat{V}$ alors marq(t) = $\bar{X}.\vec{w}$.

$\bar{R} = \{\bar{X} \to \bar{t} \mid X \underset{G}{>=>} t, X \in \hat{V}, \bar{t} = marq(t), |t| > 0\}$

Preuve

La construction de $\bar{R}$ est effective car nous avons vu dans le lemme 18 que l'on pouvait déterminer l'ensemble des t tels que $X \underset{G}{>=>} t$ pour tout $X \in V$ et cet ensemble est fini. Par induction sur la longueur des dérivations, on montre facilement que si $\bar{X}_o \underset{\bar{G}}{\overset{\ell}{D=>}} \bar{t}$ alors il existe $t \in T(\Sigma \cup V)$ tel que $\bar{t} = marq(t)$ et $X_o \underset{G}{\overset{\ell}{>=>}} t$ donc $X_o \underset{G}{\overset{*}{D=>}} t$ est une dérivation initiale. Réciproquement si $X_o \underset{G}{\overset{*}{D=>}} t$ est initiale, $X_o \underset{G}{\overset{*}{>=>}} t$ donc $\bar{X}_o = marq(X_o)$ $\underset{\bar{G}}{\overset{*}{D=>}} \bar{t} = marq(t)$.

Si $X_o \underset{G}{\overset{*}{D=>}} t$ est initiale maximale, $t = \alpha.\vec{w}$ avec $\alpha \in \Sigma$ donc $marq(t) = t$ et $\bar{X}_o \underset{\bar{G}}{\overset{*}{D=>}} t$.

Réciproquement si $\bar{X}_o \underset{\bar{G}}{\overset{*}{D=>}} t$ avec $t \in T(\Sigma \cup V)$, il n'est pas possible que $t = X.\vec{w}$ avec $X \in V$, car alors d'une part on doit avoir $t = marq(t)$ donc $X \notin \hat{V}$ et d'autre part $X_o \underset{\bar{G}}{\overset{*}{D=>}} t$ donc $X \in \hat{V}$. Donc si $\bar{X}_o \underset{\bar{G}}{\overset{*}{D=>}} t$ avec $t \in T(\Sigma \cup V)$, $X_o \underset{G}{\overset{*}{D=>}} t$ est une dérivation initiale maximale.

□

Proposition IV.20

Etant donnée une grammaire $G = \langle V, \Sigma, X_o, R\rangle$ on peut construire une grammaire G' telle que dans toute dérivation initiale depuis l'axiome dans G', les règles appliquées soient strictes et complètes, que $F(G) = F(G')$, que G' soit réduite et que si G est complète ou stricte ou de Greibach, G' le soit aussi.

Construction

On construit d'abord $\bar{G}$ du lemme IV,19.

$\bar{G}$ est corrégulière, donc $F_{IO}(\bar{G}) = F_{OI}(\bar{G})$.

Posons $\bar{G}'$ la grammaire construite depuis $\bar{G}$ par la construction III,1 la proposition III,3 nous dit que $\bar{G}'$ est complète, stricte, IO-réduite donc réduite, corégulière et que $F_{IO}(\bar{G}') = F_{IO}(\bar{G})$. $\bar{G}$ et $\bar{G}'$ étant corégulières on en déduit $F_{OI}(\bar{G}') = F_{OI}(\bar{G})$. Notons $\bar{G}' = <\bar{V}', \Sigma \cup V, \bar{X}'_o, \bar{R}'>$ avec $\bar{V}' \cap (\Sigma \cup V) = \emptyset$. Posons $G'' = <\bar{V}' \cup V, \Sigma, \bar{X}'_o, \bar{R}' \cup R>$ et G' la grammaire réduite issue de G'' conformément au théorème II.15.

Preuve

Pour montrer que G' engendre la même forêt que G, il suffit de remarquer que toute dérivation initiale maximale depuis l'axiome dans G'' est une dérivation terminale dans $\bar{G}'$. Comme $F(\bar{G}') = F(\bar{G})$, que $F(\bar{G}) = \{t \mid X_o \overset{*}{\underset{\bar{G}}{D\Rightarrow}} t$ est initiale maximale$\}$ et que toutes les règles de G sont dans G'', on a $F(G) = F(G'')$. De plus $F(G') = F(G'')$ d'après le théorème II,15 donc $F(G) = F(G')$. Les règles de $\bar{R}'$ sont strictes et complètes par la construction III.1, donc si G est stricte ou complète G'' l'est aussi. Si G est de Greibach haut, toute cette construction est sans objet, car les dérivations initiales maximales depuis l'axiome sont de longueur 1 donc strictes et complètes. Si G est de Greibach bas, $\bar{G}$ l'est aussi, car les règles de $\bar{R}$ sont issues de règles de R dans lesquelles on n'a effacé que des non terminaux. Alors $\bar{G}'$ et G'' sont aussi de Greibach bas d'après la proposition III.3. D'après le théorème II.15, on peut supposer G réduite, $\bar{G}'$ est réduite et seuls peuvent subsister dans G'' des non terminaux inaccessibles depuis l'axiome. Il suffit alors de les retirer et de retirer leurs règles (corollaire II,2) pour obtenir G'. Cette dernière réduction ne modifie aucune régle, donc aucune des propriétés qui nous intéressent.

□

Remarque On aurait pu ajouter la régularité et la linéarité aux propriétés conservées dans la proposition IV.20, mais il est bien évident qu'une grammaire régulière est complète et stricte et le théorème III.8 montre que toute grammaire linéaire peut être supposée complète et stricte.

3) ETUDE DES FORETS ALGEBRIQUES A ARBRES BALANCES

Après avoir proposé un certain nombre de réductions, nous allons montrer que certaines réductions ne sont pas toujours possibles. En particulier nous montrerons qu'il existe des forêts algébriques qui ne peuvent pas être engendrées par des grammaires strictes ou complètes. Ceci étant en contradiction avec ce qui se passe pour les langages, on ne pourra le montrer qu'en utilisant la caractéristique qui différencie les arbres des mots, à savoir leur largeur. Encore faut-il que cette largeur puisse être grande. En effet toute forêt ne contenant que des arbres de largeur bornée n'est pas essentiellement différente d'un langage, de même qu'un langage ne comportant que des mots de longueur bornée n'est pas essentiellement différent d'un alphabet.

La largeur des arbres est la longueur de leur feuillage. Il serait cependant insuffisant de considérer des arbres de feuillage arbitrairement long car l'ensemble de ces feuillages, définis sur un alphabet fini, ne serait pas plus complexe qu'un langage. Les arbres que nous considérons seront vus comme les composés d'un arbre initial arbitrairement large et d'un vecteur de degré supérieur arbitrairement grand, c'est à dire d'une liste arbitrairement longue d'arbres finaux. Afin de pouvoir toujours réaliser aisément cette décomposition, les arbres initiaux et les arbres finaux seront construits sur des alphabets disjoints.

Le plus petit alphabet nécessaire pour construire des arbres initiaux arbitrairement larges est composé d'un symbole d'arité 2 que nous noterons b. Plutôt que de puiser les arbres initiaux dans $T(\{b\})$, nous avons préféré nous limiter aux arbres balancés de $T(\{b\})$, ce qui nous permettra de nous appuyer sur leurs propriétés de structure qui sont importantes.

Aussi définirons nous la forêt B des arbres balancés comme suit :

$b^o = x_1$ et pour tout $i > 0, b^i = b(b^{i-1}, b^{i-1})$. D'où $B = \{b^i \mid i \in \mathbb{N}^+\}$.

Nous écrirons $b^i = \tilde{b}^i.e_1^i$ avec $\tilde{b}^i \in \mathcal{T}(\{b\})^1_{2^i}$ et $e_1^i \in \Theta_1^{2^i}$, e_1^i s'écrit $<1 ; x_1, \ldots x_1>$ avec 2^i occurrences de x_1. On notera aussi $\bar{B} = \{x_1\} \cup B$.

On composera avec chaque arbre de B des arbres pris dans une même forêt d'une famille de forêts de $T(\Sigma \setminus b)^1_o$.

Pour toute famille

$\mathcal{F}$ de forêts de $T(\Sigma \setminus b)^1_o$ et nous définirons alors la forêt $B\mathcal{F}$ de $T(\Sigma)$ avec $b \in \Sigma$ par :

$B\mathcal{F} = \bigcup_{F \in \mathcal{F}} B \circledast F = \{\tilde{b}^i.[t_1,\ldots t_{2^i}] \mid i \in \mathbb{N}^+$ et $\{t_1,\ldots t_{2^i}\} \subset F \in \mathcal{F}\}$ c'est à dire que dans chaque arbre de $B\mathcal{F}$ tous les arbres finaux sont pris dans une même forêt de $\mathcal{F}$.

Pour décomposer chaque arbre de $B\mathcal{F}$ nous nous donnons deux applications :

- σ_b la substitution linéaire de $T(\Sigma)$ dans $T(\Sigma \setminus b)$ qui efface tous les b et permet de sélectionner les arbre finaux :

$\sigma_b(b(x_1, x_2)) = \{x_1, x_2\}$ et $\forall\, \alpha \in T(\Sigma \setminus b)$, $\sigma(\alpha) = \{\alpha\}$

- $\bar{\sigma}_b$ une application de $T(\Sigma)$ dans $T(\{b\})$ qui ne conserve que les b :

$\bar{\sigma}_b(b(t_1, t_2)) = b(\bar{\sigma}_b(t_1), \bar{\sigma}_b(t_2))$ et $\forall\, t \in T(\Sigma)$ avec racine $(t) \neq b$, $\bar{\sigma}_b(t) = \{x_1\}$.

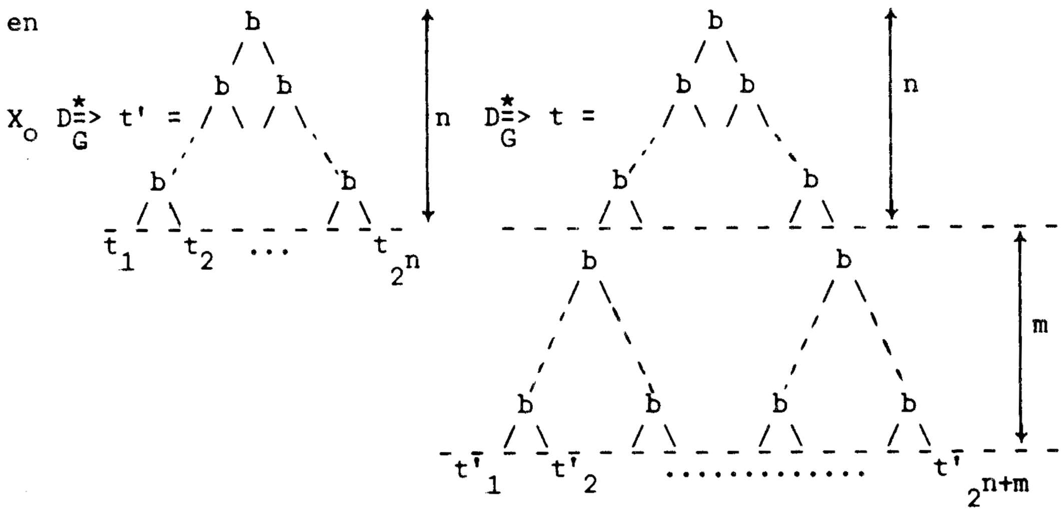
On a alors pour tout $t = \tilde{b}^i.[t_1,\ldots t_{2^i}] \in B\mathcal{F}$,

$\bar{\sigma}_b(t) = b^i$ et $\sigma_b(t) = \{t_1,\ldots t_{2^i}\} \subset F \in \mathcal{F}$,

donc $t \in \{\bar{\sigma}_b(t)\} \circledast \sigma_b(t) \subset B\mathcal{F}$.

Dans la suite on s'intéressera au cas où BF peut être engendrée par une grammaire algébrique $G = \langle V, \Sigma, X_o, R\rangle$ que l'on supposera réduite et à dérivations initiales complètes et strictes (proposition IV.20). Nous allons montrer qu'alors on peut trouver deux constantes h et k associées à G, telles que toute dérivation terminale $X_o \overset{*}{\underset{G}{D\!\!\Rightarrow}} t \in BF$ puisse se décomposer

en



avec $m \leq h$, $\{t_1, t_2, \ldots t_{2^n}\}$ contient au plus k arbres différents $t'_1, t'_2, \ldots t'_{2^{n+m}} \in T(\Sigma \setminus b)$. Etant donnée que $t \in BF$ il existe $\phi \in F$ telle $t'_1, t'_2, \ldots t'_{2^{n+m}} \in \phi$. Pour tout $i \in [2^n]$ on a nécessairement $F(G, t_i) \subset \{b^m\} \circledast \phi$, sinon on trouverait une dérivation $X_o \overset{*}{\underset{G}{D\!\!\Rightarrow}} t' \overset{*}{\underset{G}{D\!\!\Rightarrow}} t''$ avec $t'' \notin BF$. On remarquera alors que n n'étant pas borné, pour toute partie E, finie, aussi grande que l'on veut de ϕ, on peut trouver t assez large pour que $\sigma_b(t) = E$. On peut alors trouver $p \leq k$ "prototypes" $\{\tau_1, .. \tau_p\}$ tels que $F(G, \{\tau_1, .. \tau_p\}) \subset \{b^m\} \circledast \phi$, m fixé inférieur ou égal à h, et $E \subset \sigma_b(F(G, \{\tau_1, .. \tau_p\}))$.

<u>Exemple</u> BF_1. Donnons nous la grammaire due à Arnold et Dauchet [AD-ϕ-1] :

$r_0 = X_0 \rightarrow X(\#, \delta(\#, \#))$; $r_1 = X(x_1, x_2) \rightarrow X(a(x_1), Y(a(x_1), x_2))$; $r_2 = X(x_1, x_2) \rightarrow Z(d(x_1, x_2))$;

$r_3 = Z(x_1) \rightarrow b(x_1, x_1)$; $r_4 = Z(x_1) \rightarrow Z(b(x_1, x_1))$; $r_5 = Y(x_1, x_2) \rightarrow \delta(x_1, x_1)$; $r_6 = Y(x_1, x_2) \rightarrow x_2$.

posons $t_o = \delta(\#, \#)$ et $t_n = Y(a^n(\#), t_{n-1})$ alors $X(a^n(\#), t_n) \underset{r_1}{\Rightarrow} X(a^{n+1}(\#), t_{n+1})$

toute dérivation initiale est alors de la forme

$$X_o \underset{r_o}{\Rightarrow} X(\#, t_o) \overset{n}{\underset{r_1}{\Rightarrow}} X(a^n(\#), t_n) \underset{r_2}{\Rightarrow} Z(d(a^n(\#), t_n)) \overset{p}{\underset{r_4}{\Rightarrow}} (Z(b^p) \;.\; d(a^n(\#), t_n)) \underset{r_3}{\Rightarrow} (b^{p+1} \;.\; d(a^n(\#), t_n)) = t'.$$

Posons $\tau_n = d(a^n \#, t_n)$. Par récurrence sur n on montre que $F(G, \tau_n) = F_{1,n} = \{d(a^n \#, \delta(a^j \#, a^j \#)) \mid j \leq n\}$. Il suffit pour celà de remarquer que $F(G, \tau_o) = F_{1,0}$ et que si $F(G, \tau_i) = F_{1,i}$, puisque $\tau_{i+1} = d(a^{i+1} \#, Y(a^{i+1} \#, t_i))$, on a soit $\tau_{i+1} \underset{r_5}{\Rightarrow} d(a^{i+1} \#, \delta(a^{i+1} \#, a^{i+1} \#))$, soit $\tau_{i+1} \underset{r_6}{\Rightarrow} d(a^{i+1} \#, t_i) = \tau'_{i+1}$.

Par récurrence sur i on montre alors que $F(G, \tau'_{i+1}) = \{d(a^{i+1} \#, \delta(a^j \#, a^j \#)) \mid j \leq i\}$ et donc que $F(G, \tau_{i+1}) = F_{1,\, i+1}$.

En posant $\mathcal{F}_1 = \{F_{1,n} \mid n \in \mathbb{N}\}$ il est facile de voir que $F(G) = B\mathcal{F}_1$ puisque $F(G, Z) = B$.

On a alors montré que toute dérivation dans cette grammaire se décompose en $X_o \overset{*}{\underset{G}{\Rightarrow}} t' \overset{*}{\underset{G}{\Rightarrow}} t$ où $X_o \overset{*}{\underset{G}{\Rightarrow}} t'$ est la dérivation initiale maximale. Les constantes associées à cette grammaire sont alors h = 0 et k = 1. Pour toute forêt $F_{1,n} \in \mathcal{F}_1$ on a alors un unique prototype τ_n qui engendre exactement $F_{1,n}$.

Pour démontrer ce résultat nous utiliserons certaines constantes et notations liées à G que nous définissons ici :

Pour tout $X \in V$ et tout $x_i \subset \text{var}(X)$, il existe $u \in F(G, X)$ avec $x_i \in \text{var}(u)$ car G est réduite. Par contre si $E \subset \text{var}(X)$ avec Card (E) > 1 il n'existe pas toujours $u \in F(G, X)$ avec $E \subset \text{var}(u)$, sauf si G est complète, et à fortiori il n'existe pas toujours $u \in F(G, x)$ avec $E = \text{var}(u)$, par contre s'il en existe un, il en existe au moins un de profondeur minimale. Pour tout $X \in V$ et tout $E \subset \text{var}(X)$, s'il existe $u \in F(G, X)$ avec $u \notin T(\{b\})$ et $\text{var}(u) = E$, il en existe au moins un de profondeur minimale que nous notons $u_{X,E}$. Posons $H_{\bar{b}} = \text{Sup}(\rceil u_{X,E}\rceil)$ pour tout $X \in V$ et $E \subset \text{var}(X)$, $H_{\bar{b}}$ est évidemment fini.

Nous notons

$V_{b\infty} = \{X \in V \mid \forall p \in \mathbb{N}, \exists t \in F(G, X) \text{ avec } |t|_b \geq p\}$ l'ensemble des non terminaux pouvant engendrer des arbres contenant un nombre arbitrairement grand de b.

$H_b = \text{Sup}\ (|t|_b)$ pour tout $t \in F(G, X)$, $X \in V \setminus V_{b\infty}$. C'est la plus grande hauteur en b pouvant être engendrée depuis un non terminal ne pouvant engendrer des arbres de hauteur en b non bornée. H_b est alors évidemment fini.

$H_r = \text{Sup}\ (\rceil t\rceil)$ pour toute règle $(X \to t) \in R$. C'est la barre supérieure des parties droites de règles de G.

$d(V) = \text{Sup}(d(X))$ pour $X \in V$. C'est le degré du non terminal de plus haut degré.

$V_f = \{X \in V \mid F(G, X) \text{ est fini}\}$.

$H = \text{Sup}\ (|t|)$ pour tout $t \in F(G, X)$ et $X \in V_f$.

Nous rappelons qu'à la suite du corollaire 4, en l'absence d'indications contraires, tout arbre ou vecteur surmonté du signe ~ sera initial et toute torsion surmontée du signe ^ (res -) sera complète (resp. linéaire).

Lemme IV.21 Si $t \in B\mathcal{F}$, toutes les branches de t allant de la racine aux feuilles ont le même nombre d'occurrences de b.

Preuve

Evidente par définition de B et $\mathcal{F}$.

□

Lemme IV.22 Si $\tilde{u}.\tau \in B\mathcal{F}$ avec $\tilde{u} \in \mathcal{T}(\Sigma)_1^1$ alors $\bar{\sigma}_b(\tau) \in \bar{B}$.

Preuve

Si $\bar{\sigma}_b(\tau) \notin B$ et est de profondeur non nulle, il y a dans τ deux branches de la racine aux feuilles portant des nombres inégaux d'occurrences de b, il en est alors de même pour $\tilde{u}.\tau$, ce qui contredit le lemme IV.21.

□

Lemme IV.23 Si $\tilde{u}.\tau \in B\mathcal{F}$ et $\tilde{u}.\tau' \in B\mathcal{F}$ avec $\tilde{u} \in \mathcal{T}(\Sigma)_1^1$ et $|\tilde{u}| > 0$ alors $\bar{\sigma}_b(\tau) = \bar{\sigma}_b(\tau')$.

Preuve

$|\tilde{u}| > 0$ entraîne $\tilde{u} = b(u_1, u_2)$ avec soit $x_1 \in var(u_1)$, soit $x_1 \in var(u_2)$. Sans rien perdre en généralité nous supposerons $x_1 \in var(u_1)$. Posons $\bar{\sigma}_b(\tilde{u}.\tau) = b^p$. Donc toute branche joignant la racine de $\tilde{u}.\tau$ à une de ses feuilles porte p occurrences de b. D'après le lemme IV.22 on a $\bar{\sigma}_b(\tau) = b^q$ et $\bar{\sigma}_b(\tau') = b^{q'}$. Explicitons $\tilde{u}.\tau' = b(u_1.\tau', u_2)$. Les branches allant de la racine de $\tilde{u}.\tau'$ à une feuille de u_2 portent p occurrences de b, comme dans $\tilde{u}.\tau = b(u_1.\tau, u_2)$ et celles allant de la racine de $\tilde{u}.\tau'$ à une feuille de τ' en portent $p-q+q'$. D'où d'après le lemme IV.21, $q = q'$.

□

<u>Lemme IV.24</u> Pour toute dérivation $X_o \overset{*}{\underset{G}{D\Rightarrow}} u.\vec{v}$ avec $|u| > 0$, tout élément de $F(G, u)$ est de la forme $b(u_1, u_2)$ et pour tout $x_i \in var(u)$, il existe $b(u_1, u_2) \in F(G, u)$ avec $x_i \in var(u_1) \cup var(u_2)$.

<u>Preuve</u>

G est à dérivations initiales strictes. Donc $|u| > 0$ entraine, $\forall u' \in F(G,u)$ $|u'| > 0$. G est totale donc $u.\vec{v} \overset{*}{\underset{G}{D\Rightarrow}} u'.\vec{v} \overset{*}{D\Rightarrow} t \in BF$ avec $u' \in F(G, u)$ d'où racine (u') = racine $(t) = b$. G étant réduite, le lemme II,16 montre que pour tout $x_i \in var(u)$, il existe $u' \in F(G, u)$ avec $x_i \in var(u')$ et comme $u' = b(u_1, u_2)$, $x_i \in var(u_1) \cup var(u_2)$.

□

<u>Lemme IV.25</u> Pour toute dérivation $X_o \overset{*}{\underset{G}{D\Rightarrow}} \tilde{u}.X.\vec{v}$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)^1_1$, $|\tilde{u}| > 0$, $X \in \Sigma \cup V$ et $\vec{v} \in T(\Sigma \cup V)^{d(X)}_o$ alors $\bar{\sigma}_b(F(G, X.\vec{v}))$ est réduit à un seul élément qui est de la forme b^p.

<u>Preuve</u>

$\tilde{u} \in \tilde{T}(\Sigma \cup V)^1_1$ donc $var(\tilde{u}) = \{x_1\}$. D'après le lemme IV.24 il existe $b(u_1, u_2) \in F(G, \tilde{u})$ avec $x_1 \in var(u_1) \cup var(u_2)$. Posons n, le nombre d'occurrences de x_1 dans $b(u_1, u_2)$ et τ_1 un arbre quelconque de $F(G, X.\vec{v})$ qui n'est pas vide car G est réduite. L'arbre $\tilde{u}' = b(u_1, u_2).[t_1, t_2, .. t_n]$ avec $t_1 = x_1$ et pour i de 2 à n, $t_i = \tau_1$ est initial car $\tau_1 \in F(G, X.\vec{v}) \subset T(\Sigma)^1_o$ puisque $X.\vec{v} \in T(\Sigma \cup V)^1_o$. Donc $\tilde{u}' \in \tilde{T}(\Sigma)^1_1$. Pour tout $\tau \in F(G, X.\vec{v})$ on a $\tilde{u}'.\tau \in BF$ et le lemme IV.22 nous dit que $\bar{\sigma}_b(\tau) \in \bar{B}$, de même le lemme IV.23 nous dit que pour tout $\tau' \in F(G, X.\vec{v})$, $\bar{\sigma}_b(\tau) = \bar{\sigma}_b(\tau')$ d'où l'existence de p tel que $\bar{\sigma}_b(F(G, X.\vec{v})) = b^p$.

□

Ce lemme est extrêmement important pour la suite et permet de dégager une notion que nous appellerons **cote en b** et que nous définissons ainsi : A tout noeud ν non dominant dans un arbre t obtenu par dérivation descendante depuis l'axiome dans G, on peut associer un entier $C_b(\nu)$ appelé cote en b de ν dans t :

$X_o \overset{*}{D\Rightarrow} t = \tilde{u}.\nu.\vec{v}$, $|\tilde{u}| > 0$ alors $C_b(\nu)$ est l'entier p du lemme IV.25, tel que $\bar{\sigma}_b(F(G, \nu.\vec{v})) = b^p$. Pour alléger l'écriture nous écrirons souvent $C_b(t)$ pour $C_b(\text{racine}(t))$. L'intérêt majeur de cette notion de cote est qu'elle est invariante par dérivation en un sens que précise le corollaire suivant :

<u>**Corollaire IV.26**</u> Si $X_o \underset{G}{\overset{*}{D\Rightarrow}} \tilde{u}.X.\vec{v} \underset{r}{\Rightarrow} \tilde{u}.\tau.\vec{v}$ avec $r = (X \to \tau) \in R$, pour tout noeud de $\tilde{u}$ différent de la racine, pour tout noeud de $\vec{v}$, la cote en b est la même dans $\tilde{u}.X.\vec{v}$ et dans $\tilde{u}.\tau.\vec{v}$ et les cotes en b de X et de racine $(\tau.\vec{v})$ sont les mêmes.

La preuve découle directement de la définition de la cote et du lemme IV.25.

<u>**Corollaire IV.27**</u> Si $X_o \underset{G}{\overset{*}{D\Rightarrow}} \tilde{u}.X.\vec{v}$ et si $C_b(X)$ est définie, pour tout noeud ν de $X.\vec{v}$ on a $C_b(\nu) \leq C_b(X)$.

<u>**Corollaire IV.28**</u> Pour toute dérivation $X_o \overset{*}{D\Rightarrow} t$ avec racine (t) = b, tout noeud de t a une cote en b définie.

<u>Preuve</u>

$t = b(u_1, u_2)$. D'après le lemme IV.25, $C_b(u_1)$ est définie donc $C_b(t) = 1 + C_b(u_1)$.

□

Corollaire IV.29 Pour toute dérivation $X_o \overset{*}{D\Rightarrow} t = b(u_1, u_2)$ on a $t \in T(\Sigma \cup V \setminus V_{b\infty})$.

Preuve

Se déduit du corollaire IV.28 en remarquant qu'une occurrence de symbole de $V_{b\infty}$ ne peut avoir une cote en b définie.

□

Lemme IV.30 Si $X_o \overset{*}{D\Rightarrow} X(t_1,.. t_n) \overset{*}{D\Rightarrow} \tilde{u}.\hat{\theta}.\bar{\theta}.[t_1,.. t_n]$ avec $X \in V$, $\tilde{u}.\hat{\theta}.\bar{\theta} \in F(G, X) \setminus T(\{b\})$ alors pour toute composante τ_i de $\bar{\theta}[t_1,.. t_n] = [\tau_1,.. \tau_p]$ on a $C_b(\tau_i) \leq H_{\bar{b}} - 2$.

Preuve

Posons $E = var(\tilde{u}.\hat{\theta}.\bar{\theta})$ et $u_{X,E} \in F(G, X)$, $var(u_{X,E}) = E$, $|u_{X,E}| \leq H_{\bar{b}}$ et $u_{X,E} \notin T(\{b\})$. Par définition de $H_{\bar{b}}$ et de par l'existence de $\tilde{u}.\hat{\theta}.\bar{\theta}$ on est assuré de l'existence de $u_{X,E}$. Décomposons $u_{X,E} = \tilde{u}'.\hat{\theta}'.\bar{\theta}'$ selon le corollaire 4. Comme $var(u_{X,E}) = var(\tilde{u}.\hat{\theta}.\bar{\theta})$ et que $\hat{\theta}'$ et $\hat{\theta}$ sont complètes, on a $\bar{\theta} = \bar{\theta}'$. On en déduit la dérivation : $X(t_1,.. t_n) \overset{*}{D\Rightarrow} \tilde{u}'.\hat{\theta}'.\bar{\theta}.[t_1,.. t_n] = \tilde{u}'.\hat{\theta}'.[\tau_1,.. \tau_p] = t$. Puisque $\tilde{u}' \notin T(\{b\})$ il existe dans $\tilde{u}'$ une occurrence de symbole $\alpha \neq b$. Le corollaire IV.28 et $|\tilde{u}'| \leq H_{\bar{b}}$ entrainent que $C_b(t) \leq H_{\bar{b}}-1$ et puisque $|\tilde{u}'| > 0$ et $\hat{\theta}'$ est complète, pour tout $i \in [p]$ on a $C_b(\tau_i) < C_b(t)$.

□

Remarque La mise en évidence de la torsion $\hat{\theta}$ complète n'est pas essentielle car G est totale, sans arités inutiles et à dérivations terminales complètes, mais la preuve est plus simple et ce résultat suffit pour la suite.

<u>**Lemme IV.31**</u> Pour toute dérivation de la forme $X_o \overset{*}{D=>} X.\vec{v}$ $D=> b(u_1,u_2).\vec{v}$ $\overset{*}{D=>} u.\vec{v}$ avec $u \in F(G, b(u_1, u_2))$, si $u \notin T(\{b\})$ alors $C_b(u_1) = C_b(u_2) \leq (H_r - 1)H_b + H_{\bar{b}} - 2$.

<u>Preuve</u>

Rappelons que si t est un arbre, $C_b(t)$ signifie C_b(racine (t)). $X \to b(u_1, u_2)$ est une règle de R donc $|u_1| < H_r$ et $|u_2| < H_r$. D'après le corollaire IV.29 on a $\{u_1, u_2\} \subset T(\Sigma \cup V \setminus V_{b\infty})$. Puisque pour tous $X \in V \setminus V_{b\infty}$ et $\tau \in F(G, X)$ on a $|\tau|_b \leq H_b$ ceci entraine que $|u|_b \leq (H_r - 1) H_b + 1$. D'après le corollaire IV.28 la cote en b de u est définie. En posant $u = \tilde{u}.\hat{\theta}.\bar{\theta}$ on a $X_o \overset{*}{D=>} X.\vec{v} \overset{*}{D=>} \tilde{u}.\hat{\theta}.\bar{\theta}.\vec{v}$ et d'après le lemme IV.30, toute composante τ de $\bar{\theta}.\vec{v}$ a une cote en b, $C_b(\tau) \leq H_{\bar{b}} - 2$, donc $C_b(u) \leq (H_r - 1) H_b + H_{\bar{b}} - 1$ et on a en plus $C_b(u_1) = C_b(u_2) = C_b(u) - 1$.

□

<u>**Lemme IV.32**</u> Pour toute dérivation de la forme $X_o \overset{*}{D=>} X.\vec{w}$ $D=> Y.\vec{v}.\vec{w}$ $\overset{*}{D=>} \tau.\vec{v}.\vec{w} \overset{*}{D=>} u.\vec{w}$ avec X $D=> Y.\vec{v}$, $Y \overset{*}{D=>} \tau$, $\tau.\vec{v} \overset{*}{D=>} u$, $\tau \in T(\{b\})$, $u \in T(\Sigma) \setminus T(\{b\})$ et en posant $\tau = \tilde{\tau}.\hat{\theta}.\bar{\theta}$, on obtient que toute composante t de $\bar{\theta}.\vec{v}$ a une cote en b bornée : $C_b(t) \leq (H_r - 1) H_b + H_{\bar{b}} - 2$, et que $\bar{\theta}.\vec{v}$ a son degré supérieur borné par d(V).

<u>Preuve</u>

$X \to Y.\vec{v}$ est une règle de R donc $|\vec{v}| < H_r$. On a $X_o \overset{*}{D=>} \tilde{\tau}.\hat{\theta}.\bar{\theta}.\vec{v}.\vec{w} = b(u_1, u_2).\hat{\theta}.\bar{\theta}.\vec{v}.\vec{w}$, donc d'après le corollaire IV.29, il n'y a pas de symbole de $V_{b\infty}$ dans $\bar{\theta}.\vec{v}$, ce qui implique que pour toute composante t de $\bar{\theta}.\vec{v}$ et tout $t' \in F(G, t)$ on a $]t']_b \leq (H_r - 1) H_b$. On a aussi la dérivation $X_o \overset{*}{D=>} X.\vec{w}$ $\overset{*}{D=>} u.\vec{w} = \tilde{u}.\hat{\theta}'.\bar{\theta}'.\vec{w}$ avec $u \in T(\Sigma) \setminus T(\{b\})$ et d'après le lemme IV.30, toute composante t" de $\bar{\theta}'.\vec{w}$ a une cote en b bornée, $C_b(t") \leq H_{\bar{b}} - 2$. On en déduit que toute composante t de $\bar{\theta}.\vec{v}$ a une cote en b bornée $C_b(t) \leq (H_r - 1) H_b + H_{\bar{b}} - 2$. Puisque X D=> Y.$\vec{v}$ on a degré supérieur de $\vec{v} = d(Y) \leq d(V)$, et puisque $\bar{\theta}$ est linéaire, donc injective, $\bar{\theta}.\vec{v}$ a un degré supérieur au plus égal à celui de $\vec{v}$.

□

<u>Lemme IV.33</u> Il existe deux constantes h et k, associées à G, telles que toute dérivation $X_o \overset{*}{\underset{G}{D=>}} t \in BF$ puisse d'écrire :
$X_o \overset{*}{D=>} \tilde{\nu}.\theta.\vec{\omega} \overset{*}{D=>} t$ avec $\tilde{\nu} \in \tilde{T}(\{b\})^1_n$, $\theta \in \Theta^n_p$, $\vec{\omega} \in T(\Sigma \cup V)^p_o$, $p \leq k$ et pour tout $i \in [p]$, $\bar{\sigma}_b(F(G, x_i.\vec{\omega}))$ est réduit à un seul élément b^{h_i} avec $h_i \leq h$.

<u>Preuve</u>

En explicitant la dérivation initiale on obtient : X_o D=> $X_1.\vec{v}_1$ D=> ... D=> $X_q.\vec{v}_q$ D=> $b(u_1, u_2).\vec{v}_q$ D=> $\tilde{u}.\vec{w}$ avec X_q D=> $b(u_1, u_2) \overset{*}{D=>} \tilde{u}.\theta"$ et $\vec{w} \in F(G, \theta".\vec{v}_q)$. Deux cas se présentent selon que $\tilde{u} \in T(\{b\})$ ou non.

1) $u \notin T(\{b\})$, alors prenons $\tilde{\nu} = b$, $\theta = <2 ; x_1, x_2>$ et $\vec{\omega} = <0, u_1.\vec{v}_q, u_2.\vec{v}_q>$. Des lemmes IV.21 et IV.25 on déduit $\bar{\sigma}_b(F(G, u_1.\vec{v})) = \bar{\sigma}_b(F(G, u_2.v)) = \{b^i\}$ et du lemme IV.31, $i \leq (H_r - 1) H_b + H_{\bar{b}} - 2$.

2) $u \in T(\{b\})$, alors posons i le plus grand indice compris entre 0 et q-1 tel que $X_o \overset{*}{D=>} X_i.\vec{v}_i$ D=> $X_{i+1}.\vec{v}_{i+1} \overset{*}{D=>} \tilde{u}_i.\vec{w}_i = t$ dans la dérivation précédente, avec $\vec{v}_{i+1} = \vec{v}.\vec{v}_i$, X_i D=> $X_{i+1}.\vec{v}$, $X_{i+1} \overset{*}{D=>} \tau$, $\tau.\vec{v} \overset{*}{D=>} \tilde{u}_i.\theta_i$, $\vec{w}_i \in F(G, \theta_i.\vec{v}_i)$, $\tau \in T(\{b\})$ et $\tilde{u}_i \in T(\Sigma) \setminus T(\{b\})$. L'existence d'un tel indice i résulte directement de l'hypothèse $u \in T(\{b\})$ et du fait que $F(G, X_o)$ et $T(\{b\})$ sont disjoints.
Le lemme IV.32 s'applique alors et en posant $\tau = \tilde{\nu}.\hat{\theta}.\bar{\theta}$, $\omega = \bar{\theta}.\vec{v}$, on a $X_o \overset{*}{D=>} \tilde{\nu}.\hat{\theta}.\vec{\omega} \overset{*}{D=>} t$ avec $\tilde{\nu} \in \tilde{T}(\{b\})^1_n$, $\hat{\theta} \in \Theta^n_p$, $\vec{\omega} \in T(\Sigma \cup V)^p_o$, $p \leq d(V)$ et pour

tout $i \in [p]$ on a $C_b(x_i.\vec{\omega}) \leq (H_r - 1)\, H_b + H_{\bar{b}} - 2$ donc $\bar{\sigma}_b(F(G, x_i.\vec{\omega}))$ contient un seul élément b^{h_i} avec $h_i \leq (H_r - 1)\, H_b + H_{\bar{b}} - 2$ selon le lemme IV.25. Il suffit alors de regrouper les deux cas en posant $h = (H_r - 1)\, H_b + H_{\bar{b}} - 2$ et $k = \mathrm{Sup}(d(V), 2)$.

□

Si l'on appelle prototype toute composante de $\vec{\omega}$, le lemme signifie que toute partie E de $F \in \mathcal{F}$ peut se retrouver dans les sous arbres de la forêt engencrée par un nombre borné par k de prototypes. De plus ces prototypes engendrent des arbres de hauteur en b bornée par h, ils doivent donc être construits au cours d'une dérivation initiale. Si E est disjointe de toutes les autres forêts de $\mathcal{F}$, ceci implique que tout prototype ne peut engendrer un arbre portant un sous arbre appartenant à $F' \neq F$, sinon G pourrait engendrer un arbre non dans $B\mathcal{F}$. C'est ce qu'exprime le lemme suivant qui nous sera utile dans les applications traitées. En effet les forêts de $\mathcal{F}$ y seront toutes disjointes deux à deux, on aura donc pour tout $E \subset F \in \mathcal{F}$ et tout $F' \neq F$, $E \cap F' = \emptyset$.

Lemme IV.34 Pour toute grammaire $G = \langle \Sigma, V, X_o, R \rangle$ réduite et à dérivations initiales complètes strictes qui engendre $B\mathcal{F}$, il existe deux constantes h et k telles que pour toute partie finie $E \subset F \in \mathcal{F}$ avec($F \neq F'$ implique $E \cap F' = \emptyset$), il existe $t \in T(\Sigma \cup V)^1_o$ et $q \leq h$ avec $F(G, t) \subset \{b^q\} \circledast F$ et $\mathrm{Card}(\sigma_b(F(G, t)) \cap E) \geq \mathrm{Card}(E) / k$.

Preuve

Posons $n = \mathrm{Card}(E)$. Il existe $\tau \in B\mathcal{F}$ avec $\sigma_b(\tau) = E$ et $\bar{\sigma}_b(\tau) = b^r$, $r > h$. Considérons la dérivation $X_o \overset{*}{\underset{D}{\Rightarrow}} \tilde{V}.\theta.\vec{\omega} \overset{*}{\underset{D}{\Rightarrow}} \tau$ du lemme IV.33, en supposant θ complète, (si tel n'était pas le cas, il suffirait d'utiliser la décomposition du lemme 3 pour s'y ramener).

Avec les notations du lemme IV.33, $\vec{\omega}$ a p composantes $\omega_1, .. \omega_p$ et $|\tilde{\nu}| \geq 1$ car $r > h$. Pour engendrer τ il faut que $E \subset \bigcup_{i\in[p]} \sigma_b(F(G, \omega_i))$ donc il existe $i \in [p]$ tel que $\mathrm{Card}(\sigma_b(F(G, \omega_i)) \cap E) > n/p$. Le lemme IV.33 dit aussi que $\bar{\sigma}_b(F(G, \omega_i)) = \{b^q\}$ et que $q \leq h$.

Il reste à montrer que $F(G, \omega_i) \subset b^q \circledast F$, c'est à dire que, sachant que $\bar{\sigma}_b(F(G, \omega_i)) = \{b^q\}$, il faut montrer que $\sigma_b(F(G, \omega_i)) \subset F$.

On a $|\tilde{\nu}| \geq 1$ donc $var(\tilde{\nu})$ contient au moins 2 variables x_{i_1} et x_{i_2} dont l'une x_{i_1} est telle que $\theta(i_1) = i$.

$\tilde{\nu}.\theta.\vec{\omega} = \tilde{\nu}.[\vec{v}_1.\vec{\omega}, x_{i_1}.\theta.\vec{\omega}, \vec{v}_2.\vec{\omega}, x_{i_2}.\theta.\vec{\omega}, \vec{v}_3.\vec{\omega}] \overset{*}{D=>} \tau$ donc il existe τ_{i_2} tel que $x_{i_2}.\theta.\vec{\omega} \overset{*}{D=>} \tau_{i_2}$ avec $\sigma_b(\tau_{i_2}) \subset E$. S'il existait une dérivation $x_{i_1}.\theta.\vec{\omega} = \omega_i \overset{*}{D=>} \tau_{i_1}$ avec $\sigma_b(\tau_{i_1}) \not\subset F$, on aurait un sous arbre $\tau' \in \sigma_b(\tau_{i_1})$ de τ n'appartenant pas à F et un sous arbre $\tau'' \in \sigma_b(\tau_{i_2}) \subset E \subset F$ appartenant seulement à F, ce qui contredirait l'appartenance de τ à BF.

□

4) ETUDE DE QUELQUES SOUS CLASSES DES FORETS ALGEBRIQUES

Nous allons maintenant utiliser les lemmes IV.33 et IV.34 pour montrer qu'il existe des forêts algébriques qui ne peuvent être engendrées par des grammaires strictes ou complètes ou à la fois complètes et strictes. Pour celà nous utiliserons trois forêts algébriques BF_1, BF_2 et BF_3 qui serviront de contre-exemples.

Rappelons que $F_1 = \{F_{1,n} \mid n \in \mathbb{N}\}$ avec pour tout n, $F_{1,n} = \{d(a^n \#, \delta(a^j \#, a^j \#)) \mid j \leq n\}$. Arnold et Dauchet ont montré que BF_1 était algébrique en donnant une grammaire qui l'engendrait en descendant. Cette grammaire n'est ni stricte ni complète. Nous allons montrer qu'il n'existe pas de grammaire stricte G telle que $BF_1 = F(G)$. De la proposition IV.16 on déduit alors immédiatement qu'il n'existe pas de grammaire complète non plus.

Pour montrer l'inexistence de G stricte engendrant BF_1, nous utiliserons le lemme IV.34. Nous montrerons que l'on peut trouver n suffisamment grand pour qu'aucun arbre t de $T(\Sigma \cup V)$ne puisse être un prototype suffisant pour $E = F_{1,n}$ qui est fini. Rappelons, qu'aux termes du lemme IV.34 il faudrait que $F(G, t) \subset b^p \circledast F_{1,n}$ avec $p \leq h$ et Card $(\sigma_b(F(G, t))) \geq$ Card $(F_{1,n}) / k = (n+1) / k$. Le nombre Card $(\sigma_b(F(G, t)))$ dépend directement du nombre d'occurrences X de symbole dans t pouvant engendrer une occurrence de symbole δ et du nombre r d'entiers i différents tels que chaque occurrence de δ ainsi crée aie comme fils $a^i\#$. Nous dirons alors que si dans t une occurrence X de symbole n'engendre aucun arbre contenant un symbole δ son poids est nul. Par contre s'il existe r entiers distincts $i_1, i_2, .. i_r$ tels que pour tout $j \in [r]$, il existe $\tilde{v}.[v_1, \delta(t_1, t_2), \vec{v}_2] \in F(G, X)$ et $a^{i_j} \# \in t_1 \circledast F(G, \vec{v})$ c'est à dire que $\tilde{v}.[\vec{v}'_1, \delta(a^{i_j}\#, t'_2), \vec{v}'_2] \in F(G, X.\vec{v})$, nous dirons que le poids de X est Poids $(X) = r$. Chaque occurrence de δ dans un arbre de $F(G, t)$ étant engendrée depuis une occurrence de symbole dans t, il est clair que $\mathrm{Card}(\sigma_b(F(G, t))) \leq \sum_{X \text{ dans } t}$ Poids (X). Remarquons que si $F(G, t) \subset b^p \,\theta\, F_{1,n}$, tous les Poids de tous les noeuds de t seront inférieurs ou égaux à n car G est réduite.

<u>Lemme IV.35</u> Si $F(G, t) \subset b^p \,\theta\, F_{1,n}$, pour toute décomposition $t = \tilde{u}.\tau$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)_1^1$ on a : $a^q \# \in F(G, \tau)$ ssi $\{a^q \#\} = F(G, \tau)$.

<u>Preuve</u>

G étant réduite, il existe $u' \in F(G, \tilde{u})$ avec $\mathrm{var}(u') = \{x_1\}$. Posons m le nombre d'occurrences de x_1 dans u'. Décomposons u' en $u' = \tilde{u}'.\theta$ où $\tilde{u}' \in \tilde{T}(\Sigma)_m^1$ et θ est l'unique élément de Θ_1^m car $u' \in T(\Sigma)_1^1$. On peut écrire :
$t \overset{*}{\underset{D}{\Rightarrow}} u'.\tau \overset{*}{\underset{D}{\Rightarrow}} \tilde{u}'.[\tau'_1, \dots \tau'_{i-1}, \tau, \tau'_{i+q}, \dots \tau'_m] = \tilde{t}'.\tau$ avec $\tau'_1, \dots \tau'_{i-1}, \tau'_{i+1}, \dots \tau'_m \in F(G, \tau)$ et $\tilde{t}' \in \tilde{T}(\Sigma)_1^1$. Si $a^q \# \in F(G, \tau)$ on a $t \overset{*}{\underset{D}{\Rightarrow}} \tilde{t}'.\tau \overset{*}{\underset{D}{\Rightarrow}} \tilde{t}'.a^q\# \in b^p \,\theta\, F_{1,n}$. On en déduit que $\tilde{t}'$ s'écrit de l'une des trois façons suivantes :

$b^p.[\vec{v}_1, d(a^{n-q}\, x_1, v_2), \vec{v}_3]$

$b^p.[\vec{v}_1, d(a^n \#, \delta(a^{r-q}\, x_1, a^r \#)), \vec{v}_2]$

$b^p.[\vec{v}_1, d(a^n \#, \delta(a^r \#, a^{r-q} \#)), \vec{v}_2]$.

Dans chaque cas, le fait que l'on doive avoir $F(G, \tilde{t}'.\theta) \subset b^p \,\theta\, F_{1,n}$ implique $F(G, \tau) = \{a^q \#\}$.

□

<u>Lemme IV.36</u> Toute occurrence X de symbole dans $t \in T(\Sigma \cup V)_0^1$ avec $F(G, t) \subset b^p \,\theta\, F_{1,n}$, est de poids borné par une constante N ne dépendant que de la grammaire.

Preuve

Si $X \in \Sigma \setminus \delta$ on a Poids(X) = 0.

Si $X = \delta$, alors $t = \tilde{u}.\delta(t_1, t_2)$ et puisque $F(G, t) \subset b^p \oplus F_{1,n}$, il existe r tel que $t_1 \overset{*}{\underset{G}{\Rightarrow}} a^r \#$ et d'après le lemme IV.35 on a $F(G, t_1) = \{a^r \#\}$ donc Poids(δ) = 1. Si $X \in V$, nous allons nous intéresser aux δ qui peuvent être engendrés . Quand un δ est engendré depuis X, il l'est en même temps que le début de son premier fils. Ce premier fils et la variable qui le termine sont les éléments qui serviront à déterminer le poids de X.

Notons $\Delta(X)$ l'ensemble de couples d'entiers (m, i) défini par

$(m, o) \in \Delta(X) \iff \exists\, \tilde{v}.[\vec{v}_1, \delta(a^m \#, \tau), \vec{v}_2] \in F(G, X)$ et

$(m, i) \in \Delta(X) \iff \exists\, \tilde{v}.[\vec{v}_1, \delta(a^m x_i, \tau), \vec{v}_2] \in F(G, X)$.

Du lemme IV.35 et de la définition de $\Delta(X)$ on déduit que Poids (X) $\leq$ Card($\Delta(X)$). Il suffit de remarquer que $t = \tilde{u}.X.\vec{v}$ et que pour tout $x_i \in var(X)$ s'il existe $(m, i) \in \Delta(X)$ le lemme IV.35 s'applique et $card(F(G, x_i, \vec{v})) = 1$. Tout arbre de $b^p \oplus F_{I,n}$ est de profondeur p+n+3 donc F(G, t) est finie et F(G, X) est à fortiori finie.

Posons $\hat{N} = Sup(Card(\Delta(Z)))$ pour $Z \in V$ et F(G, Z) finie, cette borne existe car d(v) et H_r sont finis dans toute grammaire. On a alors Poids(X) $\leq \hat{N}$. Il suffit alors de prendre $N = Sup(\hat{N}, 1)$.

□

<u>Lemme IV.37</u> Pour tout arbre $t \in T(\Sigma \cup V)_o^1$ avec $F(G, t) \subset b^p \oplus F_{1,n}$ et G stricte et réduite, on a Card $(\sigma_b(F(G, t))) \leq N\, D^{p+2}$, N étant la constante du lemme IV.36 et D = Sup(d(X)) pour $X \in \Sigma \cup V$.

Preuve

Le principe de la preuve est de montrer que toutes les occurrences de symboles de poids non nul sont dans un sous arbre initial dont la hauteur ne dépend pas

de n. Ces occurrences sont alors en nombre borné ne dépendant pas de n et de poids borné. Pour celà notons $\tilde{u}$ le plus petit sous arbre initial de τ qui contienne toutes les occurrences X de symboles de t telles que $F(G, X) \not\subset T(\{a, \#\}) = T(\Sigma \setminus \{b, d, \delta\})$. Montrons que $|\tilde{u}| \leq p + 2$. Si tel n'était pas le cas, il existerait X à une profondeur $q > p+2$ dans t avec $\nu.[\vec{\omega}_1, \alpha(t_1, t_2), \vec{\omega}_2] \in F(G, X)$, $\alpha \in \{b, d, \delta\}$. L'arbre t s'écrirait alors $t = X_1.[\vec{v}_1, X_2.[\vec{v}_2, X_3.[\ldots X_q.[\vec{v}_q, X.\vec{w}, \vec{w}_q],..] , \vec{w}_2], \vec{w}_1]$.
Pour tout $i \in [q]$ notons p_i la position de X_{i+1} ou X sous X_i. G étant réduite, pour tout X_i il existe $t_i \in F(G, X_i)$ avec $x_{p_i} \in var(t_i)$ et ni $F(G, \vec{v}_i)$ ni $F(G, \vec{w}_i)$ ne sont vides. De plus, G étant stricte, $|t_i| > 0$.
Il existerait alors $t' \in F(G, t)$ avec un symbole $\alpha \in \{b, d, \delta\}$, engendré depuis X, à une profondeur supérieure ou égale à q donc supérieure à p+2 dans t', ce qui contredit $t' \in b^p \circledast F_{1,n}$. Prenons donc $\tilde{u}$, défini précédemment et de hauteur au plus p+2, et décomposons $t = \tilde{u}.\vec{v}$. Pour toute occurrence X de symbole dans $\vec{v}$, $F(G, X) \subset T(\{a, \#\})$ donc Poids (X) = 0.
Donc $Card(\sigma_b(F(G, t))) \leq \sum_{X \text{ dans } \tilde{u}} Poids(X) \leq \sum_{X \text{ dans } \tilde{u}} (N)$ d'après le lemme IV.36.
De plus $|\tilde{u}| \leq p+2$ et $D = Sup\ (d(Z))$ pour $Z \in \Sigma \cup V$ entraine que le nombre d'occurrences de symboles dans $\tilde{u}$ est d'au plus D^{p+2} d'où le résultat.

□

Proposition IV.38 BF_1 ne peut être engendrée par une grammaire stricte.

Preuve

Si BF_1 peut être engendrée par une grammaire stricte, elle peut l'être par une grammaire $G = \langle \Sigma, V, X_o, R \rangle$ à la fois stricte, réduite et à règles initiales complètes d'après la proposition IV.20. Toutes les forêts F_1 sont disjointes deux à deux, ce qui permet d'appliquer le lemme IV.34. Prenons alors $n > k\ N\ D^{h+2}$, $Card(F_{1,n}) = n+1$ donc il existe $t \in T(\Sigma \cup V)^1_o$ et $q \leq h$

tels que $F(G, t) \subset b^q \,@\, F_{1,n}$ et $Card(\sigma_b(F(G, t))) \geq Card(F_{1,n}) / k$.
Or $Card(F_{1,n}) / k > \frac{n}{k} > N\ D^{h+2}$, ce qui est en contradiction avec Card $(\sigma_b(F(G, t))) \leq N\ D^{q+2} < N\ D^{h+2}$ obtenu par le lemme IV.37.

□

Posons pour tout $n \in \mathbb{N}$, $F_{2,n} = \{\alpha^i\ \gamma\ a^j\ \#\ |\ i+j = n\}$.
$\mathcal{F}_2 = \{F_{2,n}\ |\ n \in \mathbb{N}\}$. Nous allons montrer que la forêt $B\mathcal{F}_2$ peut être engendrée par une grammaire stricte et que par contre elle ne peut être engencrée par aucune grammaire complète.

Donnons nous la grammaire :

```
r  = (X  →    X  ) ; r  = (X     Z) ;
 o     o     / \      1     o     |
            γ   #                 γ
            |                     |
            #                     #

r  = (  X    →     X    ) ; r  = (  X    →    Z    ) ; r  = (Z  →    Z   ) ;
 2     / \        / \        3     / \        |         4    |        |
      x   x      Y   a            x   x       Y              x        b
       1   2    / \  |             1   2     / \              1      / \
               x   a x                      x   a                   x   x
                1  |  2                      1  |                    1   1
                   x                            x
                    2                            2

r  = (Z  →   b   ) ; r  = (  Y    → γ ) ; r  = (  Y    → α ).
 5    |     / \       6     / \      |     7     / \      |
      x    x   x           x   x     x          x   x     x
       1    1   1           1   2     2          1   2     1
```

Si la règle r_1 est employée, on a $F(G, Z\ \gamma\ \#) = B \,@\, F_{2,0}$.
Si la règle r_o est employée, on emploiera ensuite n fois r_2 puis une fois r_3.
En posant $t_o = \gamma\ \#$ et $t_{i+1} = Y(t_i, a^{i+1}\ \#)$ pour tout i. On a alors

```
   X    D=>       X           =      X
  / \   r        /  \               / \
 t   a i  2     Y    a i+1      t    a i+1
  i  |         / \   |           i+1 |
     #        t   a i+1 #            #
               i  |
                  #
```

et $\begin{array}{ccc} & X & \\ / & & \backslash \\ t_i & & a_i \\ & & | \\ & & \# \end{array} \underset{r_3}{D\!\!=\!\!>} \begin{array}{ccc} & Z & \\ & | & \\ & Y & \\ / & & \backslash \\ t_i & & a^{i+1} \\ & & | \\ & & \# \end{array} = \begin{array}{c} Z \\ | \\ t_{i+1} \end{array}$ donc pour tout $n \in \mathbb{N}$

$$X_o \underset{r_o}{D\!\!=\!\!>} \begin{array}{ccc} & X & \\ / & & \backslash \\ t_o & & \# \end{array} \overset{n}{\underset{r_2}{D\!\!=\!\!>}} \begin{array}{ccc} & X & \\ / & & \backslash \\ t_n & & a^n \\ & & | \\ & & \# \end{array} \underset{r_3}{D\!\!=\!\!>} \begin{array}{c} Z \\ | \\ t_{n+1} \end{array}$$

Comme $F(G, Z) = B$ il suffit de montrer par récurrence sur n que $F(G, t_{n+1}) = F_{2,n+1}$. C'est vrai pour $n = 0$ car $t_1 = \begin{array}{ccc} & Y & \\ / & & \backslash \\ \gamma & & a \\ | & & | \\ \# & & \# \end{array}$, $t_1 \underset{r_6}{D\!\!=\!\!>} \gamma\, a\, \#$,

$t_1 \underset{r_7}{D\!\!=\!\!>} \alpha\, \gamma\, \#$ et $F_{2,1} = \{\gamma\, a\, \#,\ \alpha\, \gamma\, \#\}$. Supposons que $F(G, t_n) = F_{2,n}$ alors

$t_{n+1} = \begin{array}{ccc} & Y & \\ / & & \backslash \\ t_n & & a^{n+1} \\ & & | \\ & & \# \end{array}$, $t_{n+1} \underset{r_6}{D\!\!=\!\!>} \gamma\, a^{n+1}\, \#$, $t_{n+1} \underset{r_7}{D\!\!=\!\!>} \alpha\, t_n$

d'où $F(G, t_{n+1}) = \{\gamma\, a^{n+1}\, \#\} \cup (\{\alpha\} \circledS F_{2,n}) = F_{2,n+1}$. •

Pour montrer que BF_2 ne peut être engendrée par une grammaire complète, on utilisera le lemme IV.34 comme dans la proposition IV.38. Nous montrerons que l'on peut trouver n suffisamment grand pour qu'aucun arbre $t \in T(\Sigma \cup V)$ ne puisse être un prototype suffisant pour $E = F_{2,n}$ qui est fini. Pour celà nous nous appuierons sur l'observation des occurrences de symboles qui engendrent des arbres contenant le symbole γ.

<u>**Lemme IV.39**</u> Pour tout arbre $t = \tilde{u}.\tau$ tel que $t \in T(\Sigma \cup V)^1_o$, $F(G, t) \subset b^p \circledS F_{2,n}$ pour p et n fixés, $\tilde{u} \in \tilde{T}(\Sigma \cup V)^1_1$ et $a^q\, \# \in F(G, \tau)$ on a : $F(G, \tau)$ contient le seul élément $a^q\, \#$.

Preuve

G étant réduite, il existe $u' \in F(G, \tilde{u})$ avec $var(u') = \{x_1\}$ et $\tilde{u}.\tau \overset{*}{\Rightarrow} u'.\tau \overset{*}{\Rightarrow} u'.a^q \# \in b^p \oplus F_{2,n}$ donc $u' = \tilde{\nu}.[\vec{\omega}_1, \alpha^{n-q-j} \gamma a^j x_1, \vec{\omega}_2]$ avec $\tilde{\nu} \in \tilde{T}(\{b\})$. Il est alors clair qu'il existe $\tau' \in F(G, \tau)$, on a alors $t \overset{*}{\Rightarrow} \tilde{\nu}.[\vec{\omega}_1, \alpha^{n-q-j} \gamma a^j \tau', \vec{\omega}_2]$ et comme $F(G, t) \in b^p \oplus F_{2,n}$ il faut que $\tau' = a^q \#$.

□

<u>Lemme IV.40</u> Si G est réduite complète, pour tout arbre $t \in T(\Sigma \cup V)_o^1$ avec $F(G, t) \subset b^p \oplus F_{2,n}$, le nombre d'occurrences X de symboles dans t telles que F(G, X) contient un arbre contenant une occurrence de γ est au plus 2^p.

Preuve

Pour tout $\tau \in T(\Sigma \cup V)_o^1$, notons $N_\gamma(\tau)$ le nombre d'occurrences X de symboles dans τ avec $\tilde{\nu}.[\vec{\omega}_1, \gamma(t_1), \vec{\omega}_2] \in F(G, X)$. Montrons par récurrence sur $|\tau|$ qu'il existe $\tau' \in F(G, \tau)$ avec $N_\gamma(\tau') \geq N_\gamma(\tau)$, ce qui est évident si $|\tau| \leq 1$. Supposons ceci vrai si $|\tau| < \ell$ et posons $\tau = \alpha(\tau_1, .. \tau_m)$ avec $|\tau| = \ell$ et $\alpha \in \Sigma \cup V$. Pour tout $i \in [m]$ on a $|\tau_i| < \ell$ donc par hypothèse de récurrence il existe $\tau'_i \in F(G, \tau_i)$ avec $N_\gamma(\tau'_i) \geq N_\gamma(\tau_i)$.
Si $F(G, \alpha)$ ne contient aucune occurrence de γ on a $N_\gamma(\tau) = \sum_{i=1}^{m} N_\gamma(\tau_i)$.
G étant totale et complète, il existe $\alpha' \in F(G, \alpha)$ avec pour tout $i \in [m]$, $k_i > 0$ occurrences de x_i dans α'. Donc $\tau \overset{*}{\Rightarrow} \tau' = \alpha'.[\tau'_1, .. \tau'_m]$ et $N_\gamma(\tau') = \sum_{i=1}^{m} k_i N_\gamma(\tau'_i) \geq \sum_{i=1}^{m} N_\gamma(\tau_i) = N_\gamma(\tau)$. Si $F(G, \alpha)$ contient au moins une occurrence de γ, on a $N_\gamma(\tau) = 1 + \sum_{i=1}^{m} N_\gamma(\tau_i)$. Par le même raisonnement que plus haut on montre alors qu'il existe $\tau' \in F(G, \tau)$ avec $N_\gamma(\tau') \geq 1 + \sum_{i=1}^{m} k_i N_\gamma(\tau'_i) \geq N_\gamma(\tau)$. Pour tout $t' \in F(G, t) \subset b^p \oplus F_{2,n}$

on a $N_\gamma(t') = 2^p$ donc $N_\gamma(t) \leq 2^p$.

□

Lemme IV.41 Si G est réduite complète, il existe une constante K telle que pour tout arbre $t \in T(\Sigma \cup V)^1_o$ avec $F(G, t) \subset b^p \,\theta\, F_{2,n}$ on ait $\mathrm{Card}(\sigma_b(F(G, t))) \leq 2^p K$.

Preuve

Tout arbre de $\sigma_b(F(G, t))$ est de la forme $\alpha^i \gamma a^j \#$ avec $i + j = n$ car $F(G, t) \subset b^p \,\theta\, F_{2,n}$. Pour toute occurrence X de symbole dans t, c'est à dire que $t = \tilde{u}.X.\vec{v}$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)^1_1$, posons $E(X) = \{j \in \mathbb{N} \mid \exists\, \tilde{v}.[\vec{\omega}_1, \gamma a^j \#, \vec{\omega}_2] \in F(G, X.\vec{v})\}$. Compte tenu du lemme IV.40, il suffit de montrer que pour toute occurrence X de symbole dans t on a $\mathrm{Card}(E(X)) \leq K$ pour avoir $\mathrm{Card}(\sigma_b(F(G, t))) \leq 2^p K$.

E(X) est la réunion des ensembles

$E_o(X) = \{j \in \mathbb{N} \mid \exists\, \tilde{v}.[\vec{\omega}_1, \gamma a^j \#, \vec{\omega}_2] \in F(G, X)\}$ et pour tout $x_i \in \mathrm{var}(X)$

$E_i(X) = \{j \in \mathbb{N} \mid \exists\, \tilde{v}.[\vec{\omega}_1, \gamma a^k x_i, \vec{\omega}_2] \in F(G, X)$ et

$\exists\, a^q \# \in F(G, x_i.\vec{v})$ avec $k + q = j\}$

$F(G, t)$ est finie car contenue dans $b^p \,\theta\, F_{2,n}$ qui ne contient que des arbres de hauteur p+n+2. Tout symbole de t engendre alors des arbres de hauteur bornée par une constante H de la grammaire. D'où $\mathrm{Card}(E_o(X)) \leq H-2$. Si $\tilde{v}.[\vec{\omega}_1, \gamma a^k x_i, \vec{\omega}_2] \in F(G, X)$ on a $k \leq H-1$ et si $a^q \# \in F(G, x_i.\vec{v})$, d'après le lemme IV.39, on a $\mathrm{Card}(F(G, x_i.\vec{v})) = 1$. On en tire donc $\mathrm{Card}(E_i(X)) \leq H-1$. En notant $D = \mathrm{Sup}(d(X))$ pour tout $X \in \Sigma \cup V$, on a alors

$$E(X) = E_o(X) + \sum_{i=1}^{d(X)} E_i(X) \leq H-2 + D(H-1) = K.$$

□

Proposition IV.42 BF_2 ne peut être engendrée par une grammaire complète.

Preuve

Posons $G = \langle \Sigma, V, X_o, R \rangle$ une grammaire complète, que l'on peut supposer réduite et à dérivations initiales strictes d'après la proposition IV.20.
Les forêts de $\mathcal{F}_2$ sont disjointes deux à deux, on appliquera donc le lemme IV.34.
Posons $n > 2^h k\ K$ avec h, k les constantes du lemme IV.34 et K la constante du lemme IV.41. La forêt $F_{2,n}$ est finie, donc $\exists\ t \in T(\Sigma \cup V)^1_o$ et $q \leq h$ avec $F(G, t) \subset b^q \circledast F_{2,n}$ et $\mathrm{Card}(\sigma_b(F(G, t))) \geq \mathrm{Card}(F_{2,n}) / k > 2^h K > 2^q K$ d'après le lemme IV.34, ce qui contredit le lemme IV.41.

□

Posons pour tout $n \in \mathbb{N}$, $F_{3,n} = \{\delta(a^j \#, a^n \#) \mid j \leq n\}$ et $\mathcal{F}_3 = \{F_{3,n} \mid n \in \mathbb{N}\}$. La forêt BF_2 est algébrique et nous allons montrer qu'elle peut être engendrée par une grammaire complète mais pas par une grammaire à la fois complète et stricte. Donnons les règles :

```
r  = (X  →   X  ) ; r  = (  X     →   X   ) ; r  = (  X     →    Z   ) ;
 o     o    / \      1     / \       / \       2     / \         |
           #   #          x1  x2    Y   a           x1  x2       δ
                                    |   |                       / \
                                    x1  x2                     x1  x2

r  = (Z  →    Z    ) ; r  = (Z  →    b    ) ; r  = (Y  → a ) ;
 3    |       |         4    |      / \        5    |     |
      x1      b              x1   x1   x1           x1    x1
             / \
            x1  x1

r  = (Y  → x ). La règle r  n'est pas stricte.
 6    |     1            6
      x1
```

Toute dérivation initiale est de la forme :

```
X   D=>    X    D^n=>    X      D=>    Z    D^p=>  Z .     δ      D=>  b^(p+1) .     δ
 o  r     / \   r       / \     r      |    r      |      / \     r                / \
     o   #   #   1   Y^n   a^n   2     δ     3     b^p Y^n   a^n   4           Y^n   a^n
                     |     |          / \              |     |                 |     |
                     #     #       Y^n   a^n           #     #                 #     #
                                   |     |
                                   #     #
```

Par les règles r_5 et r_6, Y^n # engendre $\{a^j \# \mid j \leq n\}$ donc $F(G, \delta(Y^n \#, a^n \#)) = F_{3,n}$ et $F(G) = B\mathcal{F}_3$.

Il est facile de voir que si l'on remplace r_6 par $\begin{array}{c} Y \\ | \\ x_1 \end{array} \to \#$, la grammaire devient stricte et engendre toujours $B\mathcal{F}_3$ mais elle n'est plus complète.

Pour prouver que $B\mathcal{F}_3$ ne peut être engendrée par une grammaire à la fois complète et stricte, nous utiliserons encore le lemme IV.34 car ici encore les forêts de $\mathcal{F}_3$ sont disjointes deux à deux. Mais ici nous ne pourrons trouver pour tout t tel que $F(G, t) \subset b^p \circledast F_{3,n}$, une borne indépendante de n pour $Card(\sigma_b(F(G, t)))$. En effet, H étant la hauteur du plus grand arbre engendré depuis un non terminal engendrant une forêt finie, on peut imaginer qu'il existe un non terminal Y tel que $F(G, Y) = \{a^i \mid i \in [H]\}$ et que pour tout $0 < r < H$ il existe un non terminal Y_r tel que $F(G, Y_r) = \{a^i \mid i \in [r]\}$. Alors pour tout n, posons q et r tels que $n = q\, H + r$, l'arbre $Y^q\, Y_r$ engendre alors n-q-1 arbres différents.

Nous introduirons alors une nouvelle constante de la grammaire : $H_1 = 2H$. Pour tout n, nous considérerons alors la partie F_n de $F_{3,n}$ définie par $F_n = \{\delta(a^{H_1^i} \#, a^n \#) \mid H_1^i \leq n,\ i > 0\}$. Ainsi pour produire une séquence de H_1^i symboles a il faut une séquence d'au moins $2(H_1^{i-1})$ non terminaux engendrant une forêt finie et si la grammaire est complète et stricte, ces non terminaux ne peuvent engendrer une séquence de moins de $2(H_1^{i-1}) > H_1^{i-1}$ symboles.

Pour toute occurrence X de symbole dans un arbre $t \in T(\Sigma \cup V)^1_o$ avec $F(G, t) \subset b^p \circledast F_{3,n}$, Si $F(G, X)$ ne contient aucun arbre avec une occurrence de δ, nous dirons que le poids de X est nul, sinon, posons $t = \tilde{u}.X.\vec{v}$ et $E(X) = \{i_1, i_2, .. i_r\} \subset \mathbb{N}^+$ tel que $\forall\, j \in E(X)$, il existe $\tilde{v}.[\vec{\omega}_1, \delta(\tau_1, \tau_2), \vec{\omega}_2]$ dans $F(G, X)$ et $a^{H_1^j}\# \in F(G, \tau_1.\vec{v})$, nous dirons que $\text{Poids}(X) = r$, il est alors immédiat que $\text{Card}(\sigma_b(F(G, t)) \cap F_n) \leq \sum\limits_{X \text{ dans } t} \text{Poids}(X)$.

Lemme IV.43 Pour toute occurrence X de symbole dans $t \in T(\Sigma \cup V)^1_o$ avec $F(G, t) \subset b^p \circledast F_{3,n}$, on a $\text{Poids}(X) \leq d(V)$.

Preuve

Posons $t = \tilde{u}.X.\vec{v}$ pour situer X dans t.

Posons $E_o(X) = \{i \in \mathbb{N}^+ \mid \exists\, \tilde{v}.[\vec{\omega}_1, \delta(a^{H_1^i}\#, \tau_2), \vec{\omega}_2] \in F(G, X)\}$ et pour tout $\ell \in [d(X)]$

$$E_\ell(X) = \{i \in \mathbb{N}^+ \mid \exists\, \tilde{v}.[\vec{\omega}_1, \delta(a^j x_\ell, \tau_2), \vec{\omega}_2] \in F(G, X),\ \exists\, a^k\# \in F(G, x_\ell.\vec{v}) \text{ et } j+k = H_1^i\}$$

Il est clair que $E(X) = \bigcup\limits_{i=o}^{d(X)} E_i(X)$ donc $\text{Poids}(X) \leq \sum\limits_{i=o}^{d(X)} \text{Card}(E_i(X))$.

Comme $H_1 = 2H > 0$ et que si $w \in F(G, X)$, $|w| \leq H$ on a immédiatement $E_o(X) = \emptyset$.

Si $w = \tilde{v}.[\vec{\omega}_1, \delta(a^j x_\ell, \tau_2) \vec{\omega}_2]$ appartient à $F(G, X)$, on a $|w| \leq H$ donc $j \leq H-1$.

Posons $\tau = x_\ell.\vec{v}$. G étant réduite, il existe $\tau' \in F(G, \tau)$ et puisque $F(G, t) \subset b^p \circledast F_{3,n}$, τ' est de la forme $\tau' = a^k\#$. Puisque G est complète, τ ne peut contenir que des symboles d'arité 0 ou 1 et s'écrit donc $\tau = X_1 X_2 \ldots X_q \bar{X}$ avec $q \leq k$ car G est stricte.

Supposons que $j + k = H_1^i$ avec $i > 0$. On a $k \geq H_1^i - H + 1 = 2H\, H_1^{i-1} - H+1$ et puisque $H \geq 1$, $H_1 \geq 2$ donc $k > H(H_1^{i-1} + 1)$

Puisque pour tout $Y \in \{X_1, X_2, .. X_q, \bar{X}\}$ et $w' \in F(G, Y)$ on a $|w'| \leq H$ (car $F(G, t) \in b^p \circledast F_{3,n}$ implique $t \in T(\Sigma \cup V_f)$) donc

$(q+1)\ H \geq k$ et $q \geq k\ /\ H-1 > H_1^{i-1}$. D'autre part $j \geq 0$ entraine $k \leq H_1^i$ donc $q \leq H_1^i$. Pour tout $w' = \tilde{v}'.[\vec{\omega}'_1,\ \delta(a^{j'}\ x_\ell,\ \tau'_2),\ \vec{\omega}'_2] \in F(G,\ X)$ et $\tau'' = a^{k'}\ \# \in F(G,\ \tau)$ on a alors $j'+k' \leq H-1 + (q+1)H-1 = H_1^i H + 2H - 2$ et puisque $i \geq 1$ et $H_1^i \geq 2$ on obtient $j' + k' \leq 2H\ H_1^i-2 < 2H\ H_1^i = H_1^{i+1}$. D'autre part $j'+k' > qH > H_1^{i-1} H \geq H_1^{i-1}$, donc si $i \in E_\ell$, il est seul. $\mathrm{Card}(E_\ell) \leq 1$. Comme $d(X) \leq d(V)$ on a bien le résultat.

□

<u>**Lemme IV.44**</u> Dans $t \in T(\Sigma \cup V)_o^1$ avec $F(G,\ t) \subset b^P \circledast F_{3,n}$ et G complète et réduite il y a au plus 2^P occurrences X de symbole telles que $\exists\ \tau \in F(G,\ X)$ avec une occurrence de δ dans τ.

<u>Preuve</u>

Cf. lemme IV.40 en remplaçant $F_{2,n}$ par $F_{3,n}$ et γ par δ.

□

<u>**Lemme IV.45**</u> Si G est réduite complète et stricte, $t \in T(\Sigma \cup V)_o^1$ et $F(G,\ t) \subset b^P \circledast F_{3,n}$ on a $\mathrm{Card}(\sigma_b(F(G,t)) \cap F_n) \lesssim 2^P\ d(V)$.

<u>Preuve</u>

Des lemmes IV.43 et IV.44 et de $\mathrm{Card}(\sigma_b(F(G,\ t)) \cap F_n) \leq \sum_{X \text{ dans } t} \mathrm{Poids}(X)$ on déduit immédiatement le résultat.

□

<u>**Proposition IV.46**</u> $B\mathcal{F}_3$ ne peut être engendrée par une grammaire à la fois complète et stricte.

Preuve

Posons $G = <\Sigma, V, X_o, R>$ la grammaire complète et stricte que l'on peut supposer réduite selon le théorème II.15. Posons $r = 2^h\, k\, d(V) + k$, h et k étant les constantes du lemme IV.34, et $n = H_1^r$, $F_n = \{\delta(a^{H_1^i}\,\#, a^n\,\#) \mid H_1^i \leq n,\ i > 0\} \subset F_{3,n}$. Comme $F_n \cap F_{3,m} = \emptyset$ si $n \neq m$, le lemme IV.34 peut être appliqué. On doit alors trouver $t \in T(\Sigma \cup V)_o^1$ avec $F(G, t) \subset b^p \,@\, F_{3,n}$ et $Card(\sigma_b(F(G, t)) \cap F_n) \geq Card(F_n) / k$. $Card(F_n) = r = 2^h\, k\, d(V) + k$ donc $Card(F_n) / k > 2^h\, d(V)$ ce qui contredit le lemme IV.45.

□

Proposition IV.47 Quelque soit $\mathcal{F}$, $B\mathcal{F}$ ne peut être engendrée par une grammaire linéaire.

Preuve

Posons $G = <V, \Sigma, X_o, R>$ et $p > 1 + Log_2(d(V)) + ((H_r-1)H_b + H_{\bar{b}} - 2)$, $F \in \mathcal{F}$ et $t \in \{b^p\} \,@\, F$. Si $t \in F(G)$, écrivons une dérivation qui engendre t dans G que l'on peut supposer à dérivations initiales complètes strictes : $X_o \overset{*}{\underset{D}{\Rightarrow}} Z.\vec{\omega} \underset{D}{\Rightarrow} b(u_1, u_2).\vec{\omega} \overset{*}{\underset{D}{\Rightarrow}} \nu.\vec{\omega} \overset{*}{\underset{D}{\Rightarrow}} t$ avec $\nu \in T(\Sigma)$, on a $\nu \in T(\{b\})$ sinon on aurait d'après le lemme IV.31 $C_b(u_1) = C_b(u_2) \leq (H_r-1)H_b + H_{\bar{b}} - 2$ donc $|t|_b < p$. La dérivation peut donc s'écrire sous la forme $X_o \overset{*}{\underset{D}{\Rightarrow}} X.\vec{w} \underset{D}{\Rightarrow} Y.\vec{v}.\vec{w} \overset{*}{\underset{D}{\Rightarrow}} \tau.\vec{v}.\vec{w} \overset{*}{\underset{D}{\Rightarrow}} u.\vec{w} \overset{*}{\underset{D}{\Rightarrow}} t$ avec $X \underset{D}{\Rightarrow} Y.\vec{v}$, $Y \overset{*}{\underset{D}{\Rightarrow}} \tau \in T(\{b\})$, $\tau.\vec{v} \overset{*}{\underset{D}{\Rightarrow}} u \in T(\Sigma) \setminus T(\{b\})$. D'après le lemme IV.32 en posant $\tau = \tau.\hat{\theta}.\bar{\theta}$, toute composante τ' de $\bar{\theta}.\vec{v}$ a une cote en b, $C_b(\tau') \leq (H_{r-1})\, H_b + H_{\bar{b}} - 2$. On en déduit que τ contient un sous arbre initial $\tilde{b}^q$ avec $q > 1 + Log_2(d(V))$ donc que $|feuillage\ (\tau)| \geq 2^q > d(v)$ et comme $\tau \in T(\{b\})$ le feuillage de τ ne contient que des variables. On a alors $Y \overset{*}{\underset{D}{\Rightarrow}} \tau$ avec un nombre d'occurrences de variables dans τ strictement supérieur à $d(Y)$, donc G n'est pas linéaire.

□

Les propriétés de régularité, linéarité, complétude et strictitude des grammaires permettent de définir dans l'ensemble des forêts algébriques autant de classes de forêts qu'il y a de combinaisons de ces propriétés. En fait les propriétés de réduction qui ont été montrées, permettent de réduire ces classes et de se ramener quelques classes strictement incluses les unes dans les autres. Ainsi la réduction de la non strictitude monadique montre que la classe des forêts engendrée par des grammaires ayant cette propriété n'a pas lieu d'être distinguée de la classe des forêts engendrées par des grammaires strictes.

Nous noterons Reg = {F(G) | G est régulière},

Lin = {F(G) | G est linéaire, c'est à dire sans duplication},

CS = {F(G) | G est à la fois complète et stricte},

C = {G(G) | G est complète},

S = {F(G) | G est stricte},

Alg = {F(G) | G est algébrique}.

<u>Théorème IV.48</u> Rég $\subsetneq$ Lin $\subsetneq$ CS $\subsetneq$ C $\subsetneq$ S $\subsetneq$ Alg.

<u>Preuve</u>

Des définitions il découle immédiatement que Rég $\subseteq$ Lin, CS $\subseteq$ C, CS $\subseteq$ S et que toutes ces classes sont incluses dans Alg. La forêts $\{a^n\ \alpha^n\ \# \mid n \in \mathbb{N}\}$ est engendrée par $\{X_o \to \# + X\ \#\ ;\ X(x_1) \to a\ X\ \alpha(x_1) + a\alpha(x_1)\}$ qui est linéaire, et si l'on trouvait une grammaire régulière engendrant $\{a^n\ \alpha^n\ \# \mid n \in \mathbb{N}\}$ on serait en contradiction avec la théorie des langages. Donc Reg $\subsetneq$ Lin. Le théorème III.8 nous montre que Lin $\subset$ CS et la proposition IV.47 nous montre que $\{b^p.\# \mid p \in \mathbb{N}^+\}$ qui est engendrée par la grammaire complète stricte $\{X_o \to Z\# \ ; \ Z(x_1) \to Zb(x_1, x_1) + b(x_1, x_1)\}$, ne peut être engendrée par une grammaire linéaire, donc Lin $\subsetneq$ CS.

La proposition IV.46 nous montre que CS $\neq$ C, on a donc CS $\subsetneq$ C.

La proposition IV.16 nous montre que C $\subset$ S et

la proposition IV.42 que C $\neq$ S.

Enfin la proposition IV.38 nous montre que S $\neq$ Alg.

□

5) TRANSFORMATION DES FORETS ALGEBRIQUES COMPLETES STRICTES PAR HOMOMORPHISMES LINEAIRE INVERSE

Pour toute forêt BF et tout arbre t de BF, t s'écrit $\tilde{b}^p.\vec{u}$ avec $\vec{u} \in T(\Sigma \setminus b)_o^{2^p}$. Cette décomposition de t est faite suivant la "frontière" qui dans t sépare les symboles b des aurtres. On peut dire que le but essentiel du lemme IV.33 est de montrer que $\vec{u}$ qui se trouve au dessous de la frontière et peut contenir un nombre d'arbres différents arbitrairement grand doit être engendré à partir d'un nombre de "prototypes" borné. En fait on ne peut faire une coupe exactement sur la frontière et le lemme IV.33 fait la coupe à une hauteur bornée au dessus de la frontière, c'est à dire que les prototypes considérés engendrent aussi des b sur une hauteur bornée.

Pour la nouvelle application qui sera faite du lemme IV.33, il serait préférable d'obtenir une autre forme de ce lemme où la coupe serait faite à une profondeur bornée au dessous de la frontière. L'alphabet $\Sigma \setminus b$ peut contenir une grande variété de symboles et les arbres qui se trouvent au dessous de la frontière peuvent avoir des formes très diverses, aussi est-il beaucoup plus difficile de donner un énoncé général pour un lemme s'appuyant sur une coupe sous la frontière. Nous nous contenterons d'utiliser les propriétés qui servent à montrer le lemme IV.33 et la dérivation initiale sur laquelle s'appuie cette preuve.

Nous savons que la classe des forêts algébriques n'est pas fermée par homomorphisme linéaire inverse. Arnold et Dauchet l'ont montré en utilisant la forêt BF_1 qui n'est ni complète ni stricte. Certains indices porteraient à croire que ceci puisse être lié à la non strictitude, ainsi la classe des forêts de Greibach-haut est fermée par homomorphisme linéaire inverse et est incluse dans la classe des forêts strictes. Nous allons montrer que la classe des forêts algébriques complètes strictes n'est pas fermée par homomorphisme linéaire inverse et même qu'une telle application fait sortir des algébriques.

Posons $F'_{4,0} = \{b_2(\#, \#)\}$ et pour tout $i > 0$,
$F'_{4,i} = \{b_2(a^{k_i}\#, b_1(a^{k_i}\#, t)) \mid k_i \in \mathbb{N},\ t \in F'_{4,i-1}\}$ et
$F_{4,i} = \{b_1(\#, t) \mid t \in F'_{4,i}\}$. On pose alors $\mathcal{F}_4 = \{F_{4,n} \mid n \in \mathbb{N}\}$ et la forêt $B\mathcal{F}_4$ est engendrée par la grammaire complète stricte dont les règles sont :

```
r0 = (X0 →   X    ) ; r1 = (X  →    X    ) ; r2 = (X  →   Z    ) ; r3 = (Z →   Z    ) ;
             |                |     |               |      |              |     |
             b2               x1    Y               x1     b1             x1    b
            / \                    / \                    / \                  / \
           #   #                  #   x1                 #   x1              x1   x1

r4 = (Z  →   b   ) ; r5 = (  Y   →   Y    ) ; r6 = (  Y    →    b2   )
      |     / \              / \     / \              / \      / \
      x1   x1  x1           x1  x2  a   x2           x1  x2   x1  b1
                                    |                            / \
                                    x1                          x1  x2
```

En posant $t_o = b_2(\#, \#)$ et pour tout $i > 0$, $t_i = Y(\#, t_{i-1})$, toute dérivation initiale est de la forme :

```
X0 D=> X   D=p> X  D=>   Z      D=p>  ⎛Z ⎞.   b1    D=>  b^(p+1) .  b1   .
   r0  |    r1  |   r2   |       r3   ⎜| ⎟   / \    r4           / \
       to       tn       b1           ⎝b^p⎠  #  tn              #   tn
                        / \
                       #   tn
```

De plus

```
  Y    D=k>    Y     D=>     b2
 / \   r5   a^k/ \   r6   a^k/ \
#   x1      |     x1      |    b1
            #             #  a^k/ \
                              |    x1
                              #
```

d'où l'on déduit

que pour tout $n > 0$, $t_n = Y(\#, t_{n-1}) \overset{*}{\Longrightarrow} b_2(a^k\#, b_1(a^k\#, t_{n-1})$ et par induction sur n, que $F(G, t_n) = F'_{4,n}$ d'où $F(G, b_1(\#, t_n)) = F_{4,n}$.

On a alors $F(G) = \bigcup_{\substack{p\in\mathbb{N}^+\\ n\in\mathbb{N}}} B^p \circledcirc F(G, b_1(\#, t_n)) = \bigcup_{\substack{p\in\mathbb{N}^+\\ n\in\mathbb{N}}} b^p \circledcirc F_{4,n} = B\mathcal{F}_4$.

Nous savons que la classe des forêts algébriques est fermée par homomorphisme linéaire alphabétique inverse, compte tenu de la proposition 10 nous allons utiliser un homomorphisme élémentaire ordonné qui donnera de $B\mathcal{F}_4$ une image inverse non algébrique. Soit ϕ l'homomorphisme élémentaire ordonné, donc linéaire, de Σ dans Δ avec $\Sigma_o = \Delta_o = \{\#\}$, $\Sigma_1 = \Delta_1 = \{a\}$, $\Delta_2 = \{b_1, b_2, b\}$, $\Sigma_2 = \{b\}$, $\Sigma_3 = \{c\}$, défini par :
$\forall\ \alpha \in \Delta \setminus \{c\}$, $\phi(\alpha) = \alpha$ et $\phi(C(x_1, x_2, x_3)) = b_1(x_1, b_2(x_2, x_3))$.

Déterminons la forme de $\phi^{-1}(B\mathcal{F}_4)$. Pour celà, posons
$t_{k_1} = b_1(a^{k_1}\#, b_2(\#, \#))$,
$t_{k_2,k_1} = b_1(a^{k_2}\#, b_2(a^{k_1}\#, t_{k_1}))$ et pour tout $n > 1$
$t_{k_n,k_{n-1},..k_1} = b_1(a^{k_n}\#, b_2(a^{k_{n-1}}\#, t_{k_{n-1},..k_1}))$.
On a $F_{4,n} = \{t_{o,k_n,..k_1} \mid k_n,..\ k_1 \in \mathbb{N}\}$. En posant $t'_{k_1} = C(a^{k_1}\#, \#, \#)$
et pour tout $n > 1$
$t'_{k_n,..k_1} = C(a^{k_n}\#, a^{k_{n-1}}\#, t'_{k_{n-1},..k_1})$ on montre aisément par récurrence sur n que $\{t'_{k_n,..k_1}\} = \phi^{-1}(t_{k_n,..k_1})$. En effet il est clair que $\phi^{-1}(t_k) = \{t'_k\}$. Alors si $\phi^{-1}(t_{k_n,..k_1}) = t'_{k_n,..k_1}$, pour tout k_{n+1} on a

$$\begin{aligned}\phi^{-1}(t_{k_{n+1},k_n,..k_1}) &= \phi^{-1}(b_1(a^{k_{n+1}}\#, b_2(a^{k_n}\#, t_{k_n,..k_1})))\\ &= \phi^{-1}(b_1(x_1,b_2(x_2,x_3)).[a^{k_{n+1}}\#, a^{k_n}\#, t_{k_n,..k_1}])\\ &= \{C(x_1,x_2,x_3)\} \circledcirc [\phi^{-1}(a^{k_{n+1}}\#), \phi^{-1}(a^{k_n}\#), \{t'_{k_n,..k_1}\}]\end{aligned}$$

et comme pour tout $k \in \mathbb{N}$ $\phi^{-1}(a^k\#) = \{a^k\#\}$, on a

$$\phi^{-1}(t_{k_{n+1},k_n,..k_1}) = \{C(x_1,x_2,x_3).[a^{k_{n+1}} \#, a^{k_n} \#, t'_{k_n,..k_1}]\}$$
$$= \{t'_{k_{n+1},..k_1}\}.$$

Posons pour tout $n \in \mathbb{N}$, $F_{5,n} = \{t'_{o,k_n,..\,k_1} \mid k_n,..\,k_1 \in \mathbb{N}\}$.

On a $F_{5,n} = \phi^{-1}(F_{4,n})$ et puisque $\phi^{-1}(b) = \{b\}$, il est clair que si $\vec{u} = [u_1,..\,u_{2^p}] \in (F_{4,n})^{2^p}$ on a $\phi^{-1}(\tilde{b}^p.\vec{u}) = \{\tilde{b}^p.\vec{u}'\}$ avec $\vec{u}' = [u'_1,..\,u'_{2^p}]$ et pour tout $i \in [2^p]$, $\phi^{-1}(u_i) = \{u'_i\}$, donc $\vec{u}' \in (F_{5,n})^{2^p}$. En posant $F_5 = \{F_{5,n} \mid n \in \mathbb{N}\}$ il est alors clair que $BF_5 = \phi^{-1}(BF_4)$.

Si BF_5 était algébrique, il existerait une grammaire réduite et à règles initiales complètes strictes

$G = \langle V, \Sigma, X_o, R\rangle$ telle que $F(G) = BF_5$.

Posons $|t|_c$ = le nombre maximum de c sur une même branche dans t.

$V_{c\infty} = \{X \in V \mid \forall\, n \in N, \exists\, t \in F(G, X)$ avec $|t|_c \geq n\}$.

$H_c = \mathrm{Sup}(|t|_c)$ pour tout $t \in F(G, X)$ et $X \in V \setminus V_{c\infty}$

$D = \mathrm{Sup}(d(X))$ pour $X \in \Sigma \cup V$.

Nous allons d'abord montrer un lemme qui est l'équivalent du lemme IV.25 en dessous de la frontière séparant les b des autres symboles dans tout arbre de BF_5. La notion de cote en b sera alors remplacée par ce que nous appellons marque en a et c et notons μ et dont la valeur sera un couple d'entiers.Pour tout arbre $t \in T(\Sigma \cup V)$ tel que $X_o \overset{*}{D\Rightarrow} t$, pour toute occurrence X de symbole, non dominante dans t, c'est à dire que $t = \tilde{u}.X.\vec{v}$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)^1_1$ et $|\tilde{u}| > 0$, nous noterons $\mu(X) = (k, n)$ avec $k \in \mathbb{N}$ et $n \geq 0$ si $\sigma_b(F(G, X.\vec{v})) \subset \{t'_{k,k_n,..k_1} \mid k_n,..\,k_1 \in \mathbb{N}\}$, par convention d'écriture nous noterons $\mu(X) = (k,-1)$ si $\sigma_b(F(G, X.\vec{v})) = \{a^k \#\}$.

<u>Lemme IV.49</u> Pour toute dérivation $X_o \underset{G}{D\Rightarrow} \tilde{u}.X.\vec{v}$ avec $\tilde{u} \in T(\Sigma \cup V)^1_1$, $|\tilde{u}| > 0$, $X \in \Sigma \cup V$ et $\vec{v} \in T(\Sigma \cup V)^{d(X)}_o$, la marque en a et c de X est définie.

Preuve

$var(u) = \{x_1\}$ et d'après le lemme IV.24 il existe $b(u_1, u_2) \in F(G, \tilde{u})$ avec $x_1 \in var(u_1) \cup var(u_2)$. Posons p = le nombre d'occurrences de x_1 dans $b(u_1, u_2)$. Prenons arbitrairement $\tau_1 \in F(G, X.\vec{v})$ et posons $\tilde{u}' = b(u_1, u_2).[t_1, .. t_p]$ avec $t_1 = x_1$ et pour i de 2 à p, $t_i = \tau_1$. on a $\tilde{u}' \in \tilde{T}(\Sigma)^1_1$ et sans perdre en généralité, nous pouvons supposer que l'occurrence de x_1 dans $\tilde{u}'$ est sous u_1. Il est clair que $X_o \overset{*}{D=>} \tilde{u}'.X.\vec{v} \overset{*}{D=>} \tilde{u}'.\tau_1$. Donc il existe m tel que $\sigma_b(\tilde{u}'.\tau_1) \subset F_{5,m}$. Il s'ensuit que $\sigma_b(u_2.[t_1,.. t_p]) \subset F_{5,m}$ et pour tout $\tau_2 \in F(G, X.\vec{v})$ on doit avoir $\sigma_b(u_1.[t_1,.. t_p]. \tau_2) \subset F_{5,m}$. Quatre cas sont à étudier selon la position de x_1.

1er cas : x_1 est immédiatement sous un b.
$u_1.[t_1,.. t_p].\tau_2 = b^q.[t'_1,.. t'_{2^q}]$ et il existe $i \in [2^q]$ tel que $t'_i = \tau_2$. Puisque $\sigma_b(u_1.[t_1,.. t_p].\tau_2) \subset F_{5,m}$ on doit avoir $\sigma_b(\tau_2) \subset F_{5,m}$ donc $\sigma_b(\tau_2) \subset \{t'_{o,k_m,..k_1} \mid k_m,.. k_1 \in \mathbb{N}\}$ et $\mu(X) = (0, m)$.

2ème cas : x_1 est le troisième fils d'un symbole C.
$u_1.[t_1,.. t_p].\tau_2 = b^q.[t'_1,.. t'_{2^q}]$ et il existe $i \in [2^q]$ tel que $t'_i = C(\#, a^{k_m} \#, C(..C(a^{k_{n+2}} \#, a^{k_{n+1}} \#, \tau_2)...))$. Puisque $\sigma_b(u_1.[t_1,.. t_p]. \tau_2) \subset F_{5,m}$ on doit avoir $t'_i \in F_{5,m}$ donc si $n \geq 0$, $\tau_2 \subset \{t'_{k_{n+1},k_n,..j_1} \mid k_n,.. k_1 \in \mathbb{N}\}$ d'où $\mu(X) = (k_{n+1}, n)$, et si $n = -1$, dans ce cas on a aussi $k_{n+1} = 0$, alors $\tau_2 = \#$ et $\mu(X) = (0, -1)$.

3ème cas : x_1 est sous (ou est) le second fils d'un symbole C.
$u_1.[t_1,.. t_p].\tau_2 = u'.C(a^{k'}, a^j.\tau_2, \tau)$. Si $\tau = \#$ alors $a^j.\tau_2 = \#$ donc $j = 0$, $\tau_2 = \#$ et $\mu(X) = (0, -1)$. Si $\tau = C(a^{k''} \#, \tau', \tau'')$ alors $a^j.\tau_2 = a^{k''} \#$ donc $\tau_2 = a^{k''-j} \#$ et $\mu(X) = (k''-j, -1)$.

4ème cas x_1 est sous (ou est) le premier fils d'un symbole c.

Ou bien $u_1.[t_1,.. t_p].\tau_2 = b^q.[t'_1,.. t'_{2^q}]$ et il existe $i \in [2^q]$ tel que $t'_i = C(\tau_2, \tau', \tau'')$, c'est à dire qu'il y a un seul C au dessus de τ_2, alors $t'_i \in F_{5,m}$ entraine $\tau_2 = \#$ et $\mu(X) = (0, -1)$. Ou bien $u_1.[t_1,.. t_p].\tau_2 = u'.C(\tau', a^{k'} \#, C(a^j.\tau_2, \tau'',\tau'''))$, alors $a^j.\tau_2 = a^{k'} \#$ d'où $\tau_2 = a^{k'-j} \#$ et $\mu(X) = (k'-j, -1)$.

□

Corollaire IV.50 Si $X_o \overset{*}{\underset{D}{\Rightarrow}} \tilde{u}.X.\vec{v}$ avec $\tilde{u} \in \tilde{T}(\Sigma \cup V)_1^1$ et $|\tilde{u}| > 0$ alors $X \in V_{c\infty}$.

Preuve

Résulte immédiatement du lemme IV.49 et du fait que si $X \in V_{c\infty}$ il ne peut avoir de marque en a et c.

□

Lemme IV.51 Il existe une constante h_c telle que $\forall X \in V$, $\forall x_i \in var(X)$ et $\forall t \in F(G, X)$ avec $x_i \in var(t)$, si dans t il y a une occurrence de x_i qui n'est sous aucune occurrence de c alors $|t|_c \leq h_c$.

Preuve

Si tel n'était pas le cas, il existerait $X \in V$ et $x_i \in var(X)$ tels que $\forall n \in \mathbb{N}$, on pourrait trouver $t \in F(G, X)$ avec $t = \tilde{u}.[\vec{v}_1, x_i, \vec{v}_2]$, $|\tilde{u}|_c = 0$ et $|t|_c \geq n$ d'où $X \in V_{c\infty}$. D'après le corollaire IV.50, X ne peut apparaitre qu'en sommet dans une dérivation. Prenons une telle dérivation : $X_o \overset{*}{\underset{D}{\Rightarrow}} X(t_1,.. t_p)$ en posant $p = d(X)$.

D'après le lemme IV.49, on a $\mu(t_i) = (k_i, n_i)$.

Prenons $t \in F(G, X)$ avec $|t|_c \geq n > n_i$ et $t = \tilde{u}.[\vec{v}_1, x_i, \vec{v}_2]$ avec

$|\tilde{u}|_c = 0$. On obtient

$X_o \overset{*}{D\Rightarrow} t.[t_1,..t_p] = \tilde{u}.[\vec{v}_1, x_i, \vec{v}_2].[t_1,.. t_p]$

$\overset{*}{D\Rightarrow} \tilde{u}.[\vec{v}_1, x_i, \vec{v}_2].[t'_1, .. t'_p] = \tau$ avec pour tout $j \in [p]$

$t'_j \in F(G, t_j)$. On a alors $\tau \in B\mathcal{F}_5$ et $|\tilde{u}|_c = 0$ donc $\tilde{u} \in \mathcal{T}(\{b\})$ d'où

$\sigma_b(t'_i) \subset F_{5,n_i}$ et $\tau \in B \circledast F_{5,n_i}$. Par contre $|t|_c > n_i$ et $\tau = [t.\ t'_1,..t'_p]$ entrainent $\tau \notin B \circledast F_{5,n_i}$. Cette contradiction montre que l'on ne peut trouver $t \in F(G, X)$ avec $x_i \in var(X)$ sous aucune occurrence de c dans t et $|t|_c$ arbitrairement grand, l'existence de h_c est donc montrée.

□

<u>Lemme IV.52</u> Il existe une constante H_1 telle que si $X \to Y.\vec{v}$ est une règle de G avec $Y \in \Sigma \cup V$, alors $\forall\ \vec{v}' \in F(G, \vec{v})$, $|\vec{v}'|_c \leq H_1$.

<u>Preuve</u>

Du corollaire IV.50 on déduit que $\vec{v}$ ne contient aucun élément de $V_{c\infty}$. De plus $|\vec{v}| < H_r$ d'où $|\vec{v}'|_c < H_r\ H_c = H_1$.

□

Pour terminer la preuve, nous allons montrer que les propriétés de G qui viennent d'être montrées sont en contradiction avec le fait que G puisse engendrer certains arbres de $B\mathcal{F}_5$ que nous allons définir maintenant. Posons pour tout $k > 0$, $t_{1,k} = C(a^k\ \#, \#, \#)$, pour tout $n > 1$, $t_{n,k} = C(a^k\ \#, a^k\ \#, t_{n-1,k})$ et $\bar{t}_{n,k} = C(\#, a^k\ \#, t_{n-1,k})$. Il est clair que pour tout $n > 1$, $\bar{t}_{n,k} \in F_{5,n}$ et $k \neq k'$ entraine $t_{n,k} \neq t_{n,k'}$.

Pour tous $p \geq 1$, $n > 1$, posons $\hat{t}_{p,n} = \tilde{b}^p.[\bar{t}_{n,1}, \bar{t}_{n,2},..\bar{t}_{n,2^p}] \in B\mathcal{F}_5$.

L'intérêt majeur de $\hat{t}_{p,n}$ est que pour toute occurrence de C située sous au moins un autre C, la marque a et c permet d'identifier le sous arbre dans lequel on se trouve, car $\mu(racine(t_{m,k})) = (k,m)$.

Lemme IV.53 Si $X_o \overset{*}{=>} \hat{t}_{p,n}$ avec $n > H_c + h_c + H_1$ alors $p \leq \mathrm{Sup}(Q, h)$ où h est la constante du lemme IV.33 et Q el plus grand entier tel que $2^Q \leq d(V)$.

Preuve

Si $p > h$, la dérivation est dans le cas 2 de la preuve du lemme IV.33.

Détaillons cette dérivation :

$X_o \; D{=}{>} \; X_1.\vec{v}_1 \; D{=}{>} \; X_2.\vec{v}_2 \; D{=}{>} \; \ldots \; D{=}{>} \; X_q.\vec{v}_q \; D{=}{>} \; b(u_1,u_2).\vec{v}_q \; D\overset{*}{=}{>} \; u.\vec{v}_q \; D\overset{*}{=}{>} \; \hat{t}_{p,n}$

avec $X_q \; D\overset{*}{=}{>} \; u \in T(\{b\})$ et $|u| > 0$.

Posons $\hat{j}$ le plus petit j tel qu'il existe $x_i \in var(X_j)$ avec $\mu(racine(x_i.v_j)) = (0, n)$, $X_j \; D\overset{*}{=}{>} \; u_j \in T(\Sigma)$, $x_i \in var(u_j)$ et $u_j.\vec{v}_j \; D\overset{*}{=}{>} \; \hat{t}_{p,n}$.

C'est à dire que $\vec{v}_{\hat{j}}$ est le premier $\vec{v}_j$ qui contienne un prototype complet effectivement utilisé pour construire $\hat{t}_{p,n}$.

Un tel $\hat{j}$ existe et $1 < \hat{j} < q$. En effet puisque $X_q \; D\overset{*}{=}{>} \; u \in T(\{b\})$, il existe $x_i \in var(X_q)$ tel que $x_i \in var(u)$ et $\mu(racine(x_i.\vec{v}_q)) = (0, n)$ car $u.\vec{v}_q \; D\overset{*}{=}{>} \; \hat{t}_{p,n}$. D'autre part, puisque $n > H_1$, pour tout $x_i \in var(X_1)$ on a $\mu(racine(x_i.v_1)) = (k, m)$ avec $m < n$ car $X_o \to X_1.\vec{v}_1$ est une règle de G et le lemme IV.52 s'applique. On peut alors appliquer le lemme IV.51 à $X_{\hat{j}}$ et puisque $n > h_c$ on en déduit que $|u_{\hat{j}}|_c < n$.

On a alors $X_{\hat{j}-1}.\vec{v}_{\hat{j}-1} \; D{=}{>} \; X_{\hat{j}}.\vec{w}_{\hat{j}}.\vec{v}_{\hat{j}-1}$

$$D\overset{*}{=}{>} \; u_{\hat{j}}.\vec{w}_{\hat{j}}.\vec{v}_{\hat{j}-1} \; D\overset{*}{=}{>} \; \hat{t}_{p,n}.$$

D'après le lemme 52, pour tout $\vec{w} \in F(G, \vec{w}_{\hat{j}})$ on a $|\vec{w}|_c \leq H_1$ donc pour tout $t \in F(G, u_{\hat{j}}.\vec{w}_{\hat{j}})$ on a $|t|_c \leq H_1 + h_c < n$ donc pour tout sous arbre $\bar{t}_{k,n}$ de $\hat{t}_{p,n}$ il doit exister $x_\ell \in var(X_{\hat{j}-1})$ tel que $\mu(x_\ell.\vec{v}_{\hat{j}-1}) = (k, m)$ et comme $d(X_{\hat{j}-1}) \leq d(V)$ on a $2^p \leq d(V)$.

□

Proposition IV.54 BF_5 n'est pas algébrique.

<u>Preuve</u>

Le lemme IV.53 montre que si $n > H_c + h_c + H_1$ et $p > \text{Sup}(Q, h)$, $\hat{t}_{p,n} \notin F(G)$ or $\hat{t}_{p,n} \in BF_5$.

□

Bibliographie

[A, te] A. ARNOLD
Systèmes d'équations dans le magmoïde. Ensembles rationnels et algébriques. Thèse d'état. 1977. Lille

[AD. c] A. ARNOLD - M. DAUCHET
Transductions de forêts reconnaissables monadiques. Forêts corégulières. RAIRO inf. th. 10 (1976) n° 3 pp 5-28.

[AD - d] A. ARNOLD - M. DAUCHET
Un théorème de duplication pour les forêts algébriques. J. Comput. System Sci. 13 (1976) pp 223-244.

[AD-m1] A. ARNOLD - M. DAUCHET
Théorie des magmoïdes (1). RAIRO inf. th. 12 (1978) n° 3 pp 235-257.

[AD-m2] A. ARNOLD - M. DAUCHET
Théorie des magmoïdes (2). RAIRO inf. th. 13 (1979) n° 2 pp 135-154.

[AD-ϕ-1] A. ARNOLD - M. DAUCHET
Forêts algébriques et homomorphismes inverses. Information and control 37 (1978) pp 182-196.

[AD-r] A. ARNOLD - M. DAUCHET
Caractérisation algébrique des ensembles d'arbres récursivement énumérables. 1er colloque AFCET - SMF - Paris (1978) T2, pp 1-14.

[AL] A. ARNOLD - B. LEGUY
Une propriété des forêts algébriques de Greibach. Publication interne IT-12-79. Lille.

[AN] A. ARNOLD - M. NIVAT
Non deterministic recursive program schemes. In "Fundamentals of computation theory" (1977). Poznam-Lecture Notes in Computer Sciences N° 56 Springer-Verlag pp 12-21.

[AUL] G.M. ADELSON - VELSKI and Y.M. LANDIS
An algorithm for the organization of information. Doklady Academia Nauk USSR, 146, (1962) pp 263-266. English translation in Soviet Math. 3, 1259-63.

[Bo] G. BOUDOL
Langages polyadiques algébriques. Théorie des schémas de programme : sémantique de l'appel par valeur. Thèse de 3ème cycle. Paris VII (1975).

[Bo2] G. BOUDOL
Sémantique opérationnelle et algébrique des programmes récursifs non déterministes. Thèse d'état. Paris VII (1980).

[Br] W.S. BRAINERD
Tree-generating regular systems. Inf. and Control 14 (1969) pp 217-231.

[Co1] B. COURCELLE
A representation of trees by languages (1). Th. Comput. Sci. 6 (1978) pp 255-279.

[Co2] B. COURCELLE
A representation of trees by languages (2). Th. Comput. Sci. 7 (1978) pp 25-55.

[Co3] B. COURCELLE
Sur les ensembles algébriques d'arbres et les langages déterministes. Quelques applications à la théorie des schémas de programme. Thèse d'état. Paris

[Da] W. DAMM
An algebraic extension of the Chomsky hierarchy. 4e CLAAP - Lille - (1979).

[Dt3] M. DAUCHET
Transductions inversibles de forêts. Thèse de 3ème cycle. Lille (1975).

[Dte] M. DAUCHET
Transductions de forêts. Bimorphismes de magmoïdes.
Thèse d'état. Lille (1977)

[Di] E.W. DIJSTRA
A discipline of Programming. Prentice Hall (1976).

[E] J. ENGELFRIET
Simple program schemes and formal languages.
Lect. Notes in Comput. Sci. (1974).

[ES] J. ENGELFRIET and E.M. SCHMIDT
IO and OI. J. Comput. System. Sci. 15 (1977) pp 328-353.
and 16 pp 67-99.

[Fi] M.J. FISCHER
Grammars with macro-like productions. 9th IEE Symp. on switching and automata theory (1968) pp 131-142.

[Gu] I. GUESSARIAN
Program transformation and algebraic semantics. Publication interne LITP 78/21 (1978). A paraître dans TCS.

[Ku] D.E. KNUTH
The art of computer programming. Volume 3. Sorting and Searching. Addison - Wesley - (1973).

[Li] E. LILIN
Une généralisation des transducteurs d'états finis d'arbres : les S- Transducteurs. Thèse 3ème cycle. Lille (1978).

[Mai] T.S.E. MAIBAUM
A generalized approach to formal languages. J. Comput. System. Sci. 9 (1974) pp 409-439.

[Mar] P. MARCHAND
Algèbres minimales des parties reconnaissables dans les algèbres libres. 4ème CLAAP Lille (1979).

[Ni] M. NIVAT
On the interpretation of recursive program schemes .
Rapport IRIA n° 84 (1974).

[P] C. PAIR et A. QUERE
Definition et étude des bilangages réguliers.
Inf and control 13 (1968) pp 565-593.

[Ro1] W.C. ROUNDS
Mappings and grammars on trees. Math. Syst. Theory 4 pp 257-287.

[Ro2] W.C. ROUNDS
Tree-oriented proofs of some theorems on context-free and indexed langages. In 2nd ACM Symp. On theory of computing. pp 109-116.

[Ru] C. RUNCIMAN
Scarcely variabled programming and Pascal. Sigplan Notices Vol 14 n° 11 (1979) pp 97-106.

[St] M. STEINBY
Syntactic algebras and varieties of subsets of algebras.
4e CLAAP (1979).

[Th] J.W. THATCHER
Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. J. Comput. System Sci 1 (1967) pp 317-322.

[Wi] N. WIRTH
Algorithms + Data Structures = Programs.
Prentice-Hall (1976).

INDEX