N° d'ordre : 775

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR de SPECIALITE

(INFORMATIQUE THEORIQUE)

par

Jean-Philippe RETHORÉ

AUTOMATES A PILES COMMUNICANTES

Soutenue le 6 Février 1976, devant la COMMISSION D'EXAMEN

Membres du Jury :	MM.	P. BACCHUS	Président
		L. BOASSON	Examinateur
		V. CORDONNIER	Examinateur
		G. JACOB	Examinateur
		M. NIVAT	Rapporteur

UNIVERSITE DES SCIENCES
ET TECHNIQUES DE LILLE

DOYENS HONORAIRES DE L'Ancienne Faculté des Sciences

MM. R. DEFRETIN, H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES Des Anciennes Facultés de Droit Et Sciences Economiques, des Sciences et des Lettres

M. ARNOULT, Mme BEAUJEU, MM. BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, P. GERMAIN, HEIM DE BALSAC, HOCQUETTE, KAMPE DE FERIET, KOUGANOFF, LAMOTTE, LASSERRE, LELONG, Mme LELONG, MM. LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, NORMANT, PEREZ, ROIG, ROSEAU, ROUBINE, ROUELLE, SAVART, WATERLOT, WIEMAN, ZAMANSKI.

PRESIDENT HONORAIRE DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

R. DEFRETIN

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

M. PARREAU.

PROFESSEURS TITULAIRES

M. BACCHUS Pierre	Astronomie
M. BEAUFILS Jean-Pierre	Chimie Physique
M. BECART Maurice	Physique Atomique et Moléculaire
M. BLAYS Pierre	Géographie
M. BONNEMAN Pierre	Chimie Appliquée
M. BONTE Antoine	Géologie Appliquée
M. BOUGHON Pierre	Algèbre
M. BOUISSET Simon	Physiologie Animale
M. BOURIQUET Robert	Biologie Végétale
M. CELET Paul	Géologie Générale
M. CONSTANT Eugène	Electronique
M. CORSIN Pierre	Paléontologie
M. DECUYPER Marcel	Géométrie
M. DELATTRE Charles	Géologie Générale
M. DELHAYE Michel	Chimie Physique
M. DERCOURT Michel	Géologie Générale
M. DURCHON Maurice	Biologie Expérimentale
M. FAURE Robert	Mécanique
M. FOURET René	Physique du Solide
M. GABILLARD Robert	Electronique
M. GLACET Charles	Chimie Organique
M. GONTIER Gérard	Mécanique
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HEUBEL Joseph	Chimie Minérale
M. LANSRAUX Guy	Physique Atomique et Moléculaire
M. LEBRUN André	Electronique
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique Atomique et Moléculaire
M. LINDER Robert	Biologie et Physiologie Végétales
M. LOMBARD Jacques	Sociologie

.../...

M. LUCQUIN Michel	Chimie Physique
M. MAILLET Pierre	Sciences Economiques
M. MONTARIOL Frédéric	Chimie Appliquée
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. POUZET Pierre	Analyse Numérique
M. PROUVOST Jean	Minéralogie
M. SCHILTZ René	Physique Atomique et Moléculaire
Mme SCHWARTZ Marie-Hélène	Géométrie
M. TILLIEU Jacques	Physique Théorique
M. TRIDOT Gabriel	Chimie Appliquée
M. VAILLANT Jean	Analyse
M. VIDAL Pierre	Automatique
M. VIVIER Emile	Biologie Cellulaire
M. WERTHEIMER Raymond	Physique Atomique et Moléculaire
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS SANS CHAIRE

M. BELLET Jean	Physique Atomique et Moléculaire
M. BILLARD Jean	Physique du Solide
M. BODARD Marcel	Biologie Végétale
M. BOILLET Pierre	Physique Atomique et Moléculaire
M. BONNOT Ernest	Biologie Végétale
M. BRIDOUX Michel	Chimie Physique
M. CAPURON Alfred	Biologie Animale
M. DEPREZ Gilbert	Physique Théorique
M. DEVRAINNE Pierre	Chimie Minérale
M. GOUDMAND Pierre	Chimie Physique
M. GUILBAULT Pierre	Physiologie Animale
M. LABLACHE-COMBIER Alain	Chimie Organique
M. LACOSTE Louis	Biologie Végétale
Mme LEHMANN Josiane	Analyse
M. LOUCHEUX Claude	Chimie Physique
M. MAES Serge	Physique Atomique et Moléculaire
Melle MARQUET Simone	Probabilités
M. MIGEON Michel	Chimie Physique
M. MONTEL Marc	Physique du Solide
M. PANET Marius	Electrotechnique
M. RACZY Ladislas	Electronique
M. ROUSSEAU Jean-Paul	Physiologie Animale
M. SALMER Georges	Electronique
M. SEGUIER Guy	Electrotechnique

MAITRES DE CONFERENCES (et chargés d'Enseignement)

M. ADAM Michel	Sciences Economiques
M. ANDRE Charles	Sciences Economiques
M. ANGRAND Jean-Pierre	Géographie
M. ANTOINE Philippe	Analyse
M. BART André	Biologie Animale
M. BEGUIN Paul	Mécanique
M. BKOUCHE Rudolphe	Algèbre
M. BOILLY Bénoni	Biologie Animale
M. BONNEMAIN Jean-Louis	Biologie Végétale
M. BOSCQ Denis	Probabilités
M. BREZINSKI Claude	Analyse Numérique
M. BRUYELLE Pierre	Géographie
M. CARREZ Christian	Informatique
M. CORDONNIER Vincent	Informatique
M. CORTOIS Jean	Physique Nucléaire et Corpusculaire

.../...

M. COQUERY Jean-Marie	Psycho-Physiologie
M. COULON Jean	Electrotechnique
Mlle DACCHARI Monique	Géographie
M. DEBOURSE Jean-Pierre	Gestion des Entreprises
M. DEBRABANT Pierre	Géologie Appliquée
M. DHAINAUT André	Biologie Animale
M. DELAUNAY Jean-Claude	Sciences Economiques
M. DERIEUX Jean-Claude	Microbiologie
M. DOUKHAN Jean-Claude	Physique du Solide
M. DRIEUX Baudouin	Informatique
M. DUEE Gérard	Géologie Appliquée
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
Mme EVRARD Micheline	Chimie Appliquée
M. FONTAINE Jacques-Marie	Electronique
M. FOURNET Bernard	Biochimie
M. FROELICH Daniel	Chimie Physique
M. GAMBLIN André	Géographie
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GRANELLE Jean Jacques	Sciences Economiques
M. GUIGOU Jean-Louis	Sciences Economiques
M. GUILLAUME Henri	Sciences Economiques
M. HECTOR Joseph	Géométrie
M. HERMAN Maurice	Physique Spatiale
M. JOURNEL Gérard	Physique Atomique et Moléculaire
Mlle KOSMANN Yvette	Géométrie
M. KREMBEL Jean	Biochimie
M. LANGRAND Claude	Probabilités
M. LAURENT François	Automatique
Mlle LEGRAND Denise	Algèbre
Mlle LEGRAND Solange	Algèbre
M. LENTACKER Firmin	Géographie
M. LEROY Jean-Marie	Chimie Appliquée
M. LEROY Yves	Electronique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique Théorique
M. LOUAGE Francis	Electronique
M. MAHIEU Jean-Marie	Physique Atomique et Moléculaire
Mme N'GUYEN VAN CHI Régine	Géographie
M. MAIZIERES Christian	Automatique
M. MALAUSSENA Jean-Louis	Sciences Economiques
M. MESSELYN Jean	Physique Atomique et Moléculaire
M. MONTUELLE Bernard	Biologie Appliquée
M. NICOLE Jacques	Chimie Appliquée
M. PAQUET Jacques	Géologie Générale
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie Physique
M. PERROT Pierre	Chimie Appliquée
M. PERTUZON Emile	Physiologie Animale
M. PONSOLLE Louis	Chimie Physique
M. POVY Lucien	Automatique
M. ROGALSKI Marc	Analyse
M. ROY Jean-Claude	Psycho-Physiologie
M. SIMON Michel	Sociologie
M. SLIWA Henri	Chimie Organique
M. SOMME Jean	Géographie
Mlle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. THERY Pierre	Electronique
M. TOULOTTE Jean-Marc	Automatique
M. TREANTON Jean-René	Sociologie

.../...

M. VANDORPE Bernard	Chimie Minérale
M. VILLETTE Michel	Mécanique
M. WERNIER Georges	Informatique
M. WATERLOT Michel	Géologie Générale
M. YVON Jean-Pierre	Analyse Numérique
Mme ZINN JUSTIN Nicole	Algèbre

Monsieur Bacchus, Professeur à l'Université des Sciences et Techniques de Lille I, m'a fait l'honneur de présider le Jury de cette thèse. Je l'en remercie très vivement.

Je remercie Monsieur Boasson, Maître de Conférences à l'Université d'Amiens, pour l'intérêt qu'il a manifesté à l'égard de ce travail et qui a accepté de le juger.

Je remercie Monsieur Cordonnier et Monsieur Jacob, Maître de Conférences à l'Université des Sciences et Techniques de Lille I, qui ont bien voulu faire partie du Jury.

Je suis très reconnaissant à Monsieur Maurice Nivat, Professeur à Paris VII, dont l'intérêt pour ce travail a été décisif, et qui me fait le très grand honneur de rapporter cette thèse.

Je ne voudrais pas oublier de remercier ici, mes camarades, André Arnold, Max Dauchet et Michel Latteux, qui ont donné un essor déterminant à notre équipe et m'ont toujours soutenu.

Je remercie beaucoup Mademoiselle Françoise Desquiens qui a dactylographié ces pages et à qui mes retards ont imposé bien des efforts pendant ce mois de janvier 1976.
Monsieur et Madame Debock ont mis toute leur gentillesse et toute leur compétence dans l'élaboration matérielle, je les en remercie.

TABLE DES MATIERES

INTRODUCTION.

La théorie des langages consiste en l'étude des sous-ensembles, appelés langages, de l'ensemble de tout les mots que l'on peut écrire à l'aide des lettres d'un alphabet fini. Chomsky dans [1], partant du modèle général des grammaires à structure de phrases, en imposant différentes contraintes aux règles de productions, distingue trois grandes classes de langage, ordonnées par inclusion.

Cette classification est assez grossière, et de nombreuses tentatives ont été faites dans le but de l'affiner. C'est ainsi que Havel et Harrison dans [11] définissent les grammaires strictes déterministes, que Courcelle dans [16] définit les grammaires complètes déterministes, et qu'en 1968 Floyd [6] imaginait les grammaires opérateurs, reprises et généralisées par Colmérauer dans [3]. Et cette liste ne prétend pas être exhaustive.

Nous définissons dans ce travail les automates à deux piles, que nous appelons automates à piles communicantes. Ceux-ci vont permettre de retrouver un grand nombre de familles parmi les principales familles de langages connues, au travers d'un unique formalisme par restrictions naturelles.

Précisément nous obtenons :

- les langages récursivement énumérables [17]

- les langages à contexte liè, ainsi que les langages à contexte liè déterministes [15]

- les langages algébriques reconnus par les automates à pile de mémoire, ainsi que les langages algébriques déterministes [8]

- les langages rationnels [2]

- les langages simples étudiés dans [14]

- les langages reconnus par les automates à pile de mémoire déterministes, par réinitialisation de la pile, et par pile vide.

Ceci répond à une question soulevée par S. Eilenberg [5] qui, devant la très grande diversité des automates existants demandait s'il n'était pas possible de faire oeuvre d'unification. C'est ce que nous faisons : par exemple la suppression d'une seule possibilité de mouvement dans la classe des automates à piles communicantes reconnaissant les langages à contexte liè, nous donne la classe reconnaissant les langages algébriques : alors que les deux sont reconnues respc. par les automates linéairement bornés et les automates à piles de mémoire, fondamentalement différents.

Par ailleurs, on est toujours très intéressé, étant donné un automate, par la possibilité de lui associer une grammaire à structure de phrase, qui engendre le langage reconnu par l'automate.

C'est ce qui est fait par Ginsburg dans [8] par les automates à pile de mémoire, la grammaire, obtenue de manière relativement simple, n'est cependant pas très exploitable. De même Kuroda, [15], exhibe une grammaire associée aux automates linéairement bornés, mais la manière de l'obtenir est ici complexe, et la grammaire difficilement utilisable, elle ne fut par la suite pratiquement jamais reprise.

Nous sommes capables d'associer aux automates à piles communicantes des grammaires canoniques, ceci tout à fait facilement. Les grammaires obtenues ont des règles de productions dont en particulier la longueur des parties droites est toujours inférieure ou égale à deux, cette forme particulièrement simple, les rend très utilisables.

Donc, outre l'intérêt d'unifier, nous posséderons, avec ces automates, une correspondance canonique entre les grammaires et les automates existants qui doit permettre de retrouver à partir des propriétés des grammaires celle des automates et réciproquement.

Nous donnons comme exemple d'application une démonstration simple de l'inclusion de la classe langages déterministes dans la classe des langages BRC(1,1) définis par Floyd dans [7].

Ceci est une conséquence directe des résultats d'équivalence de nos automates avec les automates à pile de mémoire et des propriétés de la grammaire canonique.
Notons que cela fait l'objet d'une démonstration fort complexe dans [12].

Nous pensons de plus, qu'en fournissant un nouveau formalisme, nous contribuerons peut être à résoudre le problème difficile de l'équivalence de deux langages déterministes dont Courcelle [4] a montré qu'il se ramène à l'équivalence des arbres algébriques. Nous n'avons donné dans ce travail qu'un seul exemple d'application, application tout à fait non triviale des automates à piles communicantes, nous sommes convaincus que ceux-ci s'avèreront être un outil très efficace pour la résolution de beaucoup de problèmes. Grâce à eux nous croyons qu'il sera possible de donner une nouvelle caractérisation des grammaires déterministes. Ils pourraient également permettre une plus grande connaissance des langages situés entre les langages algébriques et les langages à contexte liè.
Dans une autre direction, il serait intéressant de retrouver sur les automates à piles, la trace des principales propriétés des automates classiques, en particulier, retrouver la hièrarchie proposée par Havel et Harrison dans [11], hièrarchie de langages stricts déterministes. Pour terminer, disons qu'ils pourraient également permettre la définition de nouvelles classes de langages.

Nous avons décomposé notre travail en quatre chapitres.

* Dans un premier chapitre nous donnons les définitions concernant les automates à piles communicantes : apc en abréviation.

A la suite de ces définitions nous indiquons quelques propriétés vérifiées par divers apc. Propriétés que nous utiliserons dans les chapitres suivants.

* Dans le deuxième chapitre nous établissons les relations existantes entre les différents apc définis au chapitre I et les principaux automates connus. Nous partons de la considération du plus général, la machine de Turing et nous allons jusqu'aux automates d'états finis.

Nous montrons que nous pouvons également atteindre des familles de langages déterministes, définies à partir des automates à piles de mémoire déterministes par restriction sur les configurations finales ainsi que les langages simples.

* L'objet du troisième chapitre est la construction d'une grammaire canonique associée aux apc.

En fait dans une première partie nous donnons la grammaire à structure de phrases associée à la plupart des apc.
Puis dans une deuxième partie nous construisons la grammaire algébrique associée à tout automate à piles communicantes non expansives à droite (I.3b)).

* C'est dans le quatrième chapitre que nous donnons quelques applications qui peuvent avoir les apc.

En particulier nous envisageons les applications aux grammaires BRC définies dans [7]. Nous montrons tout à fait simplement que les langages déterministes sont BRC(1,1), donc LR(1). Ce résultat à déjà fait l'objet de démonstrations, non correcte, selon Havel et Harrison qui en donnent une dont il est difficile d'aprécier la véracité.

LE CHAPITRE I.

Disons qu'un automate à piles communicantes est constitué de deux piles qui peuvent voir leur contenu croître ou bien décroître, au cours des mouvements. D'une manière tout à fait informelle, on peut imaginer la possibilité "d'incliner" l'automate et de cette manière de faire passer les "billes" de la pile gauche dans la pile droite et inversement. Il n'est pas exclu qu'au cours d'une telle manoeuvre une bille tombe, ou qu'une bille se scinde en deux permettant l'augmentation du nombre totale des billes. Le but étant de faire tomber toutes les billes sauf l'une d'entre elle.

Plus formellement :

Nous conviendrons qu'au début de l'analyse d'un mot w de l'alphabet Σ, ce mot est placé dans la pile droite et que seul un marqueur figure dans la pile gauche. Nous plaçons également un marqueur dans le fond de la pile droite. Nous noterons une telle configuration, dite initiale, de la manière suivante :

$$K_o = [\#_g, w\#_d].$$

Les crochets matérialisant chacune des piles, et la virgule la séparation entre ces piles.

Soit une configuration quelconque :

$$K = [\alpha A, B\beta], \; A \neq \Lambda \text{ et } B \neq \Lambda$$

αA est le contenu de la pile gauche, $B\beta$ celui de la pile droite A est l'élément placé au sommet de la pile gauche, - B celui placé au sommet de la pile droite.

Pour décider du mouvement qu'effectura l'automate, nous n'avons accès qu'aux sommets de pile, à savoir $\#_g$, et la lettre la plus à gauche de w pour K_o, A et B pour K. De plus, nous nous interdisons la lecture de la chaîne vide Λ en sommet de pile. En conséquence, un automate ayant vidé totalement l'une de ces piles bloquera. Aucun mouvement n'est possible à partir des configurations $[\Lambda,\alpha]$ ou $[\alpha,\Lambda]$.

De quelle façon pouvons nous remplacer A et B en sommet de pile, et ainsi évoluer vers une nouvelle configuration K' ?

Nous noterons un mouvement de K vers K' par :

$$K \vdash K'.$$

Nous pouvons remplacer A et B de 8 manières différentes donnant 8 mouvements possibles à partir de $K = [\alpha A, B\beta]$:

⌶ 2 mouvements croissants

(m1) $(A,B) \rightarrow (X_1, X_2X_2)$ $\quad K \vdash [\alpha X_1,\ X_2X_3\beta]$

(m2) $(A,B) \rightarrow (X_1X_2, X_3)$ $\quad K \vdash [\alpha X_1X_2, X_3\beta]$

⌶ 3 mouvements d'analyse

(m3) $(A,B) \rightarrow (\Lambda, X_1X_2)$ $\quad K \vdash [\alpha,\ X_1X_2\beta]$

(m4) $(A,B) \rightarrow (X_1X_2, \Lambda)$ $\quad K \vdash [\alpha X_1X_2,\ \beta]$

(m5) $(A,B) \rightarrow (X_1,\ X_2)$ $\quad K \vdash [\alpha X_1,\ X_2\beta]$

⌶ 3 mouvements de réduction

(m6) $(A,B) \rightarrow (\Lambda,\ X_1)$ $\quad K \vdash [\alpha,\ X_1\beta]$

(m7) $(A,B) \rightarrow (X_1, \Lambda)$ $\quad K \vdash [\alpha X_1,\ \beta]$

(m8) $(A,B) \rightarrow (\Lambda, \Lambda)$ $\quad K \vdash [\alpha, \beta]$

Bien entendu un même couple (A,B) pourra donner plusieurs mouvements possibles, c'est le cas le plus général. Lorsque pour tout couple (A,B) il n'existe qu'un seul mouvement à partir de ce couple, nous dirons que l'automate est déterministe.

Un mot w sera dit reconnu pour un apc, s'il existe une suite de mouvements permettant à partir de $[\#_g, w\#_d]$ d'atteindre une configuration finale K_F.

Une configuration finale est une configuration dans laquelle une des deux piles est vide, et l'autre contient un unique élément appartenant à un ensemble F, dit ensemble des éléments finaux.

$$K_F = [\Lambda, X] \text{ ou } [X, \Lambda] \text{ pour tout } X,\ X \in F$$

Si nous notons T(A), l'ensemble des mots reconnus par A et $\overset{*}{\vdash}$ la clôture transitive de $\vdash$ nous obtenons :

$$T(A) = \{w \ / \ \exists X \in F \text{ tq } [\#_g, w\#_d] \overset{*}{\vdash} [X_1, X_2] \text{ avec } X_1X_1 = X\}$$

Puis nous définissons les principaux apc, par restriction sur les mouvements possibles, on obtient :

- Les apc généraux : apc tout les mouvements sont possibles

- Les apcb : automates à piles communicantes bornées tout mouvements, moins (m1) et (m2)

- Les n.e.d : automates à piles communicantes non expansives à droite, tout mouvements, moins (m1), (m2) et (m3).

- Les s.t.d : automates à piles communicantes strictement décroissantes, seul mouvement possible (m7).

Notons que nous aurions pu admettre les mouvements de type (m2) dans le cadre des non expansifs à droite, nous verrons que cette possibilité n'augmente pas la capacité d'un n.e.d.

Ajoutons qu'en modifiant les configurations initiales, par suppression du marqueur de fond de pile droite :

$$K_o = [\#_g, w]$$

et en imposant la pile qui doit être vide dans la configuration finale, nous obtenons deux familles de langage reconnues par des n.e.d déterministes, mais distinctes de la famille reconnue par les n.e.d.d (voir I.3 f)).

Disons pour terminer, qu'en autorisant seulement les mouvements du type (m5) (m6) et (m7), nous obtenons un a.p.c dit simple, noté a.p.c.s.

Le reste du chapitre, étant consacré à la démonstration de quelques propriétés utiles dans les chapitres suivants.

Nous montrons notamment que, pour la plupart des a.p.c, on peut se passer des mouvements (m5) et (m8).

LE CHAPITRE II.

L'objet de ce chapitre est donc d'établir les relations, entre les principaux automates connus et les différents a.p.c.

Les méthodes employées pour montrer les équivalences, sont en fait toujours les mêmes. Il s'agit de simuler l'un par l'autre : pour cela nous construisons l'un à partir de l'autre. Cette construction est plus ou moins évidente suivant les cas mais jamais très difficile.

Les résultats d'équivalence sont des conséquences directes des constructions. En fait, nous définissons des configurations qui se correspondent dans l'a.p.c et dans l'automate classique, et nous montrons que les deux automates se "suivent" au travers de ces configurations dites correspondantes.

Nous obtenons les résultats suivants :

¤ Les a.p.c les plus généraux correspondent aux machines de Turing (II.2). Nous nous contentons de simuler une machine de Turing par un a.p.c. La thèse de Church ([17] p:7) nous assurant la réciproque, la construction d'une machine de Turing associée à un a.p.c ne poserait d'ailleurs aucun problème. Cette simulation ne pose que de petites difficultés techniques. Intuitivement, un élément important est que dans les deux cas la "zone de travail" n'est pas bornée. Ceci est assuré dans un a.p.c par les mouvements du type (m1) et (m2). D'autre part la simulation des déplacements possibles vers la gauche ou vers la droite dans une machine de Turing, est assurée dans un a.p.c par (m3) et (m4).

⌑ Les a.p.c (resp. déterministes) correspondent aux automates linéairement bornés (resp. déterministes). La simulation de l'un par l'autre est ici encore, uniquement technique. Constatons que la différence avec les a.p.c généraux réside en la suppression des mouvements (m1) et (m2), donc impossibilité de faire croître la longueur totale des deux piles, ce qui correspond bien au fait que dans un automate linéairement borné la bande de lecture est bornée. Les mouvements vers la gauche ou la droite sont ici encore assurés par (m3) et (m4).

⌑ Les n.e.d (resp. déterministes) correspondent aux automates à pile de mémoire (resp. déterministes). Il existe trois différences importantes entre les n.e.d et les a.p.m :

1 Un a.p.m peut lire la chaîne vide

2 Un n.e.d possède un marqueur de fin de mot

3 Dans un a.p.m on a accès à trois éléments pour décider d'un mouvement, et seulement à deux dans un a.p.c

la simulation d'un n.e.d par un a.p.m, ne pose de réels problèmes que dans le cas déterministe. Nous utilisons le résultat suivant, obtenu par S. Greibach et Ginsburg dans [10] à savoir : un langage L est déterministe si et seulement si Lc est déterministe, c étant un symbole n'appartenant pas à l'alphabet de L.
Cette propriété permet d'utiliser la même construction, que le n.e.d soit ou non déterministe (III.6 a)). Notons qu'elle concerne la $2^{\text{ième}}$ différence.

Nous montrons la simulation des a.p.m (d) par les n.e.d(d) en deux temps. Nous séparons le cas non déterministe du cas déterministe. Le cas non déterministe (II.4) est résolu en utilisant la propriété démontrée par Greibach dans [18] à savoir : tout a.p.m est équivalent à un a.p.m qui ne lit jamais la chaîne vide. Ceci permet une construction aisée dans le cas déterministe (II.5), nous utilisons une propriété moins forte que la précédente à savoir :

$$\delta(q,z,a) = (q',\gamma z') \Longrightarrow \delta(q',z',\Lambda) = \emptyset$$

lorsque un mouvement a été fait en empilant, $|\gamma z'| \geq 1$, alors le mouvement suivant ne peut se faire par lecture de la chaîne vide. Ceci permet ici encore une construction simple.

Nous nous intéressons également, (II.5 et II.6) aux langages déterministes reconnus par réinitialisation de la pile et par pile vide. Nous montrons que ces deux familles correspondent exactement aux sous-familles définies à partir du n.e.d.d en limitant les configurations initiales et finales. Ce que nous pouvons schématiser de la manière suivante :

Soit A un n.e.d.d l'ensemble des mots w tel que :

1 $[\#_g, w] \overset{*}{\underset{A}{\vdash}} [X,X']$ avec $XX' \subset F$ correspond aux langages déterministes reconnus par réinitialisation, à la chaîne vide près.

2 $[\#_g, w] \overset{*}{\underset{A}{\vdash}} [\Lambda,S]$ avec $S \subset F$ correspond aux langages déterministes reconnus par pile vide ou langage $L = \{\Lambda\}$ près

Ceci nous permet de déterminer l'influence du marqueur $\#_d$, marqueur de fond de pile droite pour les n.e.d. Il est facile de voir que dans le cas non déterministe, $\#_d$ peut être supprimé sans modifier la puissance du n.e.d.

Ȳ Les a.p.m simples correspondent exactement au a.p.c simple, et les automates d'états finis ou a.p.c strictement décroissants : s.t.d. Ces deux résultats sont tout à fait faciles à montrer.

LE CHAPITRE III.

Nous nous intéressons dans ce chapitre aux grammaires susceptibles d'être associées aux différents a.p.c.

Dans une première partie, nous construisons à partir de l'a.p.c le plus général, la grammaire à structure de phrase qui engendre le même langage. Nous imposons à l'a.p.c de vérifier quelques propriétés, toutes triviales, afin de rendre plus simple les démonstrations. Cette grammaire se déduit de l'a.p.c d'une façon très intuitive. La démonstration de l'équivalence est technique, c'est une conséquence directe de la construction, elle est basée sur un parallèle entre les mouvements dans l'automate et les dérivations dans G, par configurations correspondantes.

Cette grammaire permet de retrouver quelques résultats du chapitre précédent. Par exemple, la grammaire à structure de phrase associée à un a.p.c borné est à contexte liè.

Quant à celle associée à un n.e.d, elle n'est pas à algébrique,, à cause de mouvements (m4); mais le langage qu'elle engendre est cependant algébrique, voir (III. 1 h).

Dans la deuxième partie nous proposons une grammaire à contexte libre associée aux n.e.d. Ici encore nous n'imposons auxn.e.d de vérifier quelques propriétés triviales, sauf une que nous montrons dans le chapitre I (I.3). C'est l'obtention de cette grammaire qui nous a posé le plus de problèmes. Ces règles de production sont cependant obtenues de manière simple. La démonstration de l'équivalence se fait en établissant une correspondance biunivoque entre les configurations du n.e.d, et les expressions dites bien formées de G, qui sont des mots obtenus par dérivation la plus à droite à partir de l'axiome.
Le parallèle entre les mouvements de l'automate et les dérivations dans G, sera très utile dans le chapitre IV ou il est question d'analyse "top-down" c'est-à-dire de famille vers la racine.

LE CHAPITRE IV.

Nous consacrons la première partie de ce chapitre à la définition des grammaires "BRC(l,k)". La définition donnée dans [12] n'étant pas aisée à manier, nous donnons une définition équivalente qui fait apparaître plus clairement l'idée de contexte dans lequel une règle est applicable au cours d'une analyse "top-down", de gauche à droite, dans le sens d'une "réduction". C'est-à-dire remplacement dans le mot en cours d'analyse de la partie droite de la règle par la partie gauche.
Informellement : à chaque règles nous associons à l'aide d'une applicable c, le contexte dans lequel elle est applicable (1).
Nous dirons que G est BRC(l,k), si l'application c est une application de P (ensemble des règles de G) dans les parties de $V^l \times \Sigma^k$.

(1) applicable dans le sens d'une réduction

Ce qui signifie que nous avons accès jusqu'à une profondeur de longueur 1 sur la gauche du mot faisant l'objet de la réduction par p, et d'une profondeur k sur la droite. L'application c permettant une analyse déterministe "top-down", de la gauche vers la droite du mot.

Dans une deuxième partie nous montrons que la grammaire contexte libre associée aux n.e.d définie dans le chapitre IV est BRC(1,1), en exhibant une application c. Intuitivement : nous avons vu dans le chapitre III qu'il existe un parallèle étroit entre les mouvements de l'automate et les dérivations dans G. Or l'analyse d'un mot dans un n.e.d,part des "feuilles" c'est-à-dire du mot lui-même, pour aboutir à une configuration finale réduite à un seul élément correspondant à l'axiome de G. Cette analyse étant faite de la gauche vers la droite du mot. Nous déduisons de ce mouvement une analyse de tout mot de L(G), "top-down" correspondante à la dérivation la plus à droite. La correspondance établie entre les configurations de l'automate et les expressions bien formées de G étant biunivoque, le fait que le n.e.d soit déterministe implique que l'analyse précédente est unique pour tout mot. Nous montrons très simplement, qu'à chaque pas de cette analyse, pour lever une éventuelle ambiguité quant à la règle à appliquer, nous n'avons jamais besoin de considérer plus d'un élément à droite et plus d'un élément à gauche du mot faisant l'objet de la réduction. Ceci implique que G est BRC(1,1).

Nous indiquons en corollaire que tout langage appartenant à la formule n.e.d.d_1 est BRC(1,O).

CHAPITRE I

DEFINITION DES AUTOMATES A PILES COMMUNICANTES

Nous donnons tout d'abord la définition du concept le plus général.

I.1 AUTOMATES À PILES COMMUNICANTES : définition.

a) *Définition.*

Un automate à piles communicantes (abréviation : apc) est un quintuplet : $A = (\Gamma, \Sigma, M, \delta, F)$ tel que :

(I) . Σ et Γ deux alphabets finis disjoints. Notons $V = \Gamma \cup \Sigma$

(II) . $M = \{\#_g, \#_d\}$ ensemble de 2 marqueurs $M \cap V = \emptyset$
notons $V\#_g = V \cup \{\#_g\}$, $V\#_d = V \cup \{\#_d\}$

(III) . δ est une application de $V\#_g \times V\#_d$ dans les parties finies de $(\mathring{V\#}_g \mathring{V} \times \mathring{V\#}_d) \cup (\mathring{V\#}_g \times \mathring{V}\mathring{V\#}_d)$ (1) telle que : si $(\alpha, \beta) \in \delta(A,B)$ alors :

$$A \neq \#_g \Longrightarrow \alpha \notin \#_g V^o$$

et

$$B \neq \#_d \Longrightarrow \beta \notin V^o \#_d$$

(IV) . $F \subset V$ ensemble des éléments finaux.

Donnons à présent le formalisme utile à la description des mouvements dans un a.p.c.

b) *Définition.*

Une configuration d'un a.p.c sera un élément de la forme : $[w,w']$ avec $w \in \mathring{V\#}_g V^*$ et $w' \in V^*\mathring{V\#}_d$.

Dans la configuration $[w,w']$, nous dirons que l'automate lit les sommets de pile A et B si A est l'élément le plus à droite de w, B l'élément le plus à gauche de w'. L'élément A est donc une lettre de $\mathring{V\#}_g$ égal à Λ si et seulement si $w=\Lambda$, et l'élément B une lettre de $\mathring{V\#}_d$ égal à Λ si et seulement si $w'=\Lambda$.

c) *Définition.*

Etant donné un a.p.c. $A = (\Gamma, \Sigma, M, \delta, F)$ soit $\vdash_A$ (ou $\vdash$ lorsqu'il n'y a pas de confusion possible sur l'automate) une relation sur $\mathring{V\#}_g V^* \times V^* \mathring{V\#}_d$ définie comme suit.

(1) Pour tout ensemble E nous notons $E \cup \{\Lambda\}$ par E^o.

$$[\omega A, B\omega'] \underset{A}{\vdash} [\omega\alpha, \beta\omega'] \text{ si } (\alpha,\beta) \in \delta(A,B)$$

Pour deux configurations C et C' nous écrivons :

$$C \overset{*}{\underset{n}{\vdash}} C'$$

si $\exists C_1, C_2, \ldots, C_n$ telles que :

$$\forall i \quad 1 \leq i < n \; C_i \underset{A}{\vdash} C_{i+1}$$

. L'ensemble F des éléments finaux est utilisé pour reconnaître un sous-ensemble de Σ^* la manière suivante :

d) *Définition.*

Un mot $t \in \Sigma^*$ est accepté pour un a.p.c $A = (\Gamma, \Sigma, M, \delta, F)$ si $[\#_g, t\ \#_d] \overset{*}{\underset{A}{\vdash}} [X, X']$ avec $XX' \in F$
ou encore $[X, X'] = [y, \Lambda]$ ou $[\Lambda, y]$ avec $y \in F$.

Notons T(A) l'ensemble des mots acceptés par A.

e) Ces définitions étant données, nous pouvons répertorier les divers types de mouvements possibles dans un a.p.c. Soit $(\alpha,\beta) \in \delta(A,B)$:

- Mouvement "croissant"

Π "croissant à droite" si $\boxed{|\alpha| = 1 \text{ et } |\beta| = 2}$
la longueur totale des deux piles augmente
d'un élément placé sur la pile droite

Π "croissant à gauche" si $\boxed{|\alpha| = 2 \text{ et } |\beta| = 1}$
même chose, ici c'est la pile gauche qui augmente d'un.

- Mouvement "constant" ou d'"analyse"

Ces mouvements conservent lalongueur globale des deux piles :

⌶ "analyse droite" si $\boxed{|\alpha| = 2 \text{ et } |\beta| = 0}$

dans ce mouvement cela se passe comme si la virgule se déplaçait d'un cran sur la droite en modifiant éventuellement les sommets de pile considérés.

⌶ "analyse gauche" si $\boxed{|\alpha| = 0 \text{ et } |\beta| = 2}$

ici la virgule se déplace sur la gauche

⌶ "analyse stationnaire" si $\boxed{|\alpha| = 1 \text{ et } |\beta| = 1}$

en fait il est facile de voir que l'on pourra toujours se passer de ce type de mouvement

- Mouvement "décroissant" ou de "réduction"

⌶ "réduction à gauche" $\boxed{|\alpha| = 0 \text{ et } |\beta| = 1}$

la longueur de la pile gauche diminue de 1

⌶ "réduction à droite" $\boxed{|\alpha| = 1 \text{ et } |\beta| = 0}$

la longueur de la pile droite diminue de 1

⌶ "réduction totale" $\boxed{|\alpha| = 0 \text{ et } |\beta| = 0}$

la longueur des deux piles diminue de 1.

Remarque :

Nous aurions pu faire un autre choix pour les mouvements croissants tout en limitant la croissance à 1 élément.

Nous aurions pu opter pour $|\alpha| = 0$ et $|\beta| = 3$ pour les croissants à gauche. Ceci ne modifierai en rien les résultats que nous obtenons ; le seul fait important pour ce type de mouvement étant qu'il fasse effectivement croître la longueur totale des deux piles.

I.2

Nous allons au travers de restrictions sucessives sur ces mouvements définir 3 automates principaux déduits de l'automate à piles communicantes le plus général, et un quatrième moins important.

a) AUTOMATE À PILES COMMUNICANTES BORNÉ : abréviations a.p.c.b.

Définition.

$$\text{a.p.c.b.} \quad A = (\Gamma, \Sigma, M, \delta, F)$$

Un automate à pile communicantes borné est un automate à piles communicantes dans lequel on a interdit les mouvements croissants :

$\forall A,B \quad \forall \alpha,$ tels que : $(\alpha,\beta) \in \delta(A,B)$ alors $|\alpha\beta| \leqslant 2$

Remarquons que dans ce cas la pile gauche comme la pile droite peut croître d'un élément au cours d'un mouvement, mais leur longueur est décroissante au sens large.

b) AUTOMATE À PILES COMMUNICANTES NON EXPANSIVES À DROITE

abréviation : n.e.d (comme non expansif à droite)

Définition.

Un automate à piles communicantes non expansives à droite

$$A = (\Gamma, \Sigma, M, \delta, F)$$

est un automate à piles communicantes bornées dans lequel on a interdit les mouvements d'analyse droites :

$\forall A,B \quad \forall \alpha,\beta$ tels que : $(\alpha,\beta) \in \delta(A,B)$ alors $|\alpha\beta| \leqslant 2$ et $|\beta| \leqslant 1$

Ici comme la terminologie l'indique la pile droite ne peut jamais croître au cours d'un mouvement.

c) AUTOMATE À PILES COMMUNICANTES STRICTEMENT DÉCROISSANTES

abréviation : s.t.d comme strictement décroissant

Définition.

Un automate à piles communicantes strictement décroissantes est un n.e.d $A = (\Gamma, \Sigma, M, \delta, F)$ tel que :

$\forall A,B \quad \forall \alpha,\beta$ tels que : $(\alpha,\beta) \in \delta(A,B)$ alors $|\alpha| = 1$ et $|\beta| = 0$

Pour un tel automate il n'y a plus qu'un seul mouvement possible c'est la réduction droite. En effet les autres mouvements envisageables dans ce cadre seraient les mouvements de réduction gauche, de réduction totale mais l'un comme l'autre videraient la pile gauche et l'automate ne pouvant lire Λ en sommet de pile il bloquerait.

d) AUTOMATE À PILES COMMUNICANTES SIMPLE

Dans un tel automate ne sont plus possibles principalement que les mouvements d'analyse droite, de réduction droite et de réduction totale. En plus pour aboutir à la configuration finale et seulement dans ce cas on utilisera un mouvement de réduction gauche.

Définition.

Nous dirons qu'un a.p.c est simple si et seulement si :

$$\forall X \in \Gamma \cup \Sigma \cup \{\#_g\},\ a \in \Sigma,\ y \in \Gamma$$

1 $\quad |\delta(X,a)| \leqslant 1$ et $\delta(X,y) = \emptyset$

2 $\quad \delta(X,a) = (\alpha,\beta) \Longrightarrow (|\alpha| \leqslant 2$ et $\beta = \Lambda)$ ou $(\beta \in F$ et $\alpha = \Lambda)$

3 $\quad$ un mot w sera accepté par un a.p.c.s ssi :

$$\exists\ (\#_g, w) \overset{*}{\vdash} [\Lambda, S] \text{ avec } S \in F.$$

La restriction $\delta(X,Y) = \emptyset$, signifie qu'on autorise pas la lecture d'un élément n'appartenant pas à Σ en sommet de pile droite.

e) A chacun des automates précédents on peut associer un automate déterministe.

Définition.

Nous dirons qu'un a.p.c est déterministe ssi pour tout (A,B),

$$|\delta(A,B)| \leqslant 1 \quad (1)$$

Nous ajouterons la lettre "d" pour indiquer que l'automate est déterministe. Nous obtenons donc

a.p.c	donne a.p.c.d
a.p.c.b	donne a.p.c.b.d
n.e.d	donne n.e.d.d
s.t.d	donne s.t.d.d

Les a.p.c simples étant déterministes, par définition nous ne parlerons pas d'a.p.c.s.d.

f) A présent nous allons donner la définition de familles de langages obtenues à partir d'un n.e.d.d, en changeant les conditions de reconnaissance d'un mot.

Définition des ensembles $T_1(A)$ et $T_2(A)$ pour A n.e.d.d

Pour un n.e.d.d A nous noterons :

$$T_1(A) = \{\omega \mid \exists X \in F \ \exists \ [\#_g,\omega] \overset{*}{\vdash} [Y,Y'] \text{ avec } YY' = X\}$$

$$T_2(A) = \{\omega \mid \exists X \in F \ \exists \ [\#_g,\omega] \overset{*}{\vdash} [\Lambda,X]\}$$

Remarque :

Il est facile de voir que l'on peut sans changer $T_1(A)$, se limiter aux configurations finales du type $[X,\Lambda]$.

(1) Lorsque $\delta(A,B)$ est réduit à un seul élément, soit (u,v) cet élément, nous noterons souvent $\delta(A,B) = (u,v)$, au lieu de $\delta(A,B) = \{(u,v)\}$

Notation.

Nous pouvons par extention noter $T_o(A)$ l'ensemble :

$$T_o(A) = \{\omega \mid \exists x \in F\ ]\ [\#_g, \omega\#_d] \overset{*}{\vdash} [Y,Y'] \text{ avec } YY' = X\}$$

Nous obtenons les familles suivantes :

Définition.

$$n.e.d.d_i = \{L \ /\ \exists A \text{ n.e.d.d tel que } L = T_i(A)\}$$

Remarque :

Ces trois familles coïncident si A n'est pas déterministe, ceci est une conséquence des résultats obtenus au chapitre II.

g) Nous allons à présent donner les exemples d'automates à piles communicantes, et indiquer les langages qu'ils reconnaissent.

Exemple 1.

Nous construisons tout d'abord un a.p.c.b A reconnaissant L tel que :

$$L = \{a^n b^n c^n \ /\ n \geq 1\}$$

Soit : $A = (\Sigma, \Gamma, \delta, M, F)$

$.\Sigma = \{a, b, c\}$ $.\Gamma = \{A, \bar{A}, X, S', S\}$

$.M = \{\#_g, \#_d\}$ $.F = \{S\}$

et δ est défini comme suit :

1 $\delta(\#_g, a) = (\#_g\ a, \Lambda)$	2 $\delta(a,a) = (aa, \Lambda)$	3 $\delta(a,b) = (A, \Lambda)$
4 $\delta(A,b) = (\Lambda, \bar{A}A)$	5 $\delta(a,\bar{A}) = (A, \Lambda)$	6 $\delta(A,A) = (AA, \Lambda)$
7 $\delta(A,c) = (\Lambda, X)$	8 $\delta(A,X) = (X, \Lambda)$	9 $\delta(X,c) = (\Lambda, X)$
10 $\delta(\#_g, X) = (S', \Lambda)$	11 $\delta(S', \#_d) = (\Lambda, S)$	12 $\delta(A,\bar{A}) = (\Lambda, \bar{A}A)$

Indiquons à présent la suite des configurations amenant à la reconnaissance de $a^3b^3c^3$. Convenons que C $\overset{i^n}{\vdash}$ C', veut dire que C $\vdash_{n}^{*}$ C' grâce à la règle i, appliquée n fois.

$$[\#_g, \text{aaabbbccc}\#_d] \overset{1}{\vdash} [\#_g\text{a}, \text{aabbbccc}\#_d] \overset{2^2}{\vdash} [\#_g\text{aaa}, \text{bbbccc}\#_d]$$

$$\overset{3}{\vdash} [\#_g\text{aaA}, \text{bbccc}\#_d] \overset{4}{\vdash} [\#_g\text{aa}, \bar{\text{A}}\text{Abccc}\#_d] \overset{5}{\vdash} [\#_g\text{aA}, \text{Abccc}\#_d]$$

$$\overset{6}{\vdash} [\#_g\text{aAA}, \text{bccc}\#_d] \overset{4}{\vdash} [\#_g\text{aA}, \bar{\text{A}}\text{Accc}\#_d] \overset{12}{\vdash} [\#_g\text{a}, \bar{\text{A}}\text{AAccc}\#_d]$$

$$\overset{5}{\vdash} [\#_g\text{A}, \text{AAccc}\#_d] \overset{6^2}{\vdash} [\#_g\text{AAA}, \text{ccc}\#_d] \overset{7}{\vdash} [\#_g\text{AA}, \text{Xcc}\#_d]$$

$$\overset{8}{\vdash} [\#_g\text{AX}, \text{cc}\#_d] \overset{9}{\vdash} [\#_g\text{A}, \text{Xc}\#_d] \overset{8}{\vdash} [\#_g\text{X}, \text{c}\#_d]$$

$$\overset{9}{\vdash} [\#_g, \text{X}\#_d] \overset{10}{\vdash} [\text{S}', \#_d] \overset{11}{\vdash} [\Lambda, \text{S}]$$

<u>Exemple 2.</u>

Nous allons donner le n.e.d A qui reconnait le langage L

$$L = \{a^n b^n \ / \ n \geq 0\} \cup \{a^n \ / \ n \geq 0\}.$$

Soit A = (Σ, Γ, δ, M, F) telle que :

. Σ {a,b} . F = {C, S, B, $\bar{C}$}

. M = $\{\#_g, \#_d\}$. F = {S}

et δ est défini par :

1 $\delta(\#_g, a) = (\#_g a, \Lambda)$	2 $\delta(a,a) = (aa, \Lambda)$	3 $\delta(a, \#_d) = (\Lambda, C)$
4 $\delta(a,C) = (\Lambda, C)$	5 $\delta(\#_g, C) = (\Lambda, S)$	6 $\delta(a,b) = (A, \Lambda)$
7 $\delta(A,b) = (\Lambda, B)$	8 $\delta(a,B) = (A, \Lambda)$	9 $\delta(A, \#_d) = (\Lambda, \bar{C})$

$10\ \delta(\#_g,\bar{c}) = (\Lambda,S)$ $\qquad$ $11\ \delta(\#_g,\#_d) = (\Lambda,S)$

La reconnaissance de a^2b^2 se fait comme suit :

$$[\#_g,aabb\#_d] \overset{1}{\vdash} [\#_g a,abb\#_d] \overset{2}{\vdash} [\#_g aa,bb\#_d] \overset{6}{\vdash} [\#_g aA,b\#_d]$$
$$\overset{7}{\vdash} [\#_g a,B\#_d] \overset{8}{\vdash} [\#_g A,\#_d] \overset{9}{\vdash} [\#_g,\bar{c}] \overset{10}{\vdash} [\Lambda,S]$$

La reconnaissance de a^n se faisant comme suit :

$$[\#_g,a^n\ \#_d] \overset{1}{\vdash} [\#_g a,a^{n-1}\#_d] \overset{*2}{\vdash} [\#_g a^n,\#_d] \overset{3}{\vdash} [\#_g a^{n-1},c] \overset{*4}{\vdash} [\#_g,c]$$
$$\overset{5}{\vdash} [\Lambda,S]$$

remarquons que A est déterministe. Remarquons encore qu'il n'existe pas de n.e.d A telle que

$$T_1(A) = L$$

c'est-à-dire : $L = \{w\ /\exists\ [\#_g,w] \underset{A}{\overset{*}{\vdash}} [X,X']$ avec $XX' \in F\}$

Ceci est facile à montrer, en effet quand l'automate lit les "a" il faut que d'une part il les compte et que d'autre part, il se trouve à chaque pas de la reconnaissance du mot dans un état final, c'est deux choses sont impossibles à assurer en même temps.

<u>Exemple 3</u>.

Nous allons dans cette exemple construire un a.p.c qui reconnait les expressions arithmétiques. Nous avons simplifié au maximum, Σ est réduit à $\Sigma = \{(,),\ +,\ a\}$. Et $\Gamma = \{E,D\}$, E comme expression arithmétique. D comme début d'expression arithmétique. De plus $F = \{E\}$. Nous admettrons que ( ) est une expression arithmétique.

Donnons à présent les valeurs de la fonction δ :

$11\ \delta(\#_g,a) = (E,\Lambda)$ $\qquad$ $12\ \delta(\#_g,() = ((,\Lambda)$

$21\ \delta((,a) = ((E,\Lambda)$ $\qquad$ $22\ \delta((,() = (((,\Lambda)$ $\qquad$ $23\ \delta((,)) = (E,\Lambda)$

$31\ \delta(E,+) = (D,\Lambda)$ $\qquad$ $32\ \delta(E,)) = (\Lambda,))$

$41\ \delta(D,a) = (E,\Lambda)$ $\qquad$ $42\ \delta(D,() = ((,\Lambda)$

Remarquons que nous n'avons pas besoin de $\#_d$.

Le mot ((a+a) + a) + (a) est reconnu de la manière suivante :

$$
\begin{aligned}
& [\#_g, ((a+a) + a) + (a)] \\
\vdash\ & [(\ , \ (a+a) + a) + (a)] \\
\vdash\ & [((\ , \ a+a) + a) + (a)] \\
\vdash\ & [((E \ , \ +a) + a) + (a)] \\
\vdash\ & [((D \ , \ a) + a) + (a)] \\
\vdash\ & [((E \ , \ ) + a) + (a)] \\
\vdash\ & [((\ , \ ) + a) + (a)] \\
\vdash\ & [(E \ , \ +a) + (a)] \\
\vdash\ & [(D \ , \ a) + (a)] \\
\vdash\ & [(E \ , \ ) + (a)] \\
\vdash\ & [(\ , \ ) + (a)] \\
\vdash\ & [E \ , \ + (a)] \\
\vdash\ & [D \ , \ (a)] \\
\vdash\ & [(\ , \ a)] \\
\vdash\ & [(E \ , \ )] \\
\vdash\ & [(\ , \ )] \\
\vdash\ & [E \ , \ \Lambda]
\end{aligned}
$$

I.3 PROPRIÉTÉS CONCERNANT LES A.P.C ET LES AUTOMATES À PILES DE MÉMOIRES[1] UTILES DANS LES CHAPITRES SUIVANTS :

Cette partie pourra être passée en première lecture, nous y faisons référence lorsque cela est nécessaire.

Dans cette partie nous définissons tout d'abord, en a), deux ensembles, que nous utiliserons constamment. Brièvement disons que ce sont les ensembles des éléments apparaissant aux cours de mouvements d'un a.p.c en fond de pile. Nous montrons en b) que nous pouvons toujours faire en sorte que ces éléments soient caractéristiques des fonds de piles, c'est-à-dire qu'ils ne peuvent jamais apparaître ailleurs.

Nous définissons ensuite les a.p.c simplifiés, pour lesquels les mouvements de réduction totale et d'analyse stationnaires sont supprimés. Puis nous terminons, par la démonstration d'une propriété concernant les automates à piles de mémoire déterministes.

a) *Définition.*

Soit un a.p.c $A = (\Sigma, \Gamma, \delta, M, F)$, posons $V = \Sigma \cup \Gamma$.

Soient $T_r^o(\#_g) = \{\#_g\}$ et $T_r(\#_d) = \{\#_d\}$ alors pour tout i, $i \geq 1$ soient :

$$T_r^{i+1}(\#_g) = \{X / \exists Y' \in V \cup \{\Lambda\},\ \exists Y \in V,\ \exists Z \in T_r^i(\#_g) \text{ tels que :}$$

$$(XY', \Lambda) \in \delta(Z,Y)\} \cup T_r^i(\#_g)$$

(1) On trouvera la définition des automates à piles de mémoire au § II.4 a)

$$T_r^{i+1}(\#_d) = \{X \ / \ \exists Y' \in V \cup \{\Lambda\},\ \exists Y \in V,\ \exists Z\ T_r^i(\#_d) \text{ tels que : } (\Lambda,\ Y'X) \in \delta(Y,Z)\} \cup T_r^i(\#_d)$$

Alors :

$$T_r(\#_g) = \lim_{i\to\infty} T_r^i(\#_g)$$

$$T_r(\#_d) = \lim_{i\to\infty} T_r^i(\#_d)$$

les ensembles $T_r(\#_g)$ et $T_r(\#_d)$ sont appellés respectivement les ensembles des traces de $\#_g$ et des traces de $\#_d$.

Les limites $T_r(\#_g)$ et $T_r(\#_d)$ existent car ces ensembles sont inclus dans $\Sigma \cup \Gamma \cup M$ qui sont des ensembles finis.

. Nous énonçons maintenant une propriété dite des traces des marqueurs :

b) Propriété des traces des marqueurs : prop <u>t.d.m</u>

Un a.p.c A vérifiera la propriété <u>t.d.m</u> si et seulement si pour tout $X, X', Y' \in \Gamma \cup \Sigma \cup M$, $Y' \in (\Gamma \cup \Sigma)^o$, $v \in (\Gamma \cup \Sigma)^* \#_d^o$

si $(X'Y', v) \in \delta(X,Y)$ alors :

$$X \notin T_r(\#_g) \Longrightarrow X' \notin T_r(\#_g) \text{ et } Y' \notin T_r(\#_g)$$

si $(v, X'Y') \in \delta(X,Y)$ alors :

$$Y \notin T_r(\#_d) \Longrightarrow X' \notin T_r(\#_d) \text{ et } Y' \notin T_r(\#_d)$$

Cette propriété impose aux traces des marqueurs de ne figurer qu'en fond de pile, ce sont donc deux ensembles d'éléments caractéristiques des fonds de pile.

Proposition.

A tout a.p.c.(d) A *on peut associer un* a.p.c.(d) A' *du même type qui vérifie la prop* t.d.m *et qui est équivalent à* A.

Nous laissons le soin au lecteur de démontrer cette proposition.

Remarque :

Un même élément peut appartenir à la fois à $T_r(\#_g)$ et à $T_r(\#_d)$, ceci n'amène aucune ambiguité, si l'élément est sur la pile gauche, il marque le fond de pile gauche, et si il figure sur la pile droite, il marque le fond de la pile droite.

c) *Définition.* a.p.c simplifié

Nous dirons qu'un a.p.c $A = (\Sigma, \Gamma, \delta, M, F)$ est simplifié ssi

1 il vérifie la prop t.d.m

2 il n'effectue jamais de mouvement d'analyse stationnaire

3 il n'effectue jamais de mouvement de réduction totale.

Propriété.

Tout a.p.c (d) A, dans lequel les mouvements décroissants gauche et droit sont autorisés est équivalent à un a.p.c (d) simplifié. Dans le cas des n.e.d (d) les ensembles $T_1(1)$ et $T_2(A)$ sont conservés.

Preuve.

. Soit A vérifiant le point 1, nous savons que c'est possible par la proposition du § b), nous allons construire A' vérifiant le point 2 à partir A.

Pour cela définissons une relation, noté $\rightarrow$ entre les éléments de $(V \cup M) \times (V \cup M)$:

$$(X,Y) \rightarrow (X',Y') \iff |X| = |X'| = |Y| = |Y'| = 1$$

et $(X',Y') \in \delta(X,Y)$

soit $\overset{*}{\rightarrow}$ la clôture transitive de $\rightarrow$.

Nous dirons que : $(X,Y) \overset{*f}{\rightarrow} (X',Y')$ ssi $\exists$ (u,v) tel que :

$$|u| \neq 1 \text{ ou } |v| \neq 1 \text{ et } (u,v) \in \delta(X',Y')$$

Soit :

$$S(X,Y) = \{(u,v) \mid |u| \neq 1 \text{ ou } |v| \neq 1 \text{ et } \exists\, X',Y' \text{ telles que : } (X,Y) \overset{*f}{\rightarrow} (X',Y') \text{ et } (u,v) \in \delta(X',Y')\}$$

et

$$S'(X,Y) = \{(X',Y') \mid (X,Y) \rightarrow (X',Y')\}$$

alors δ' est obtenu à partir de δ de la manière suivante :

$$\delta'(X,Y) = \delta(X,Y) \setminus S'(X,Y) \cup S(X,Y)^{(1)}$$

Si $\nexists$ (X',Y') tel que $(X,Y) \overset{*f}{\rightarrow} (X',Y')$, ceci implique que l'a.p.c A boucle sur une suite de mouvement d'analyse stationnaire, ceci implique aussi $S(X,Y) = \emptyset$ et $S'(X,Y) = \delta(X,Y)$ d'où :

$$\delta'(X,Y) = \delta(X,Y) \setminus S'(X,Y) = \emptyset$$

l'a.p.c A' bloquera donc là ou A bouclait.

Notons que cette transformation est telle que si A est déterministe, il en est de même pour A'. En effet si A est déterministe on a :

$$|S'(X,Y| \leq 1 \text{ et } |S(X,Y)| \leq 1$$

soit si $|S'(X,Y)| = 1,\ (X',Y') \in S'(X,Y)$

et si $|S(X,y)| = 1,\ (u,v) \in S(X,Y)$

il vient :

$$\delta'(X,Y) = \delta(X,Y) \setminus \{(X',Y')\} \cup \{(u,v)\} = \{(u,v)\}$$

(1) Nous conviendrons que l'opérateur $\setminus$ a priorité sur U.

A' est donc déterministe.

Par ailleurs : $\Sigma' = \Sigma$, $\Gamma = \Gamma$, $M' = M$ et $F = F$ alors $A' = (\Sigma', \Gamma', \delta', M', F')$ est équivalent à A et vérifie 2).

Nous laissons le soin au lecteur de vérifier que $T(A) = T(A')$ et de constater que A' est nécessairement du même type que A. De plus les ensembles $T_1(A)$ et $T_2(A)$ sont inchangés dans le cas des n.e.d.

o Nous allons à présent construire A' à partir de A vérifiant les points 1 et 2, de manière à ce que A' vérifie 1, 2 et 3.

Soit $A' = (\Sigma', \Gamma', \delta', M', F')$.
Soit N un nouveau symbole $N \notin \Sigma \cup \Gamma \cup M$ il vient $\Gamma' = \Gamma \cup \{N\}$ et δ' est obtenu à partir de δ de la manière suivante :

o Si $(\Lambda,\Lambda) \in \delta(X,Y)$ alors :

1 Si $X \notin T_r(\#_g)$ $\delta'(X,Y) = \delta(X,Y) \setminus \{(\Lambda,\Lambda)\} \cup \{(\Lambda,N)\}$
2 Si $X \in T_r(\#_g)$ $\delta'(X,Y) = \delta(X,Y) \setminus \{(\Lambda,\Lambda)\} \cup \{(N,\Lambda)\}$

o Si $(\Lambda,\Lambda) \notin \delta(X,Y)$ alors :

$$\delta(X,Y) = \delta'(X,Y)$$

o $\forall X \in \Gamma \cup \Sigma \cup M$

$$\delta'(M,X) = (\Lambda,X) \text{ et } \delta'(X,N) = (X,\Lambda)$$

par ailleurs $\Sigma' = \Sigma$, $M' = M$.

Il est immédiat dans ces conditions que si A vérifie 1 et 2 alors A' vérifiera 1, 2 et 3. De plus $T(A) = T(A')$ et il y a conservation des familles $T_1(A)$ et $T_2(A)$. Nous laissons le soin au lecteur de le vérifier.
Ici encore A' est du même type que A.

Remarque :

On peut faire la remarque suivante si A se trouve dans la configuration K :

$$K = [Y,X]$$

et si

$$\delta(Y,X) = (\Lambda,\Lambda)$$

la construction indique alors : $\delta'(Y,X) = (M,\Lambda)$ car $Y \in T_r(\#_g)$ donc A' bloque dans la configuration

$$K'_A = [M,\Lambda]$$

alors que A bloque dans la configuration

$$K_A = [\Lambda,\Lambda]$$

K_A n'est en aucun cas une configuration finale de A, pas plus que K'_A.

d) Nous allons maintenant montrer une propriété qui nous sera utile dans le chapitre III (III.3).

Propriété.

Tout n.e.d (d) A est équivalent à un n.e.d (d) A' vérifiant : pour tout $X,Y,X',Y' \in \Gamma' \cup \Sigma' \cup M'$

$$(X'Y',\Lambda) \in \delta(X,Y) \Longrightarrow X' = X$$

de plus $T_1(A) = T_1(A')$ et $T_2(A) = T_2(A')$.

Démonstration.

Nous supposons que A est simplifié. Soit $A' = (\Sigma', \Gamma', M', \delta', F')$ tel que :

. $\Sigma' = \Sigma$, $M' = M$, $F' = F$

. $\Gamma' = \Gamma \cup (\Gamma \cup \Sigma)^2$

les éléments de $(\Gamma \cup \Sigma)^2$ seront notés $[X,Y]$.

Alors δ' est telle que :

a) Si $(X'Y',\Lambda) \in \delta(X,Y)$ alors :

a1	Si $X' = X$	$(XY',\Lambda) \in \delta'(X,Y)$
a2	Si $X' \neq X$	$(X[X',Y'],\Lambda) \in \delta'(X,Y)$
a3	$\forall Z \in \Gamma$	$([Z,X][X',Y'],\Lambda) \in \delta'([Z,X], Y)$
a4	$\forall Z, Z' \in \Gamma$	$([Z,Z'] [X',Y'],\Lambda) \in \delta'([Z,Z'], [X,Y])$
a5	$\forall Z \in \Gamma$	$(Z[X',Y'],\Lambda) \in \delta'(Z,[X,Y])$

b) Si $(X',\Lambda) \in \delta(X,Y)$ alors :

b1	$(X',\Lambda) \in \delta'(X,Y)$	
b2	$\forall Z \in \Gamma$	$([Z,X'],\Lambda) \in \delta'([Z,X], Y)$
b3	$\forall Z \in \Gamma$	$(X',\Lambda) \in \delta'(Z,[X,Y])$
b4	$\forall Z, Z' \in \Gamma$	$([Z,X'],\Lambda) \in \delta'([Z,Z'], [X,Y])$

c) Si $(\Lambda,X') \in \delta(X,Y)$ alors :

c1	$(\Lambda,X') \in \delta'(X,Y)$	
c2	$\forall Z \in \Gamma$	$(\Lambda,[Z,X]) \in \delta([Z,X],Y)$
c3	$\forall Z \in \Gamma$	$(\Lambda,X') \in \delta(Z,[X,Y])$
c4	$\forall Z, Z' \in \Gamma$	$(\Lambda,[Z,X']) \in \delta([Z,Z'], [X,Y])$

Il est immédiat que A' est simplifié et possède la propriétée. Notons de plus que si A est déterministe, A' l'est aussi. En effet l'existence d'une quelconque des règles de A' implique l'existence dans A d'une unique règle, par construction même de δ'.

Il nous reste à montrer que $T(A) = T(A')$.
Pour cela nous définissons une correspondance entre les configurations de A et celle de A'.

Nous dirons que K_A correspond à $K_{A'}$. Nous noterons $K_A \longleftrightarrow K_{A'}$.

Si nous sommes dans un des cas suivants :

$$K_A = [X_1X_2..X_n, \; Z \; a_1a_2..a_p \; \#^o_d]$$

pour tout i, $X_i \in \Gamma \cup \Sigma \cup M$, $a_i \in \Sigma$, $A \in \Gamma \cup \Sigma$, $Z \in \Gamma \cup \Sigma' \cup M$

1 - $K_{A'} = K_A$

2 -

21 $\exists i \quad 0 \leqslant i \leqslant n-3$ et $3 \leqslant n$

$$K_{A'} = [X_1X_r..X_i \; A \; [X_{i+1},X'_{i+1}][X_{n-1},X_n], \; Za_1..a_p \; \#^o_d]$$

22 pour $2 \leqslant n$

$$K_{A'} = [X_1...X_{n-2} \; A[X_{n-1},X_n], \; Z \; a_1... \; a_p \; \#^o_d]$$

3 -

31 $\exists i \quad 0 \leqslant i \leqslant n-2$ et $n \geqslant 2$

$$K_{A'} = [X_1 \; ... \; X_i \; A \; [X_{i+1},X'_{i+1}] \; ... \; [X_{n-1},X'_{n-1}],[X_n,Z] \; a_1 \; ... \; a_p \; \#^o_d]$$

32 $n \geqslant 1$

$$K_{A'} = [X_1 \; ... \; X_{n-1} \; A, \; [X_n,Z] \; a_1 \; ... \; a_p \; \#^o_d]$$

De cette définition nous pouvons énoncer les points suivants :

pt 1 Dans $K_{A'}$, tout élément A de $\Gamma \cup \Sigma$ placé immédiatement à gauche d'un élément de $(\Gamma \cup \Sigma)^2$ n'a pas de signification car il n'existe pas dans la configuration K_A correspondante.

pt 2 Dans $K_{A'}$: tout élément $[X_i,X'_i]$ de $(\Gamma \cup \Sigma)^2$ placé sur le pile gauche, non au sommet, ou au sommet et rencontrant un élément de $(\Gamma \cup \Sigma)^2$, s'intréprête comme suit :

X_i est en fait l'élément qui le précéde dans K_A

X'_i n'a pas de signification car il n'existe pas dans la configuration correspondante K_A.

pt 3 Dans $K_{A'}$: tout élément de la forme $[X,Y]$ de $(\Gamma \cup \Sigma)^2$ placé en sommet de pile gauche et rencontrant un élément de $\Gamma \cup \Sigma'$, donne dans K_A

X est l'élément qui précéde

Y est en fait le sommet de la pile gauche.

pt 4 Dans $K_{A'}$: tout élément de la forme $[X,Y]$ placé en sommet de pile droite signifie qu'un fait X et Y sont respectivement les sommets de pile gauche et droite dans A.

Ces quatre points suffisent partant d'une configuration de A' pour obtenir une configuration correspondante de A et réciproquement. Notons qu'une configuration de A' correspond à une seule configuration de A.

Exemple.

$$
\begin{tikzcd}
1 - \quad K_A = [& X & X_1 & X_2 & X_3 & X_4 & , & Z & a_1 & \#_d &] \\
K_{A'} = [& X & A & {[X_1,X'_1]} \arrow[ul, dashed, "4"] & {[X_2,X'_2]} \arrow[ul, dashed, "3"] & {[X_3,X_4]} \arrow[ul, dashed, "2"] \arrow[u, dashed, "1"'] & , & Z & a_1 & \#_d &]
\end{tikzcd}
$$

les flèches 1 et 2 indique le respect du point 3.

les flèches 3 et 4 indique le respect du point 2.

A ne correspond à rien dans K_A, respect du point 1.

2 -

$$\begin{tikzcd}[column sep=small]
K_A = [& X & X_1 & X_2 & X_3 & X_4 & , & Z & a_1 & \#_d &] \\
K_{A'} = [& X & A & {[X_1,X'_1]} \arrow[ul, dashed, "5"'] & {[X_2,X'_2]} \arrow[ul, dashed, "4"'] & {[X_3,X'_3]} \arrow[ul, dashed, "3"'] & , & {[X_4,Z]} \arrow[ull, dashed, "2"'] \arrow[u, dashed, "1"'] & a_1 & \#_d &]
\end{tikzcd}$$

les flèches 1 et 2 respectent le point 4.
les flèches 3,4,5 respectent le point 2.

A ne correspond à rien dans K_A, respect du point 1.

Le lecteur vérifiera aisément que les règles de A' vérifie les 4 points énoncés plus haut, en effet :
on suppose que le couple auquel s'applique δ' vérifie les points 1, 2, 3 et 4, nous dirons que ce couple (X,Y) est le couple de départ, et on constate alors que $\delta'(X,Y)$ vérifie les quatre points.

On peut donner le tableau suivant :

Règle envisagée	Points vérifiés par le couple de départ (X,Y)	Points vérifiés par Ω $\Omega \in \delta'(X,Y)$
a2	-	1,3
a3	3	2,3
a4	2,4	2,3
a5	1,4	1,4
b2	3	3
b3	1,4	-
b4	2,4	3
c2	3	1,4
c3	1,4	-
c4	2,4	1,4

Ceci nous permet d'affirmer qu'il y a équivalence, pour toutes configurations initiales identiquesdans A et dans A' $K^o_A <-> K^o_{A'}$ entre

1 - $\exists\ (K^i_A)^n_{i=1},\ K^o_A \vdash K'_A \vdash \dots \vdash K^n_A$

2 - $\exists\ (K^i_{A'})^n_{i=1},\ K^o_{A'} \vdash K'_{A'} \vdash \dots \vdash K^n_{A'}$

avec pour tout i, $K^i_A <—> K^i_{A'}$.

Il suffit alors de remarquer que les configurations finales de A' et de A sont les mêmes, en effet les correspondances ne sont définies que pour $n \geq 1$ et $Z \neq \Lambda$.

Donc si $K_A = [\Lambda,S]$, la seule configuration qui lui corresponde et $K_A' = K_A = [\Lambda,S]$ par 1, de même si $K_A = [S,\Lambda]$.

Ceci nous permet de dire que : $T(A) = T(A')$ et que $T_1(A) = T_1(A')$ et $T_2(1) = T_2(A')$.
<u>cqfd</u>.

e) <u>Propriété concernant les automates à piles de mémoire déterministes.</u>

Tout automate à pile de mémoire déterministe est équivalent à un automate à pile de mémoire déterministe : $M = <\Sigma, \Gamma, Q, \delta, q_o, Q_F, Z_o>$

1 - pour tout $q, q' \in Q$, $Z, Z' \in \Gamma$, $\gamma \in \Gamma^*$, $a \in \Sigma^o$

$\delta(q,Z,a) = (q',\gamma Z') \Longrightarrow \delta(q',Z',\Lambda) = \emptyset$

2 - $\delta(q_o, Z_o,\Lambda) = \emptyset$

3 - $\forall q \in Q_F$, $\delta(q,Z_o,\Lambda) = \emptyset$ ou $\{q',\Lambda\}$

De plus :

$$T_o(M) = T_o(M'),\ T_1(M) = T_1(M')$$

et si

$$T_2(M) \neq \{\Lambda\}, \qquad T_2(M) = T_2(M')$$

Démonstration.

Soit l'a.p.m.d $M = \langle \Sigma, \Gamma, Q, \delta, q_o, Q_F, Z_o \rangle$.

Nous allons construire à partir de M, l'a.p.m.d $M' = \langle \Sigma, \Gamma, Q', \delta', q_o, Q'_F, Z_o \rangle$ vérifiant la propriété.

Point 1 de la propriété.

Donnons tout d'abord l'idée intuitive de la démonstration. Après avoir lu Λ ou une lettre $a \in \Sigma$ sur le mot en empilant $\delta Z'$, l'automate M étant déterministe on peut anticiper les mouvements qui suivent par lecture de Λ, jusqu'à ce que soit, on ait idée $\gamma Z'$ de la pile, soit, qu'on ait empilé $\bar{\gamma}\bar{Z}$ en x pouvant dans un état $\bar{q}$, tel que $\delta(\bar{q}, \bar{Z}, \Lambda) = \emptyset$.
De cette manière on peut toujours s'assurer qu'après avoir fait un mouvement en empilant, il n'y aura pas de mouvement par lecture de Λ, ces mouvements ayant été anticipés.

Un problème est alors de distinguer les cas où M dans ses mouvements par lecture de Λ passe par un état final simplement pour Δ_o, passe par un état final avec Z_o sur la pile par Δ_1, ou en vidant la pile pour Δ_2.

Un autre problème est de distinguer les cas où M boucle par lecture de Λ.

Nous présentons à présent la construction des éléments de M' :

Rappelons une notation employée dans [8]

$$(q, \gamma Z, w) \overset{*}{\underset{d}{\vdash}} (q', \gamma' Z, \Lambda) \text{ si et seulement si :}$$

$$\text{(i)} \quad (q, \gamma Z, w) \overset{*}{\vdash} (q', \gamma' Z', \Lambda)$$

$$\text{(ii)} \quad \delta(q, Z', \Lambda) = \emptyset.$$

A partir de ceci définissons la relation $\overset{*}{\underset{d'}{\vdash}}$

$(q, \gamma_2, w) \overset{*}{\underset{d'}{\vdash}} (q', \gamma' Z, \Lambda)$ si et seulement si soit $(q, \gamma Z, w) \overset{*}{\underset{d}{\vdash}} (q', \gamma' Z', \Lambda)$ soit $\gamma' Z' = \Lambda$.

Dans ces conditions soit :

$$A = \{\bar{\gamma} \;/\; \text{pour } a \in \Sigma^0 \quad \exists \delta(q,z,a) = (q',\ \gamma' Z')$$

$$\text{et } [q',\ \gamma' z',\ \Lambda] \overset{*}{\underset{d'}{\vdash}} [\bar{q},\ \bar{\gamma},\Lambda]\}$$

et

$$Q_A = \{[q,\gamma] \;/\; q \in Q,\ \gamma \in A\}$$

nous obtenons alors Q' et Q'_F ; pour $e \notin Q$ et $pt \notin Q$:

$$Q' = Q \cup \{e,pt\} \cup \{q_f/q \in Q\} \cup Q_A$$

$$Q'_F = \{q_f/q \in Q\} \cup \{e\} \cup Q_A.$$

Cette construction de Q' et Q'_F va trouver sa justification dans la construction de δ', que nous donnons maintenant :

Soit trs une application de $Q \times \Gamma^* \to Q' \times \Gamma^*$ définie comme suit :

A - Pour tout $(q',\ \gamma' z')$ telle que $\delta(q',\ z',\ \Lambda) \neq \emptyset$

A1 - Si $\nexists\ (q',\ \gamma' z',\ \Lambda) \overset{*}{\underset{d'}{\vdash}} (\bar{q},\ \bar{\gamma},\ \Lambda)$

A11 - Si $\exists\ (q',\ \gamma' z',\ \Lambda) \overset{*}{\vdash} (q_F,\ \gamma,\ \Lambda)$
alors trs $(q',\ \gamma' z') = (e,\gamma)$

A12 - Sinon :

trs $(q',\ \gamma' z') = (pt,\ Z')$

A2 - Si $\exists\ (q',\ \gamma' z',\ \Lambda) \overset{*}{\underset{d'}{\vdash}} (\bar{q},\ \bar{\gamma},\ \Lambda)$

A21 - Si $\exists\ (q',\ \gamma' z',\ \Lambda) \overset{*}{\vdash} (q_F,\ \gamma,\ \Lambda)$ alors :

si $\gamma \neq Z_o$ trs $(q',\ \gamma' z') = (\bar{q}_f,\ \bar{\gamma})$ (1)
si $\gamma = Z_o$ trs $(q',\ \gamma' z') = ([\bar{q},\bar{\gamma}],\ z_o)$ (2)

A22 - Sinon :

trs $(q',\ \gamma' z') = (\bar{q},\bar{\gamma})$.

Alors δ' est obtenu à partir δ de la manière suivante :

Si $\delta(q, z, a) = (q', \gamma' z')$

o Si $\delta(q', z', \Lambda) \neq \emptyset$

$\delta'(q, z, a) = \delta'(q_f, z, a) = trs(q', \gamma' z')$ (3)

$\delta'([q, \gamma z], z_o, a) = (\bar{q}, \gamma\bar{\gamma})$ si $tr(q', \gamma' z') = (\bar{q}, \bar{\gamma})$ (4)

o Si $\delta(q', z', \Lambda) = \emptyset$

$\delta'(q, z, a) = \delta(q, z, a)$

o Si $\delta(q, z, a) = \emptyset$

$\delta'(q, z, a) = \emptyset$

Il reste à vérifier que

1 - M' possède la propriété 1

2 - $T_o(M) = T_o(M')$, $T_1(M) = T_2(M')$ et si

$T_2(M) \neq \{\Lambda\}$ $T_2(M) = T_2(M')$.

Le point 1 est facile à vérifier.

Le point 2 se vérifie également facilement. Notons que dans le cas ou M boucle, nous faisons passer M' dans un état final e si M passe par un état final au cours de son mouvement (A11). Aucun mouvement n'étant plus possible à partir de l'état e. Si M ne passe pas par un état final, nous faisons passer M' dans l'état pt qui est un état puit non final (A12). Dans le cas (A11) en même temps que nous passons dans l'état e nous empilons γ, de sorte que si $\gamma = Z_o$ ceci assure $T_1(M) = T_1(M')$. Nécessairement $\gamma \neq \Lambda$ car $\nexists$ $(q', \gamma z', \Lambda) \xrightarrow[d']{*} (\bar{q}, \bar{\gamma})$, nous préservons donc également $T_2(M) = T_2(M')$.

Dans le cas ou M ne boucle pas. Si au cours de son mouvement par lecture de Λ. M passe par un état final alors : si à ce moment la pile est réduite à Z_o, il faut en tenir compte afin de préserver $T_1(M) = T_1(M')$, c'est ce qui est fait en (2) ou nous mémorisons également dans l'état le contenu de la pile à la fin du mouvement par lecture de Λ, afin de préserver une éventuelle suite au cas où le mot d'entrée ne serait pas entièrement lu.

Dans ce cas $[q,\gamma] \in Q'_F$. Notons que (4), assure une éventuelle suite par lecture d'une lettre de Σ'.

Si maintenant passant par un état final $\gamma \neq Z_o$, il suffit de faire passer M' dans un état final $\bar{q}f$, en empilant le contenu de la pile à la fin du mouvement dans M. La suite éventuelle par lecture de lettre de Σ est assurée à la ligne (3).

Ici la présentation de $T_2(M)$ est automatique. En effet si M passe par un état final en vidant la pile, c'est que nécessairement : $\bar{q} = q_F$ et $\bar{\gamma} = \Lambda$ par définition de $\vdash_{\overline{d'}}$ et étant donné que M ne peut plus faire de mouvement si sa pile est vide : la ligne (1) suffit donc à assurer $T_2(M) = T_2(M')$.

Remarquons enfin qu'à la ligne (3) on ne peut avoir $tr(q', \gamma z') = ([\bar{q},\bar{\gamma}], Z_o)$ car $\nexists$ v $\in \Gamma^*$ tel que $z' = Z_o v$, nous ne pouvons donc nous trouver dans le cas de la ligne (2). Z_o marquant le fond de pile et ne pouvant se trouver à l'intérieur de la pile. Ceci est très important car on aurait

$$\delta'([q,\gamma z], Z_o, a) = ([\bar{q}, \gamma\bar{\gamma}], Z_o, a)$$

et Q' ne serait pas nécessairement un ensemble fini.

<u>cqfd</u>.

CHAPITRE II

RELATIONS EXISTANTES ENTRE LES AUTOMATES A PILES COMMUNICANTES ET LES PRINCIPAUX AUTOMATES CONNUS

II.1 INTRODUCTION

Il s'agit dans ce chapitre de montrer l'équivalence qui existe entre les diverses automates à piles communicantes et les automates classiques.

Nous allons procéder d'une manière assez analogue pour chacune des parties de ce chapitre :

- Rappel des définitions concernant l'automate classique
- Définition d'automate associé , soit l'a.p.c associé à l'automate classique soit l'inverse
- Définition de ce que nous appellerons les configurations correspondantes, cette définition met en rapport les configurations des deux automates considérés
- Lemme des "mouvements correspondants" il montre que les deux automates peuvent "se suivre" aux travers de configurations correspondantes
- Théorème de "l'inclusion"

l'élément important est à chaque fois la construction de l'automate associé, et les résultats sont toujours des conséquences directes de la construction. Nous ne mènerons donc pas toujours, les démonstrations à leur terme , celles-ci faisant appel à des procédés classiques sans intérêts pour eux mêmes. Nous laissons au lecteur le soin et le plaisir de les achever dans leurs moindres détails.

II.2 AUTOMATES A PILES COMMUNICANTES ET MACHINE DE TURING

a) Donnons tout d'abord la définition d'une machine de Turing : MT

Définition.

Une machine de Turing est un quadruplet $<\Sigma, Q, \delta, q_o>$

Σ est un alphabet fini

Q est un ensemble fini appelé ensemble des états

δ est une application de $Q \times (\Sigma \cup \{B\})$ dans les parties de $Q \times (\Sigma \cup \{B\}) \cup Q \times \{-1,+1\}$ ou B est un symbole spécial appelé blanc, on notera : $\Sigma_B = \Sigma \cup \{B\}$

$q_o \in Q$, état distingué appelé état initial.

Une configuration K d'une machine de Turing est un mot tel que :

$$K \in (\Sigma\Sigma_B^* \cup \{\Lambda\})\ Q\ (\Sigma_B'^*\Sigma \cup \{\Lambda\})$$

<u>exemple</u> : $K = uqv$

Nous dirons que K_2 dérive directement de K_1 nous noterons :

$$K_1 \vdash K_2$$

dans chacun des cas suivants :

1 - $K_1 = uqav \quad K_2 = uq'a'v$

Si $(q',a') \in \delta(q,a)$

2 - $K_1 = uqav \quad K_2 = uaq'v$

si $(q',+1) \in \delta(q,a)$

3 - $K_1 = abqav \quad K_2 = uq'bav$

si $(q',-1) \in \delta(q,a)$

4 - $K_1 = qav \quad K_2 = q'Bav$

si $(q',-1) \in \delta(q,a)$

5 - $K_1 = uq \quad K_2 = uq'a$

si $(q',a) \in \delta(q,B)$

6 - $K_1 = uaq \quad K_2 = uq'a$

si $(q',-1) \in \delta(q,B)$

7 - $K_1 = uq \quad K_2 = uBq'$

si $(q',+1) \in \delta(q,B)$

On désigne par $\overset{*}{\longmapsto}$ la clôture reflexive et transitive de $\longmapsto$. Nous pouvons maintenant donner la définition du langage accepté par une machine de Turing.

Définition.

Nous noterons T(MT) le langage accepté par une machine de Turing MT :

$$T(MT) = \{w \in \Sigma^* \;/\; \exists q \in Q,\ u,\ v \in \Sigma_B^*,\ x \in \Sigma_B \text{ tels que } q_o w \overset{*}{\longmapsto} uqxv \text{ et } \delta(q,x) = \emptyset\}$$

b) Nous allons montrer que l'on peut associer à toute machine de Turing M un automate à piles communicantes qui reconnaisse le même langage que M'.

Nous donnons d'abord la construction de A à partir de M, nous commenterons cette construction après.

Définition.

Soit une machine de Turing $M = \langle\Sigma,\ Q,\ \delta,\ q_o,\ Q_F\rangle$ nous dirons qu'un automate à piles communicantes $A = \langle\Gamma_A,\ \Sigma_A,\ M_A,\ \delta_A,\ F_A\rangle$ est associé à MT si :

1 - $\Gamma_{AA} = Q \cup \{q_\# \;/\; q \in Q\} \cup \{x_\# \;/\; x \in \Sigma\} \cup \{B\}$

2 - $\Sigma_A = \Sigma$

3 - $M_A = \{\#\}$ c'est-à-dire $\#_g = \#_d = \#$

4 - $F_A = \{Ax\}$

5 - δ_A est obtenu à partir de δ de la manière suivante :

pour tout $X \in \Sigma_A' \cup \{B\}$, tout $q \in Q$

a - Si $(-1,q') \in \delta(q,X)$ alors :

a1 $(\Lambda,q'x) \in \delta_A(q,X)$
a2 $(q_\#',Bx) \in \delta_A(q_\#,X)$
de plus si X = B (B étant le blanc).

a3 $(\Lambda, q'\#) \in \delta_A(q,\#)$

a4 $(q'_\#\ ,\ B\#) \in \delta_A(q_\#,\#)$

b - Si $(+1,q') \in \delta(q,X)$

b1 $(Xq',\Lambda) \in \delta_A(q,X)$

b2 $(X_\# q',\Lambda) \in \delta_A(q_\#,X)$

De plus si $X = B$

b3 $(q',B\#) \in \delta_A(q,\#)$

b4 $(q'_\#,\ B\#) \in \delta_A(q_\#,\#)$

c - Si $(X',q') \in \delta(q,X)$

c1 $(q',X') \in \delta_A(q,X)$

c2 $(q'_\#,\ X') \in \delta_A(q_\#,X)$

De plus si $X = b$

c3 $(q',X'\#) \in \delta_A(q,\#)$

c4 $(q'_\#,\ X'\#) \in \delta_A(q_\#,\#)$

d - $\forall x \in S$

d1 $\{(q,X)\} = \delta_A(X,q)$

d2 $\{(q\ \ ,X)\} = \delta_A(X_\#,\ q)$

e -

e1 si $(-1,q') \in \ (q_o,X)$	alors $(q'_\#\ ,BX) \in \delta_A(\#,X)$
e'1 de plus si $X = B$	alors $(q'_\#\ ,B\#) \in \delta_A(\#,\#)$
e2 si $(+1,q') \in \ (q_o,X)$	alors $(X\ q',\Lambda) \in \delta_A(\#,X)$
e'2 de plus si $X = B$	alors $(q'_\#\ ,B\#) \in \delta_A(\#,\#)$
e3 si $(X',q') \in \delta(q_o,X)$	alors $(q'_\#\ ,X') \in \delta_A(\#,X)$
e'3 de plus si $X = B$	alors $(q'_\#,X'\#) \in \delta_A(\#,\#)$

f Si $\delta(q,X) = \emptyset$ alors $\delta_A(q,X) = \{(A_x,\Lambda)\}$

et $\delta_A(q_\#,X) = \{(\Lambda,A_x)\}$

de plus si $X = B$ alors $\delta_A(q,\#) = \{(\Lambda,A_x)\}$

et $\delta_A(q_\#,\#) = \{(\Lambda,A_x)\}$

g $\forall X \in \Gamma_A$

$$\delta(A_x,X) = \{(A_x,\Lambda)\}$$

$$\delta(A_x,\Lambda) = \{(\Lambda,A_x)\}$$

$$\delta(x,A_x) = \{(\Lambda,A_x)\}$$

c) Nous allons à présent donner la définition de ce que nous appellerons des configurations correspondantes.
Définition qui est cohérente avec la construction.

Définition.

Configurations correspondantes et configurations de transitions. Nous dirons que deux configurations K_M dans M et K_A dans A sont correspondantes si l'on se trouve dans un des cas suivants : Nous noterons $K_M \longleftrightarrow K_A$ pour tout $\alpha \in \Sigma$, $q \in Q/\{q_o\}$, $w \in \Sigma^*$, $w' \in \Sigma'^*$.

0 $K_M = q_o w$, $K_A = [\#,w\#]$

1 $K_M = \alpha w q w'$, $K_A = [\alpha_\# w q, w'\#]$

2 $K_M = q w'$, $K_A = [q_\#, w'\#]$

Une configuration de A sera dite de transition si elle est d'une des formes suivantes pour $q \in Q$, w, $w' \in \Sigma_B^*$

$$- [w,qw'\#] \ - [wAx,w'] \ - [w,A_x]$$

d) Nous avons alors le lemme :

Lemme.

Pour tout couple de configurations correspondantes K_A dans A et K_M dans M, il y a équivalence entre

1 - $\exists K'_M$ telle que $K_M \vdash K'_M \Longrightarrow \exists K'_A$ tel que

$$K'_A \leftrightarrow K'_M \text{ et } K_A \overset{n}{\vdash} K'_A \text{ avec } n=1 \text{ ou } n=2.$$

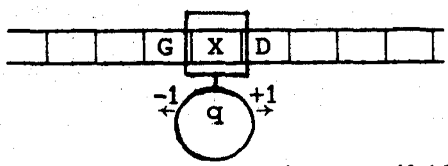
2 - $\exists K'_A$ tel que K'_A ne soit pas de transition et

$K_A \overset{n}{\vdash} K'_A$ n=1 ou 2 $\Longrightarrow \exists K'_M$ telle que $K'_M \longleftrightarrow K'_A$
et $K_M \vdash K'_M$

<u>Preuve</u>.

On a fait en sorte que l'a.p.c soit capable de "suivre" dans ses mouvements la machine de Turing. Dans les deux cas pour décider du mouvement à effectuer nous avons la connaissance de 2 éléments qui sont l'état et le symbole écrit sur la bande pour la machine de Turing. Nous conviendrons de faire figurer l'état au sommet de la pile gauche, et le symbole de la bande de lecture sur la pile droite dans l'a.p.c associé à M.
Nous pouvons alors nous trouver dans les situations que nous schématisons d'une manière classique de la façon suivante.

A - Dans M



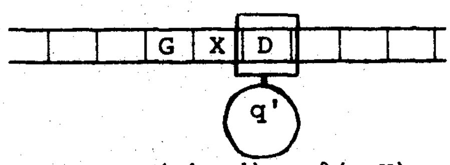
Dans A

[-Gq,XD-]

dans M nous avons trois possibilités de mouvement qu'il faut simuler dans A

A1 - Mouvement sur la droite avec changement d'état

Dans M



$(+1,q') \in \delta(q,X)$

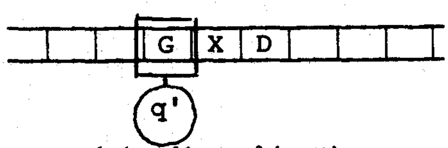
Simulation dans A

[-GXq',D-]

donc : $(Xq',\Lambda) \in \delta_A(q,X)$ (voir (b))

A2 - Mouvement sur la gauche avec changement d'état

Dans M



$(-1,q') \in \delta(q,X)$

Simulation dans A en 2 temps, en passant par une configuration de transition.

[-G,q'XD-] $\vdash$ [-q',GXD-]

donc $(\Lambda,q'X) \in \delta_A(q,X)$ (voir(a))

pour le premier mouvement et

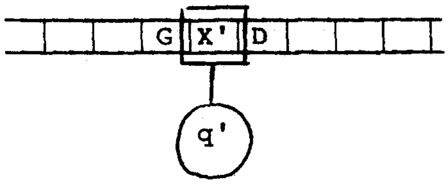
$(q',G) = \delta_A(G,q')$ (voir(d))

pour le deuxième mouvement

A3 - Pas de mouvement, changement de la lettre sur la bande et de l'état.

Dans M

Simulation dans A



$(X',q') \in \delta(q,X)$

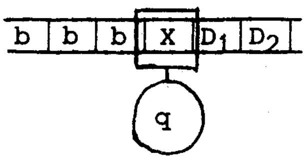
$[-Gq',X'D-]$

donc : $(q',X') \in \delta(q,X)$ (voir (c))

Nous remarquons que dans aucun de ces trois cas nous avons employé de mouvements croissants dans A. Nous allons voir deux cas où ils sont indispensables.

Supposons que M et A se trouvent dans la configuration suivante :

B -

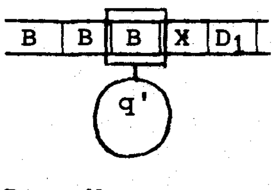


$[q_{\#}, XD_1D_2-]$

Dans M

Configuration associée dans A

Il est facile de voir que de telles configurations sont atteintes par éventuellement une suite de mouvements (A2)
faisons faire dans M un mouvement sur la gauche il vient :



$[q'_{\#}, BXD_1-]$

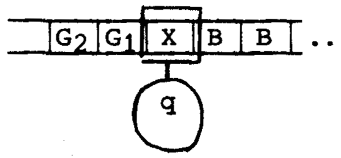
Dans M

Dans A

soit $(-1,q') \in \delta(q,X) \Longrightarrow$ par (a2) $(q'_{\#}, BX) \in \delta_A(q_{\#},X)$

De même pour le cas :

C -

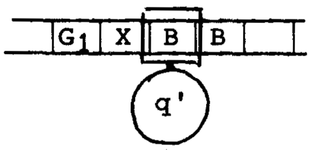


$[-G_2G_1q,X\#]$

Dans M

Dans A

Faisons faire ici un mouvement sur la droite.



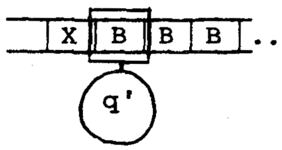
$[-G_1Xq,\#]$

soit $(+1,q') \in \delta(q,X)$

soit $(Xq,\Lambda) \in \delta_A(q,X)$ (voir (b))

On constate donc que le symbole # doit ici être lu comme étant un symbole blanc B (voir les règles a3, a4, b3, b4, c3, c4, e'1, e'2, e'3).

Si M effectue à nouveau un mouvement sur la droite il vient



$[-G_1Xq',B\#]$

soit $(+1,q') \in \delta(q,B)$

soit $(q', B\#) \in \delta_A(q,\#)$ (voir b3)

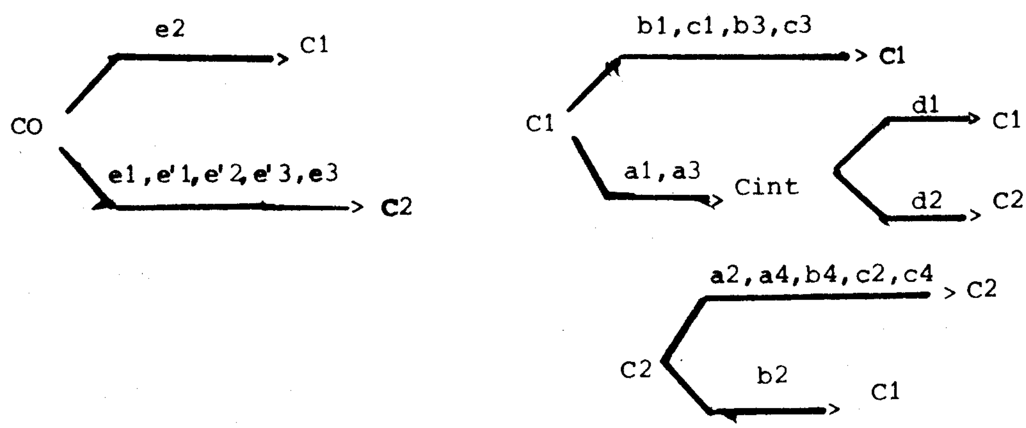
ici encore nous avons utilisé un mouvement croissant.

Nous avons procédé par configurations correspondantes dans M et dans A, la construction de A est étudiée pour que partant des configurations initiales A puissent suivre M, les deux automates ayant des évolutions parallèles au travers de configurations correspondantes. Plus précisément :

Le schéma de la démonstration, qui ne présente pas de difficultés mais qui comporte un certains nombre de cas à envisager en plus de ceux déjà vus est le suivant. Nous dirons que deux configurations sont ci, $o \leq i \leq Z$.
Si l'une dans M et l'autre dans A sont correspondantes d'après le point i de la définition. Nous allons exhiber tous les passages possibles de 2 configurations ci vers 2 configurations cj, en indiquant les règles ci de la construction qui ont permis le passage, nous noterons ci $\xrightarrow{x_i}$ cj.

Pour cela nous écrirons, Cint (configuration intermédiaire) toutes configurations dans A du type [w,qw'], c'est-à-dire ayant un élément appartenant à Q au sommet de la pile droite.

Nous obtenons les graphes suivants :



constatons que toutes les règles de la construction sauf f et g sont utilisées en effet ces règles permettent lorsque les deux automates se trouve dans deux configurations ci, et qu'aucun mouvement n' est plus possible dans M ce qui veut dire que M se trouve dans une configuration finale, d'amener A dans une configuration finale.

<u>cqfd</u>.

e) Un corollaire du lemme précédent

<u>Corollaire</u>.

A chaque mouvement menant dans M d'une configuration initiale à une configuration finale correspond dans A un mouvement partant de la configuration initiale correspondante et menant à une configuration finale et réciproquement.

Preuve.

Nous pouvons enoncer, comme une extention du lemme précédent pour tout $w \in \Sigma^*$ il y a équivalence entre :

1 - $\exists\ [q_o,w\#] \overset{*}{\vdash} [uq,av]$ dans M, $a \in \Sigma_B \cup \{\Lambda\}$

2 - $\exists \alpha \in \Sigma_B$ $u' \in \Sigma_B^*$ tels que $u = \alpha u'$ et :

$[\#,w\#] \overset{*}{\vdash} [\alpha\ _{\#}u'q,av\#]$ dans A ou $u = \Lambda$ et :

$[\#,w\#] \overset{*}{\vdash} [q_{\#},av]$ dans A

En effet si $[uq,av]$ est final dans M, c'est-à-dire si $\delta(q,a) = \emptyset$ nous avons d'après les règles (f), qui prévoient tout les cas possibles :

$$\delta_A(q,a) = (A_x,\Lambda).$$

Ceci plus les règles (g), assure dans A, dans les deux cas de 2, le cheminement vers la configuration $[\Lambda,A_x]$ qui est final dans A.

Réciproquement, il suffit de constater que :

$$\exists\ [\#,w\#] \overset{*}{\vdash} [\Lambda,A_x]$$

si et seulement si

$$\exists q \in Q,\ u,\ v \in \Sigma_B^* \text{ tels que : } \delta_A(q,{}^{(1)}v) = (A_x,\Lambda)^{(1)}$$

$$[\#,w\#] \overset{*}{\vdash} [uq,v\#] \overset{*}{\vdash} [\Lambda,A_x]$$

ce qui assure l'existence dans M du mouvement :

$$[q_o,w\#] \overset{*}{\vdash} [uq,v] \text{ avec } \delta(q,{}^{(1)}v) = \emptyset$$

cqfd.

(1) ${}^{(1)}v$ est la lettre la plus à gauche de v si $v \neq \Lambda$, et vaut Λ si $v=1$.

Ce qui nous permet d'énoncer le théorème, qui en est une conséquence directe.

Théorème.

La classe des langages reconnus par une machine de Turing est incluse dans la classe des langages reconnus par un a.p.c.

f) Nous pourrions aisément montrer, d'une manière identique la réciproque du théorème précédent. Ceci ne présentant aucun intérêt, nous évoquerons la Thèse de Church [17], pour affirmer la réciproque du résultat précédent. Nous pouvons donc énoncer le théorème.

Théorème.

La classe des langages reconnus par les a.p.c, coïncide avec la classe des langages récursivement énumérables.

II.3 AUTOMATE A PILES COMMUNICANTES BORNEES ET AUTOMATES LINAIREMENT BORNES

a) Donnons la définition d'un automate à mémoire linéairement borné abrégé M.L.B.

Définition.

Un automate linéairement borné M.L.B est un quintuplet M.L.B = $<\Sigma, Q, \delta, q_o, Q_F>$ où :

1 - Σ ensemble fini, alphabet des mots, contenant un symbole distingué #

2 - Q ensemble fini d'état

3 - δ application de : $Q \times \Sigma$ dans les parties de $Q \times \Sigma/\{\#\} \cup Q \times \{-1,+1\}$

4 - $q_o \in Q$ état distingué appelé état final

5 - $Q_F \subseteq Q$ ensemble des états finaux

Nous avons de plus sur δ les deux restrictions suivantes :

$\nexists q \in Q \ \nexists q' \in Q, \nexists a \in \Sigma$ tels que :

$$(q',-1) \in (q_o,a)$$

et $$(q',+1) \in (q,\#)$$

Une configuration d'un automate à mémoire linéairement bornée K est un mot de

$$\Sigma^* Q \Sigma^* \#^o \quad (1)$$

Nous dirons qu'une configuration K_2 dérive directement d'une configuration K_1. Dans les trois cas suivants :

1 - $K_1 = uqav \quad K_2 = uq'a'v$

si $(q',a') \in \delta(q,a)$

2 - $K_1 = ubqav \quad K_2 = uq'bav$

si $(q',-1) \in \delta(q,a)$

3 - $K_1 = uqav \quad K_2 = uaq'v$

si $(q',+1) \in \delta(q,a)$

<u>Remarque</u> :

D'après la règle 3) de la définition nous constations qu'éventuellement le marqueur # peut être effacé mais qu'il ne peut jamais apparaître ailleurs qu'en fin de mot.

De la même manière que pour la machine de Turing nous noterons $K_1 \vdash K_2$, et $\overset{*}{\vdash}$ la clôture reflexive et transitive de $\vdash$. Nous pouvons définir le langage reconnu par un M.L.B noté T(M.L.B).

$$T(M.L.B) = \{w \in \Sigma'^* / \exists u,v, \exists q \in Q' \text{ tels que : } q_o w\#d \overset{*}{\vdash} uqv\}$$

(1) $\#^o = \# \cup \{\Lambda\}$

Un automate à mémoire linéairement borné est une machine de Turing dont la "tête" est astreinte à se déplacer dans une zone de longueur égale au mot w qui est analysé, nous pouvons donc lui associer un automate à piles communicantes bornées, c'est-à-dire dont la largeur globale des 2 piles est décroissante au sens large. Notons que cette définition est en tout point équivalente à cette donnée dans [15]. Il est immédiat que l'on peut se passer de marqueur de début de mot dans le cas non déterministe comme dans le cas déterministe, nous nous passons de ce marqueur pour rendre plus claire la simulation d'un M.L.B par un a.p.c.b que nous abordons maintenant.

b) *Définition.*

Nous dirons qu'un automate à piles communicantes bornées $A = <\Gamma_A, \Sigma_A, M_A, \delta_A, F_A>$ est associé à un automate linéairement borné M.L.B $= <\Sigma, Q, \delta, q_o, Q_F>$ si

1 - $\Gamma_A = Q$

2 - $\Sigma_A = \Sigma$

3 - $M_A = \{q_o, \#_d\}$ c'est-à-dire $\#_g = q_o$ et $\#_d = \#$

4 - $F_A = \{A_x\}$

5 - δ_A est obtenu à partir de δ de la manière suivante :

A - pour tout $q \notin Q_F$

a) Si $(q',+1) \in \delta(q,a)$ alors :

$(aq,\Lambda) \in \delta_A(q,a)$

b) Si $(q',-1) \in \delta(q,a)$

$(\Lambda,qa) \in \delta_A(q,a)$

c) Si $(q',X) \in \delta(q,a)$

$(q',X) \in \delta_A(q,a)$

d) $\forall q$, $\forall a$

$$\delta_A(a,q) = \{(q,a)\}$$

B - Si $q \in Q_F$ alors :

$$(A_x,\Lambda) = \delta_A(q,x) \text{ si } x \notin T_r(\#_d)$$

$$(\Lambda,A_x) = \delta_A(q,x) \text{ si } x \in T_r(\#_d)$$

C - $\delta(A_x,x) = (A_x,\Lambda)$ si $x \notin T_r(\#_d)$

$$\delta_A(A_x,x) = (\Lambda,A_x) \text{ si } x \in T_r(\#_d)$$

$$\delta_A(x,A_x) = (\Lambda,A_x) \ \forall x$$

c) La considération de configurations que nous dirons correspondantes dans A et M éclairera le pourquoi de cette construction.-

Définition.

Nous dirons que deux configurations K_A dans et K_M dans M sont correspondantes si :

$$K_A = [uq,v] \text{ et } K_M = uqv$$

nous noterons $K_A \longleftrightarrow K_M$.

Un cas particulier sera le suivant :

$$[q_o,w\#_d] \longleftrightarrow q_o w\#_d$$

la correspondance est ici particulièrement simple.

Les règles a) et c) permettent de manière tout à fait claire à l'a.p.c.b de suivre le M.L.B, de même que les règles b) associées à d).

Les règles B- et C- permettent à l'a.p.c.b de passer dans une configuration finale lorsque le M.L.B est dans une configuration finale est ceci d'une manière simple.

Il est tout aussi évident qu'il existe une correspondance entre les mouvements dans A et dans le MLB.

d) D'où le théorème :

Théorème.

A tout automate linéairement borné on peut associer un a.p.c.b qui reconnaisse le même langage, de plus il sera déterministe si l'automate à mémoire linéairement borné l'est.

Démonstration.

Nous laissons au lecteur le soin de constater que dans les détails la construction précédente convient.

Notons que pour le cas déterministe : nous avons pris le parti, lorsque M passe par une configuration finale de faire évoluer A vers une configuration finale ici $[\Lambda, A_x]$ par les règles B- et C-. Dans ce cas si M ne bloque pas ce qui est possible. A ne "suit" plus M. En effet les cas A- et B- s'excluent si $q \in Q_F$, nous opérons $(A_x, \Lambda) \in \delta_A(q,x)$ par B-, même si $\exists (q',e)$, $(q',e) \in \delta(q,x)$.

Par ailleurs il est immédiat que :

$$|\delta(q,x)| = 1 \Longrightarrow |\delta_A(q,x)| = 1.$$

cqfd.

e) Nous allons à présent établir la réciproque de la propriété précédente à savoir que l'on peut associér à tout a.p.c.b A un M.L.B qui reconnaisse le même langage.

Nous procédons comme pour la propriété directe.

Définition.

Nous dirons qu'un M.L.B, $M = \langle \Sigma_M, Q_M, \delta_M, q_o, Q_A \rangle$ est associe à un a.p.c.b $A = \langle \Gamma, \Sigma, M, \delta, F \rangle$ simplifié, possédant les propriétés P1 et P2 :

P1 $\delta(X,Y) = (u,\Lambda) \Longrightarrow Y \notin T_r(\#_d)$ ou $(u \in F$ et $X \in T_r(\#_g))$

P2 $\delta(X,Y) = (\Lambda,u) \Longrightarrow X \notin T_r(\#_g)$ ou $(u \in F$ et $Y \in T_r(\#_d))$

1 - $\Sigma_M = \Sigma \cup \Gamma \cup M \cup \{\phi\}$

2 - $Q_M = \{q_a \ / \ a \in \Sigma \cup \Gamma\}$

3 - $q_o = q\#_g$

4 - $Q_F = \{q_F\}$

5 - δ_M est obtenu à partir de δ de la manière suivante :

a) Si $(a'b',\Lambda) \in \delta(a,b)$ alors :

$(\vec{q}_{b'}, a') \in \delta_M(q_a,b)$ (a1)

et

$(q_{b'}, +1) = \delta_M(q_b{}', a')$ (a2)

b) Si $(\Lambda,a'b') \in \delta(a,b)$ alors :

$(\overleftarrow{q}_{a'}, b') \in \delta_M(q_a,b)$ (b1)

et

$(\overset{<}{q}_{a'}, -1) = \delta_M(\overleftarrow{q}_a{}', b')$ (b2)

c) Si $(a',\Lambda) \in \delta(a,b)$ alors :

si $b \notin T_r(\#_d)$ $(q_a{}', \Phi) \in \delta_M(q_a,b)$ (c1)

si $b \in T_r(\#_d)$ $(q_F, \Phi) \in \delta_M(q_a,b)$ (c2)

d) Si $(\Lambda,a') \in \delta(a,b)$ alors :

si $a \notin T_r(\#_g)$, $(\overset{<}{q}_{a'}, \Phi) \in \delta_M(q_a,b)$ (d1)

si $a \in T_r(\#_g)$, $(q_F, \Phi) \in \delta_M(q_a,b)$ (d2)

e) $\delta_M(q_a, \Phi) = (q_a, +1)$ (e1)

$\delta_M(\overset{<}{q}_a, \Phi) = (\overset{<}{q}_a, -1)$ (e2)

$\delta_M(\overset{<}{q}_a, b) = (q_b, a)$ (e3)

Le lecteur vérifiera aisément que les propriétés P1 et P2 ne restreignent pas la généralité de la définition, et conserve éventuellement le déterminisme.

f) Nous allons à présent donner la définition des configurations correspondantes dans A et M.

Définition.

Nous dirons que deux configurations K_A dans A et K_M dans M sont correspondantes et nous noterons $K_A \longleftrightarrow K_M$. Si nous nous trouvons dans un des 4 cas suivants :

1 - $K_1 = [\#_g, w\#_d]$, $K_M = q_{\#_g} w\#_d$
pour tout $w \in \Sigma^*$

2 - $K_A = [u_1 u_2 \ldots u_k, u_{k+1} \ldots u_n]$

$K_M = \phi^* u_1\ \phi^* u_2\ \phi^* \ldots \phi^* q_{u_k}\ u_{k+1}\ \phi^* u_{k+1}\ \phi^* \ldots \phi^* u_n\ \phi^*$
pour tout $u_i \in (\Sigma \cup \Gamma)^o$, $u_k \neq \Lambda$, $u_{k+1} \neq \Lambda$

3 - $K_A = [u, \Lambda]$, $K_M = \Phi^* q_F \phi^*$ si $u \in F$

$K_A = [\Lambda, u]$, $K_M = \phi^* q_F \Phi^*$ si $u \in F$

toute configuration de K_M pour laquelle l'état est marqué soit $\vec{q}_a$, $\overleftarrow{q}_a$, $\overset{<}{q}_a$, ou pour laquelle des éléments Φ substistent entre q_{u_k}, et u_{k+1} sera dite intermédiaire.

g) Lemme.

A tout couple de configuration de A, K_A et K'_A, tels que $K_A \vdash K'_A$ on peut associer de manière unique un couple de configuration dans M, K_M et K'_M non intermédiaires telles que :

$$K_M \longleftrightarrow K_A,\ K'_A \longleftrightarrow K_M \text{ et } K_M \overset{*}{\vdash} K'_M$$

et réciproquement.

Démonstration.

La forme des règles de M, permet de déterminer pour chacune d'entre elle la règle de A dont elle est issue et ceci de manière unique, en effet :

$$\forall a,\ b,\ a',\ b'$$

$(\vec{q}_{b'}, a') \in \delta(q_a, b)$ ne peut provenir que de $(a', b', \Lambda) \in \delta(a,b)$ il en est de même pour les (b1), c) et d) de la construction.

Les autres règles de la construction n'étant là que pour assurer dans M le cheminement vers une configuration qui ne soit pas de transition, et qui permettent l'établissement d'une correspondance avec une configuration de A.

Nous traiterons donc la propriété directe et la propriété réciproque en même temps, sans le préciser pour chaque cas.

Notons que les cheminements par états marqués sont nécessairement déterministes.

Soit $K_A = [u_1 \ldots u_k,\ u_{k+1} \ldots u_n]$ et la configuration correspondante

$$K_M = \phi^* u_1 \phi^* \ldots \phi^* q_{u_k} u_{k+1} \phi^* \ldots \phi^* u_n \phi^*.$$

Nous allons envisager toutes les possibilités de mouvement à partir de K_A.

⌶ Règles a) $(u'_k u'_{k+1},) \in (u_k, u_{k+1})$

1 Ce qui nous donne dans A :

$$K_A \vdash K'_A = [u_1 u_2 \ldots u'_k\ u'_{k+1},\ u_{k+2} \ldots u_n]$$

il vient dans M

$$K_M \vdash K^1_M = \phi^* u_1 \phi^* \ldots \phi^* \vec{q}_{u'_{k+1}}\ u'_k \phi^* \ldots \phi^* u_n \phi^* \quad \text{par (a1)}$$

$$K_M^1 \vdash M_M^2 = \text{———} \; \phi^* u'_k q_{u'_{k+1}} \phi^* \; \text{———} \quad \text{par (a2)}$$

$$K_M^2 \overset{*}{\vdash} K'_M = \text{———} \; \phi^* u'_k \phi^* q_{u'_{k+1}} u_{k+2} \phi^* \text{—} \quad \text{par (e1)}$$

Nous avons bien $K'_A \longleftrightarrow K'_M$.

2 Dans le cas d'application d'une règle a), si $u_{k+1} \ldots u_k = \Lambda$
$K'_A = [u_1 \ldots u'_{k+1}, \Lambda]$ configuration dans laquelle A bloque.

Notons qu'alors par les règles (a2) et (e1), M évolue vers
$K'_M = \ldots\ldots\ldots \phi^* q_{u'_{k+1}}$ configuration dans laquelle M bloque également.

II Règles b) $(\Lambda, u'_k u'_{k+1}) \in \delta(u_k, u_{k+1})$

1 Ce qui nous donne dans A :

$$K_A \vdash K'_A = [u_1 \ldots u_{k-1}, u'_k u'_{k+1} \ldots u_n]$$

il vient dans M :

$$K_M \vdash K_M^1 = \phi^* u_1 \phi^* \ldots \phi^* \overleftarrow{q}_{u'_k} u'_{k+1} \phi^* \ldots \phi^* u_n \phi^* \quad \text{par (b1)}$$

$$K_M^1 \vdash K_M^2 = \text{——} \; u_{k-1} \phi^* \overset{<}{q}_{u'_k} \Phi u'_{k+1} \phi^* \; \text{———} \quad \text{par (b2)}$$

$$K_M^2 \overset{*}{\vdash} K_M^3 = \text{——} \; \phi^* \overset{<}{q}_{u'_k} u_{k-1} \phi^* u'_{k+1} \phi^* \; \text{———} \quad \text{par (e2)}$$

$$K_M^3 \vdash K'_M = \text{——} \; \phi^* q_{u_{k-1}} u'_k \phi^* u'_{k+1} \phi^* \; \text{———} \quad \text{par (e3)}$$

ici encore $K'_A \longleftrightarrow K'_M$.

2 De même que précédemment, si ici $u_1 \ldots u_{k-1} = \Lambda$,
$K'_A = [\Lambda, u'_k \ldots u_n]$ et $K'_n = q_{u_{k-1}} \Lambda \phi^*$ ... plus aucun mouvement n'est possible ni dans A ni dans M.

⌶ Règles (c1) $(u'_k, \Lambda) \in \delta(u_k, u_{k+1})$

ce qui nous donne dans A :

$$K_A \vdash K'_A = [u_1 \;\dots\; u'_k,\; u_{k+2} \;\dots\; u_n]$$

il vient dans M :

$$K_M \vdash K_M^1 = \phi^* u_1 \phi^* \;\dots\; \phi^* u_{k+2} \phi^* \;\dots\; \phi^* u_n \phi^* \qquad \text{par (c)}$$

$$K_M^1 \vdash K'_M = \text{———————} \phi^* q_{u'_k} u_{k+2} \phi^* \text{———————} \qquad \text{par (e3)*}$$

et $K'_A <\!\!\longrightarrow K'_M$

⌶ Règles (d1) $(\Lambda, u'_{k+1}) \in \delta(u_k, u_{k+1})$

ce qui nous donne dans A :

$$K_A \vdash K'_A = [u_1 \;\dots\; u_{k-1},\; u'_{k+1} \;\dots\; u_n]$$

il vient dans M :

$$K_M \vdash K_M^1 = \phi^* u_1 \phi^* \;\dots\; u_{k-1} \phi^* \overset{<}{q}_{u'_{k+1}} \Phi \phi^* u_{k+2} \phi^* \;\dots\; \phi^* \quad \text{par (d1)}$$

$$K_M^1 \overset{*}{\vdash} K_M^2 = \text{———————} \phi^* \overset{<}{q} u'_{k+1} u_{k-1} \phi^* u_{k+2} \phi^* \text{———————} \quad \text{par (e2)*}$$

$$K_M^2 \vdash K'_M = \text{———————} \phi^* q_{u_{k-1}} u'_{k+1} \phi^* \text{———————} \quad \text{par (e3)}$$

et $K'_A <\!\!\longrightarrow K'_M$

⌶ Règle (d2) Pour $|u| = |v| = 1$

Si $K_A = [u,v]$, configuration qui correspond à $K_M = \phi^* q_u v \phi^*$ dans M.

Alors une règle (d2) est applicable. Notons que si $K_A = [u, vv_1 \;\dots\; v_n]$,

(d2) ne peut se produire, car A possèdant la propriété P2 :

$$(\Lambda, a') \in \delta(a,b) \Longrightarrow a \notin T_r(\#_g) \text{ ou } (a' \in F \text{ et } b \in T_r(\#_d)).$$

Il vient :

si $(\Lambda, a') \in \delta(u,v)$ alors $(q_F, \Phi) \in \delta(q_u, v)$

donc :

$$K_A = [u,v] \vdash [\Lambda,a'] \text{ avec } a' \in F$$

et

$$K_M = \phi^* q_u v \phi^* \vdash \phi^* q_F \phi^* \text{ qui est finale}$$

II Règle (c2) Pour $|u| = |v| = 1$.

Si $K_A = [u,v]$, configuration qui correspond à $K_M = \phi^* q_u v \phi^*$ dans M. Alors une règle (c2) est applicable. Ici encore notons que si $K_A = [u_1 \ldots u_n\, u,\, v]$, (c2) ne peut se produire, car A possédant la propriété P1

$$(a',\Lambda) \in \delta(a,b) \Longrightarrow b \notin T_r(\#_d) \text{ ou } (a' \in F \text{ et } a \in T_r(\#_g))$$

Il vient

si $(a',\Lambda) \in \delta(u,v)$ alors $(q_F,\Phi) \in \delta(q_u,v)$

donc :

$$K_A = [u,v] \vdash [a',\Lambda] \text{ avec } a' \in F$$

$$K_M = \phi^* q_u v \phi^* \vdash \phi^* q_F \phi^* \text{ qui est finale}$$

Remarquons que tous les mouvements qui dans M permettent d'atteindre une configuration qui puisse être mise en correspondance avec une configuration dans A sont déterministes.

Il est immédiat par la construction que l'on a :

$$|\delta(a,b)| = 1 \Longrightarrow |\delta_M(q_a,b)| = 1$$

Nous pouvons énoncer le théorème dont nous laissons au lecteur le détail de la démonstration.

Théorème.

A tout a.p.c.b (resp. déterministe) on peut associer un M.L.B (resp. déterministe) qui reconnaisse le même langage.

h) Les théorèmes énoncés en d) g) peuvent se résumer un seul.

Théorème.

Les automates à piles communicantes bornées sont équivalents aux automates linéairement bornés.

II.4 AUTOMATE A PILES COMMUNICANTES NON EXPENSIVES A DROITE ET AUTOMATE A PILE DE MEMOIRE

Nous commençons par rappeler les définitions des automates à pile de mémoire. Définitions que l'on peut trouver dans [8].

a) AUTOMATE À PILE DE MÉMOIRE DÉFINITION . PROPRIÉTÉS

Définition.

Un automate à pile est un septuplet $M = <\Sigma, \Gamma, Q, \delta, q_o, Q', Z_o>$

Σ est un alphabet, alphabet dit d'entrée
Γ est un alphabet, alphabet de pile
Q ensemble fini appelé ensemble d'états
δ est une relation finie $(Q \times \Gamma \times \Sigma^0) \times (Q \times \Gamma^*)$
nous noterons $(q,\gamma) \in \delta(q,Z,a)$ (1)
q_o élément distingué de Q, état initial
Q' sous ensemble de Q, ensemble des états terminaux
Z_o élément distingué de Γ, appelé borne de la pile.

Une configuration d'un automate à pile sera un triplet

$K = (q,y,x)$ (1), $q \in Q$, $x \subset \Sigma^*$ et $y \in \Gamma^*$

on dit que K_2 dérive directement de K_1 dans chacun des cas suivants : nous noterons $K_1 \vdash K_2$

1) $K_1(q,yZ,ax)$ $K_2 = (q',ym,x)$
si $(q',m) \in \delta(q,Z,a)$

(1) Cette disposition des éléments dans le triplet et dans les configurations a été choisie pour sa plus grande similitude avec les configurations dans les ned où le mot d'entrée est sur la pile droite.

2) $K_1 = (q,\gamma Z,ax)$ $K_2 = (q',\gamma m,ax)$

si $(q',m) \in \delta(q,Z,\Lambda)$.

Soit $\overset{*}{\vdash}$ la clôture réflexive et transitive de $\vdash$.
A un automate on associe le langage dit reconnu par cet automate, nous le noterons T(M) :

$$T(M) = \{w \in \Sigma^* \ / \ \exists q \in Q' \text{ et } \gamma \in \Gamma^* \text{ telles que :}$$

$$[q_o,Z_o,w] \overset{*}{\vdash} [q,\gamma,\Lambda]\}$$

A l'intérieur de la classe des langages reconnus par un automate à pile, de nombreuses sous classes ont été définies, qui correspondent à des formes particulières de l'automate.

On dit qu'un automate est déterministe si :

$\forall q \in Q$, $\forall Z \in \Gamma$ on a

soit $\delta(q,Z,\Lambda) = \emptyset$ et $\forall a \in \Sigma$, $|\delta(q,Z,a)| \leq 1$

soit $\delta(q,Z,\Lambda)$ a un élément et $\forall a \in \Sigma$

$\delta(q,Z,a) = \emptyset$

nous noterons a.p.m.d comme abréviation d'automates à pile de mémoire déterministe.

b) Il s'agit donc de simuler un A.P.M par un n.e.d. Dans un A.P.M nous avons accès pour déterminer la configuration suivante de l'automate à trois éléments : un état élément de Q une lettre ou la chaine vide élément de Σ^o, une lettre élément de Σ', dans un n.e.d nous n'avons accès qu'aux deux sommets de piles.

Nous choississons de placer en sommet de pile gauche des éléments de $Q \times \Gamma^*$ et en sommet de pile droite des éléments de Σ, nous aurons donc les configurations de la forme :

$[$——$(q,\gamma Z),a$——$]$ dans le n.e.d

et

$[q,$ —— $\gamma Z,a$——$]$ dans l'A.P.M.

Comme précédemment nous allons définir un n.e.d A associé à un a.p.m, mais nous imposons à l'a.p.m, de ne jamais lire la chaîne vide sur le mot d'entrée, propriété qui ne restreint pas la généralité, et qui est démontrée dans [18].

Définition.

Nous dirons qu'un n.e.d $A = <\Gamma_A, \Sigma_A, M_A, \delta_A, F_A>$ est associée à un a.p.m M possédant la propriété suivante :

- $\forall q, \forall Z\ \delta(q, Z, \Lambda) = \emptyset$

si $M = <\Sigma, \Gamma, Q, \delta, q_o, Q', Z_o>$ nous avons :

1- $\Sigma_A = \Sigma$

2- Soit :

$$E = \{v / \exists q, q' \in Q,\ a \in \Sigma,\ Z \in \Gamma,\ u \in \Gamma^* \text{ tels que : } (q', vu) \in \delta(q, Z, a)\}$$

Alors :

$$\Gamma_A = Q \times E \cup \{(q,a) \ / \ q \in Q,\ a \in \Sigma \cup \{\#\}\}$$

3- $M_A = \{(q_o, Z_o), \#\}$ i.e $\#_g = (q_o, Z_o)$, $\#_d = \#$

4- $F_A = \{A_x\}$

5- δ_A est obtenu à partir de δ de la manière suivante :

pour tout $q', q \in Q$, $\gamma' \in \Gamma^+$, $Z \in \Gamma$, $a \in \Sigma$

a) Si $(q', \gamma') \in (q, Z, a)$ alors :

$$\forall \gamma \in E\ ((\gamma)(q', \gamma'), \Lambda) \in \delta_A((q, \gamma Z), a)$$

b) Si $(q',\Lambda) \in (q,Z,a)$ alors :

$\forall \gamma \in E \; ((q',\gamma),\Lambda) \in \delta_A((q,\gamma Z),a)$

c) $\forall a \in \Sigma \cup \{\#\} \; \forall \gamma \in E$

c1 $\delta_A((q,\Lambda),\ a) = (\Lambda,\ (q,a))$

c2 $\delta_A((\gamma),(q,a)) = ((q,\gamma),a)$

d) $\forall q_F \in Q' \; \forall \gamma \in E$

d1 $\delta_A((q_F,\gamma_Z),\#) = (\Lambda,A_x)$

d2 $\delta_A((\gamma),A_x) = (\Lambda,A_x)$

Dans ce qui précéde, nous convenons que $(\Lambda) = \Lambda$.

c) Donnons à présent la définition des configurations correspondantes.

Définition.

Nous dirons que deux configurations K_A dans A et K_M dans M sont correspondantes nous noterons $K_A \longleftrightarrow K_M$ si et seulement si

$$K_M = (q,\ \gamma,\ w)$$

$$K_A = [(\gamma_1)\ \ldots\ (\gamma_{n-1})\ (q,\gamma_n),\ w\#]$$

avec $\gamma_1 \ldots \gamma_n = \gamma$ et $\gamma_n \neq \Lambda$

Notons que la configuration initiale (q_o,Z_o,w) correspond dans A à $[(q_o,Z_o),w\#]$.

Toutes configurations de A, ayant un élément de la forme (q,Λ) en sommet de pile gauche ou (q,a) en sommet de pile droite sera dite intermédiaire.

d) Lemme.

Pour tout couple de configurations correspondantes non intermédiaires K_A dans A et K_M dans M il y a équivalence entre.

1 - $\exists K'_A$ non intermédiaire, et ne contenant pas A_x telle que

pour n = 1 ou 2 $K_A \overset{n}{\vdash} K'_A$

2 $\exists K'$ telle que $K_M \vdash K'_M$

de plus $K'_M <-> K'_A$

Démonstration.

Nous allons envisager les cas possibles de couple de configurations K_A et K_M correspondantes et les mouvements possibles à partir de ces configurations.

Nous ne préciserons pas à chaque fois que $K'_M <\!\!-\!\!-\!\!> K'_A$ le lecteur le vérifiera aisément.

Notons que la cas :

$$K_M = (q, \gamma_1 \ldots \gamma_n Z, \Lambda), \quad K_A = [(\gamma_1) \ldots (q, \gamma_n Z), \#]$$

n'est pas à envisager ici, car il donne naissance dans A à un élément A_x si $q \in Q'$.

CAS 1.

$$K_M = (q, \gamma_1 \ldots \gamma_n Z, aw), \quad K_A = [(\gamma_1) \ldots (\gamma_{n-1})(q, \gamma_n Z), aw\#]$$

$\amalg$ par a) de la construction :

$$K_M \vdash (q', \gamma_1 \ldots \gamma_n \gamma', w) = K'_M$$
$$K_A \vdash ((\gamma_1) \ldots (\gamma_{n-1})(\gamma_n)(q', \gamma'), w\#) = K'_A$$

$\amalg$ par b) de la construction :

$$K_M \vdash (q', \gamma_1 \ldots \gamma_n, w) = K'_M$$
$$K_A \vdash ((\gamma_1) \ldots (\gamma_{n-1})(q', \gamma_n), w) = K'_M$$

CAS 2. $K_M = (q, \gamma_1 \ldots \gamma_n Z, aw)$, $K_A = [(\gamma_1) \ldots (\gamma_n)(q,Z), aw\#]$

$\amalg$ par a)

$$K_M \vdash (q', \gamma_1 \ldots \gamma_n \gamma', w) = K'_M$$

$$K_A \vdash ((\gamma_1) \ldots (\gamma_n)(q',\gamma'), w\#) = K'_A \text{ car } (\Lambda) = \Lambda$$

$\amalg$ par b)

$$K_M \vdash (q', \gamma_1 \ldots \gamma_n, w) = K'_M$$

$$K_A \vdash ((\gamma_1) \ldots (\gamma_n)(q',\Lambda), w\#) = K'_A$$

Soit $a_1 w_1 = w$, notons que si $w = \Lambda$, le symbole # figure sur la pile droite qui donc n'est en aucun cas vide.

Il vient par (c1) et (c2)

$$K_A^I \vdash [(\gamma_1) \ldots (\gamma_n), (q',a_1)w_1\#] \vdash [(\gamma_1) \ldots (q',\gamma_n), w_1\#] = K'_A$$

Notons que si $\gamma_1 \ldots \gamma_n = \Lambda$, M bloque dans la configuration : $[q', \Lambda, w]$ et A dans la configuration : $[\Lambda, (q',a_1)\ w_1\#]$

cqfd.

e) Corollaire.

Pour tout mot $w \in \Sigma^*$ on a :

Il existe une suite de configuration $(K_i)_{i=o}^{n}$ de M telle que $K_o = [q_o, Z_o, w]$ et pour $q_F \in Q'$ $K_n = [q_F, \gamma Z, \Lambda]$ et $K_O \vdash K_1 \vdash \ldots \vdash K_n$
si et seulement si il existe une suite de configuration $(Ci)_{i=o,\ n<p}^{p}$ de A telle que : $C_o = [(q_o, Z_o), w\#]$ et
$C_o \vdash C_1 \vdash \ldots \vdash C_p = [\Lambda, A_x]$.

Démonstration.

Nous avons comme conséquence du lemme précédent :

$$[q_o, Z_o, w] \overset{*}{\vdash} [q_F, \gamma Z, \Lambda] = K_n$$

ssi $$[(q_o, Z_o), w\#] \overset{*}{\vdash} [(\gamma_1) \ldots (\gamma_{k-1}) \ldots (\gamma_{k-1})(q_F, \gamma_k Z), \#] = C_{n'}$$

avec $q_F \in Q'$.

Comme $q_F \in Q'$ les règles d) assurent dans A le mouvement

$$C_{n'} \overset{*}{\vdash} [\Lambda, A_x]$$

ceci de manière déterministe.

La condition est de ce fait suffisante.

La condition est nécessaire en effet :

si $\exists C_o \overset{*}{\vdash} [\Lambda, A_x]$ alors $\exists C_n$ telle que :

$$C_o \overset{*}{\vdash} C_n = [(\gamma_1) \ldots (q, \gamma_n Z), \#] \overset{*}{\vdash} [\Lambda, A_x]$$

avec $q \in q_F$

donc $\exists K_o \overset{*}{\vdash} [q_F, \gamma Z, \Lambda]$ avec $\gamma_1 \ldots \gamma_n = \gamma$ conséquence du lemme

cqfd.

f) Ceci nous permet d'énoncer le théorème, conséquence immédiat du corollaire précédent :

Théorème.

Tout langage reconnu par un a.p.m est reconnaissable par un n.e.d.

II.5 SIMULATION DES A.P.M.D. PAR LES N.E.D.D.

SIMULATION DES FAMILLES Δ_i

a) Nous allons nous intéresser dans cette partie aux langages reconnus par des a.p.m.d, par réinitialisation de la pile, et par pile vide. Plus précisément :

Définition.

Soit M un a.p.m.d : $M = <\Sigma, \Gamma, Q, \delta, q_o, Q', Z_o>$.

Pour un sous ensemble E, $E \subseteq \Gamma^*$ nous définissons le langage $T(M,E) \subseteq \Sigma^*$ comme suit :

$$T(M,E) = \{w \in \Sigma^* \ / \ (q_o, Z_o, w) \overset{*}{\vdash} (q, \alpha, \Lambda) \text{ pour } q \in Q' \text{ et } \alpha \in E\}$$

en particulier soit :

$$T_o(M) = T(M,\Gamma^*)$$

$$T_1(M) = T(M,Z_o) \quad (1)$$

$$T_2(M) = T(M,\Lambda)$$

$T_o(M)$ correspond pour tout M aux langages déterministes.

Dans les cas non déterministes ces trois familles coïncident, il n'en est pas de même pour les non-déterministes.

Définition.

Nous définissons trois familles de langage comme suit :
pour i = 0,1,2

$$\Delta_i = \{T_i(M) \ / \ M \text{ est un a.p.m.d}\}$$

(1) Notons que l'on a : $T(M,\Gamma) = T(M,Z_o)$

b) Nous allons donner la construction d'un n.e.d.d associé à un a.p.m.d. Cette construction ne varie que sur des points relatifs aux configurations initiales et finales particulières à chacune des familles Δ_O, Δ_1 ou Δ_2, nous la donnons une seule fois pour les trois familles en précisant les points où elle diffère.

Définition.

Nous dirons qu'un n.e.d.d $A = <\Sigma_A, \Gamma_A, M_A, \delta_A, F_A>$ est associé à un a.p.m.d $M = <\Sigma, \Gamma, Q, \delta, q_o, Q', Z_o>$ vérifiant les propriétés suivantes :

pour tout $q, q' \in Q$ $z, z' \in \Gamma$, $\gamma' \in \Gamma^*$, $a \in \Sigma'^{o}$

P1 $\delta(q,Z,a) = (q', \gamma'Z') \Longrightarrow \delta(q',Z',\Lambda) = \emptyset$

P2 $\delta(q_o,Z_o,\Lambda) = \emptyset$

P3 $\delta(q_F,Z_o,\Lambda) = \emptyset$ pour tout $q_F \in Q'$ ou
$\delta(q_F,Z_o,\Lambda) = (q',\Lambda)$ pour tout q_F, $q' \in Q'$
si et seulement si :

1 - $\Sigma_A = \Sigma$

2 - Si $E = \{u \ / \ \exists v \in \Gamma^* \ \exists q, Z, a, q' \text{ tq } \delta(q,Z,a) = (q',w)\}$
alors :
$\Gamma_A = E \cup Q \times E \cup Q$

3 - $F_A = \{S\}$ pour Δ_O et Δ_2
$F_A = \{(q,Z_o) \ / \ q \in Q'\}$ pour Δ_1

4 - $\#_g = (q_o,Z_o)$ pour Δ_O, Δ_1, Δ_2
$\#_d = \Lambda$ pour Δ_1 et Δ_2

5 - δ_A est obtenu à partir de δ de la manière suivante :

a) pour $a \neq \Lambda$, $|\gamma'| \geq 1$

$\delta(q,Z,a) = (q',\gamma') \Longrightarrow \forall\gamma \in E\ \delta_A((q,\gamma z),a) = ((\gamma)(q',\gamma'),\Lambda)$

b) pour $a \neq \Lambda$

b1) $\delta(q,Z,a) = (q',\Lambda) \Longrightarrow \forall\gamma \in E\ \delta_A((q,\gamma Z),a) = ((\gamma),(q'))$

b2) plus dans Δ_2 si $Z = Z_o$ et $q' \in Q'$ il vient :

$\delta(q,Z_o,a) = (q',\Lambda) \Longrightarrow \delta_A((q,Z_o),a) = (\Lambda,S)$

c) pour $|\gamma'| \geq 1$

$\delta(q,Z,\Lambda) = (q',\gamma') \Longrightarrow \forall\gamma \in E\ \delta_A((\gamma Z), (q)) = ((\gamma)(q',\gamma'),\Lambda)$

d)

d1) $\delta(q,Z,\Lambda) = (q',\Lambda) \Longrightarrow \forall\gamma \in E\ \delta_A((\gamma Z),(q)) = ((\gamma),(q'))$

d2) plus dans Δ_2 si $Z = Z_o$ et $q' \in Q'$ il vient :

$\delta(q,Z_o,\Lambda) = (q',\Lambda) \Longrightarrow \delta_A\ ((Z_o),(q)) = (\Lambda,S)$

e) $\delta(q,Z,\Lambda) = \emptyset \Longrightarrow \forall\gamma \in E\ \delta_A((\gamma Z),(q)) = ((q,\gamma Z),\Lambda)$

f) pour tout $q \in Q'$ et tout γ

$\delta((q, \gamma z), \#_d) = (\Lambda,S)$

$\delta((\gamma),S) = (\Lambda,S)$

c) La définition des configurations correspondantes qui est compatible avec cette construction est la suivante :

Définition.

Nous dirons que deux configurations K_M de M et K_A de A sont correspondantes, nous noterons $K_M \longleftrightarrow K_A$ si :

pour $\gamma_i \in \Gamma^*$ pour tout i, $Z \in \Gamma$, $w \in \Sigma^*$, $q \in Q$

$$K_M = [q,\gamma_1\gamma_2 \ldots\gamma_n Z,w] \text{ alors :}$$

CAS 1.

$\amalg$ si $\delta(q,Z,\Lambda) = \emptyset$ alors :

. $K_A = [(\gamma_1) \ldots (q,\ \gamma_n Z),\ w\ \#_d]$ pour Δ_0

. $K_A = [(\gamma_1) \ldots (q,\gamma_n Z),w]$ pour Δ_1, Δ_2

CAS 2.

$\amalg$ si $\delta(q,Z,\ ) \neq \emptyset$ alors :

. $K_A = [(\gamma_1) \ldots (\gamma_n Z),(q)\ w\ \#_d]$ pour Δ_0

. $K_A = [(\gamma_1) \ldots (\gamma_n Z),\ (q)w]$ pour Δ_1, Δ_2

Remarquons que dans A nous aurons un type de configuration qui sera dite intermédiaire, ce type correspond au cas 2 avec $\delta(q,Z,\Lambda) = \emptyset$, en effet dans ce cas la règle e) assure le passage vers une configuration du type cas 1, ce mouvement dans A n'ayant pas de correspondans dans M, ceci justifie l'appelation intermédiaire.

d) Lemme "des mouvements correspondants".

Pour tout couple de configurations correspondantes K_A dans A, et K_M non finale dans M on a

1 - $\exists K'_A$ telle que $K_A \vdash K'_A$ ou $\exists K'_A$, $\exists K^I_A$ configuration intermédiaire de A.

telles que : $K_A \vdash K^I_A \vdash K'_A$

$\Longrightarrow \exists K'_M$ unique telle que $K_M \vdash K'_M$ et $K'_A \longleftrightarrow K'_M$

2 - $\exists K'_M$ telle que $K_M \vdash K'_M$

$\Longrightarrow$ $\exists K'_A$ unique telle que $K'_A \longleftrightarrow K'_M$ et

. soit $K_A \vdash K'_A$

. soit $\exists K^I_A$ configuration intermédiaire de A telle que :

$$K_A \vdash K^I_A \vdash K'_A.$$

Preuve.

Remarquons tout d'abord que toutes les impliquations de 5) de la définition sont en fait des équivalences logiques, ceci est dû à la forme particulière qu'a chaque règle de A, forme variant suivant la règle de M dont elle est issue.

Nous allons donc mener simultanément les démonstrations des points 1 et 2.

Nous allons envisager les divers cas suivant que l'on applique une règle de type a) b1) c) d1) ou e) en effet remarquons que b2) c2) et f) s'appliquant nécessairement à des configurations finales ne sont pas à envisager ici.

A) Soient $(q,Z,\Lambda) = \emptyset$ et

$$K_A = [(\gamma_1) \ldots (q,\gamma_n Z),\ aw\ \#^o_d] \longleftrightarrow K_M = [q,\gamma_1 \ldots \gamma_n Z, aw] \quad (1)$$

$\amalg$ par a) nous avons

$$\delta(q,Z,a) = (q',\gamma') \Longleftrightarrow \forall\gamma \in F\ \delta_A((q,\gamma Z),a).((\gamma)(q',\gamma'),\Lambda)$$

ce qui permet

$$K_A \vdash [(\gamma_1) \ldots (\gamma_n)(q',\gamma'),\ w\ \#^o_d] = K'_A$$

$$K_M \vdash [q',\ \gamma_1 \ldots \gamma_n \gamma', w] = K'_M$$

nous avons bien $K'_A \longleftrightarrow K'_M$ car P1 implique $\delta(q',\ \gamma'^{(1)},\Lambda) = \emptyset$ (2)

(1) $\#^o_d = \{\#_d, \Lambda\}$

(2) $\gamma'^{(1)}$ vaut l'élément le plus à droite de γ' si $\gamma' \neq \Lambda$ et Λ si $\gamma' = \Lambda$.

II par b1) nous avons :

$$\delta(q,Z,a) = (q',\Lambda) \Longleftrightarrow \forall\gamma \in E\ \delta_A((q,Z),a) = ((\gamma),(q'))$$

ce qui permet :

$$K_A \vdash [(\gamma_1) \ldots (\gamma_n),\ (q')\ w\ \#\overset{o}{d}\ ] = K_A^1 \quad \text{et :}$$

$$K_M \vdash [q',\ \gamma_1 \ldots \gamma_n, w] = K_M'$$

Nous avons alors deux cas : si $\gamma_n = \gamma_n' Z_n$.

1 - $\delta(q',Z_n,\Lambda) = \emptyset$

alors grâce à une règle e) :

$$\delta_A((\gamma_n' Z_n),\ (q')) = ((q',\gamma' Z_n),\Lambda)$$

ceci correspond au cas ou K_A^1 est une configuration intermédiaire il vient :

$$K_A^1 \vdash [(\gamma_1) \ldots (\gamma_n,q'),\ w\ \#\overset{o}{d}] = K_A'$$

et on a bien $K_A' \longleftrightarrow K_M'$

2 - $\delta(q',Z_n,\Lambda) \neq \emptyset$

alors K_A^1 n'est pas une configuration intermédiaire et on a :

$$K_A^1 \longleftrightarrow K_M'$$

Ces deux cas, application de a) et b1) sont les seuls à considérer pour K_A

B) Soient $\delta(q,Z,\Lambda) \neq \emptyset$ et
$K_A = [(\gamma_1) \ldots (\gamma_n Z),(q) w\ \#\overset{o}{d}] \longleftrightarrow K_M = [q,\gamma_1 \ldots \gamma_n Z,\ w\ \#\overset{o}{d}]$

$\amalg$ par c) nous avons :

$$\delta(q,Z,\Lambda) = (q',\gamma') \Longleftrightarrow \forall\gamma \in E \ \delta_A((\gamma Z),(q)) = ((\gamma) \ (q',\gamma'),\Lambda)$$

ce qui permet :

$$K_A \vdash [(\gamma_1) \ldots (\gamma_n) \ (q',\gamma'),w \ \overset{o}{\#d}\ ] = K'_A \quad \text{et}$$

$$K_M \vdash [q', \gamma_1 \ldots \gamma_n\gamma', w \ \overset{o}{\#d}] = K'_M$$

donc $K'_A \longleftrightarrow K'_M$ cas P1 implique $\delta(q',\gamma'^{(1)},\Lambda) = \emptyset$

$\amalg$ par d1 nous avons :

$$\delta(q,Z,\Lambda) = (q',\Lambda) \Longrightarrow \forall\gamma\in E \ \delta_A((\gamma Z),(q)) = ((\gamma),(q'))$$

ce qui permet :

$$K_A \vdash [(\gamma_1) \ldots (\gamma_n), \ (q')w \ \overset{o}{\#d}] = K^1_A \quad \text{et :}$$

$$K_M \vdash [q', \gamma_1 \ldots \gamma_n,w] = K'_M$$

Nous retrouvons ici un cas déjà traité précédemment.

Ceci achève de démontrer le lemme.

e) <u>Lemme 1.</u>

$$T_d(M) = T(A).$$

Rappellons que :

$$[q,\ \gamma Z,\ w] \overset{d*}{\vdash\!\!-} [q',\ \gamma' Z',\ w']$$

ssi $[q,\ \gamma Z,\ w] \overset{*}{\vdash\!\!-} [q',\gamma' Z',w']$ et si $\delta(q',Z',\ \Lambda) = \emptyset$ et qu'un mot n'est d - accepté ssi :

$$[q_o,Z_o,w] \overset{d*}{\vdash\!\!-} [q',\gamma Z,\Lambda] \text{ et } \delta(q',Z,\Lambda) = \emptyset \text{ et } q' \in Q'$$

alors $T_d(M)$ est l'ensemble des mots d - acceptés par M.

<u>Preuve</u>.

$\amalg$ Montrons que $T_d(M) \subseteq T(A)$, soit $w \in T_d(M)$.

On a $[q_o,Z_o,w] \overset{*}{\vdash\!\!-} [q,\gamma Z,\Lambda]$ avec $q \in Q'$ et $\delta(q,Z,\Lambda) = \emptyset$ alors le lemme des "mouvements correspondants" étendu à une suite de mouvements implique :

$$\exists \gamma_1,\ldots,\gamma_n \in \Gamma^* \text{ telle que } \gamma_1 \ldots \gamma_n = \gamma \text{ et}$$

$$\exists\ [(q_o,Z_o),w\ \#d] \overset{*}{\vdash\!\!-} [(\gamma_1) \ldots (q,\ \gamma_n Z),\ \#d]$$

les règles f) de la construction assurent alors :

$$\exists\ [(\gamma_1) \ldots (q,\gamma_n Z),\ \#d] \vdash\!\!- [(\gamma_1)\ldots(\gamma_{n-1}),S] \overset{*}{\vdash\!\!-} [\Lambda,S]$$

car $q \in Q'$ donc $w \in T(A)$

$\amalg$ Montrons que $T(A) \subseteq T_d(M)$, soit $w \in T(A)$

Nous avons alors nécessairement :

$$\exists \gamma_1,\ \ldots,\ \gamma_n \in \Gamma^*,\ q \in Q',\ Z \in \Gamma$$

$$[(q_o,Z_o),\ w\ \#d] \overset{*}{\vdash\!\!-} [(\gamma_1) \ldots (q,\gamma_n Z),\#d] \overset{*}{\vdash\!\!-} [\Lambda,S]$$

En effet e) est la seule règle faisant apparaître S et elle n'est applicable que si $q \in Q'$.

Ceci, plus le lemme des mouvements correspondants étendus implique :

$$\exists\ [q_o, Z_o, w] \overset{*}{\vdash} [q, \gamma_1 \ldots \gamma_n Z, \Lambda]$$

avec $q \in Q'$ et $\delta(q, Z, \Lambda) = \emptyset$ par définition des configurations correspondantes donc $w \in T_d(M)$.

<u>cqfd</u>.

f) <u>Lemme 2.</u>

$$T_1(A) = T_1(M)/\{\Lambda\}$$

<u>Preuve</u>.

Π Montrons que $T_1(M)/\{\Lambda\} \subseteq T_1(A)$. Soit $w \in T_1(M)/\{\Lambda\}$, il vient :

$$\exists q \in Q' \text{ tel que } [q_o, Z_o, w] \overset{+}{\vdash} [q, Z_o, \Lambda]$$

Le lemme des mouvements correspondants étendu implique

$$\exists\ [(q_o, Z_o), w] \vdash [(q, Z_o), \Lambda]$$

car P3 implique $\delta(q, Z_o, \Lambda) = \emptyset$, q appartenant à Q' ce qui montre :

$$T_1(M)\ /\ \{\Lambda\} \subseteq T_1(A)$$

Π Réciproquement montrons $T_1(A) \subseteq T_1(M)\ /\ \{\Lambda\}$

$$w \in T_1(A) \Longrightarrow \exists\ [(q_o, Z_o), w] \overset{+}{\vdash} [(q, Z_o), \Lambda]$$

En effet les éléments de F_A sont nécessairement de la forme (q, Z_o), ils ne peuvent donc figurer qu'en sommet de pile gauche.

Ici encore le lemme des mouvements correspondant étendu implique :

$$\exists\ [q_o,\ Z_o, w] \overset{+}{\vdash} [q, Z_o, \Lambda]$$

donc $w \in T_1(M) / \{\Lambda\}$.

cqfd.

g) Lemme 3.

$$T_2(A) = T_2(M).$$

Preuve.

Π Montrons tout d'abord : $T_2(M) \subseteq T_2(A)$. Soit $w \subset T_2(M)$, la restriction 2 de la construction $\delta(q_o, Z_o, \Lambda) = \emptyset$, implique $w \neq \Lambda$ il vient :

$$\exists\ [q_o,\ Z_o,\ w] \overset{+}{\vdash} [q,\ \Lambda,\ \Lambda] \text{ avec } q \in Q'$$

comme nous imposons à Z_o de marquer le fond de pile la configuration qui précède $[q,\ \Lambda,\ \Lambda]$ est nécessairement de la forme $[q',\ Z_o,\ a]$ car $w \neq \Lambda$ avec $q' \in Q$ (éventuellement $q' = q_o$) et $a \in \Sigma \cup \{\Lambda\}$ et $w \in \Sigma^+$ nous avons

$$[q_o, Z_o, w] \overset{*}{\vdash} [q', Z_o, a] \vdash [q,\ \Lambda, \Lambda]$$

Ceci implique, lemme des mouvements correspondants étendu.
Deux cas :

1 $a \neq \Lambda$ et $\delta(q', Z_o, a) = (q, \Lambda)$

alors b2) implique : $\delta_A((q', Z_o), a) = (\Lambda, S)$ et le lemme d) $[(q_o, Z_o), w] \overset{*}{\vdash} [(q', Z_o), a]$ donc :

$$[(q', Z_o), a] \overset{*}{\vdash} [\Lambda, S]$$

2 $a = \Lambda$ et $\delta(q', Z_o, \Lambda) = (q, \Lambda)$ avec $q' \neq q_o$ alors d2 implique $\delta_A(Z_o),\ (q')) = (\Lambda, S)$ et le lemme d) $[(q_o, Z_o), w] \overset{+}{\vdash} [(Z_o),\ (q')]$ car $(q', Z_o, \Lambda) \neq \emptyset$. Donc $[(Z_o),\ (q')] \vdash [\Lambda, S]$.

Donc $w \in T_2(A)$, dans les deux cas.

II Montrons que : $T_2(A) \subseteq T_2(M) \: / \: \{\Lambda\}$

Soit $w \in T_2(A)$. La seule configuration finale possible est ici $[\Lambda,S]$, donc $w \neq \Lambda$.

Nous avons deux cas possibles :

1 la configuration qui précède $[\Lambda,S]$ est de la forme $[(q,Z_o),a]$, avec nécessairement $a \neq \Lambda$ il vient :

$$[(q_o,Z_o),w] \overset{*}{\vdash} [(q,Z_o),a] \vdash [\Lambda,S]$$

avec éventuellement $q = q_o$ donc :

$$\exists \; [q_o,Z_o,w] \overset{*}{\vdash} [q,Z_o,a]$$

$$[(q,Z_o),a] \vdash [\Lambda,S] \Longrightarrow \delta_A((q,Z_o),a) = (\Lambda,S)$$

d'où nécessairement

$$\delta(q,Z_o,a) = (q',\Lambda) \text{ avec } q' \in Q'$$

donc nous avons dans M :

$$[q_o,Z_o,w] \overset{*}{\vdash} [q,Z_o,a] \vdash [q',\Lambda,\Lambda]$$

donc $w \in T_2(M)$ et $w \neq \Lambda$.

2 La configuration qui précède $[\Lambda,S]$ est de la forme

$[(Z_o),(q)]$ il vient :

$$[(q_o,Z_o),w] \overset{+}{\vdash} [(Z_o),(q)] \vdash [\Lambda,S] .$$

D'où nécessairement :

$$\delta_A((Z_o),(q)) = (\Lambda,S)$$

ce qui ne peut venir que du fait :

$$(q,Z_o,\Lambda) = (q',\Lambda) \text{ avec } q' \in Q'$$

le lemme des mots correspondants étendu, plus cette dernière règle permettent d'écrire :

$$[q_o,Z_o,w] \overset{+}{\vdash} [q,Z_o,\Lambda] \vdash [q',\Lambda,\Lambda]$$

avec $q' \in Q'$ donc $w \in T_2(M)$ et $w \neq \Lambda$.

cqfd.

h) Ginsburg et Greibach ont montré dans [10] p.628 Lemme 2.2.

Lemme.

Pour tout automate à pile déterministe M, il existe un automate à pile déterministe N tel que :

$$T(M) = T_d(N) = T(N).$$

Ce lemme et les lemme 1, 2 et 3 permettent d'énoncer en résumé le théorème :

Théorème.

Pour tout langage $L \subseteq \Sigma^*$

$$L \in \Delta_0 \Longrightarrow L \in \text{n.e.d.d}$$
$$L \in \Delta_1 \text{ et } \{\Lambda\} \notin L \Longrightarrow L \in \text{n.e.d.d}_1$$
$$L \in \Delta_2 \text{ et } \{\Lambda\} \notin L \Longrightarrow L \in \text{n.e.d.d}_2$$

Notons que si $L \in \Delta_2$ et $\{\Lambda\} \in L$ alors nécessairement $L = \{\Lambda\}$. Et que, la chaîne vide ne peut appartenir à un langage de la famille n.e.d.d_1 car pour cela il faudrait que $[\#_g, \Lambda]$ soit une configuration finale or $\#_g$ ne peut appartenir à F, pour un a.p.c A. La chaîne vide ne peut non plus appartenir à un langage de la famille n.e.d.d_2, en effet il faudrait que le mouvement :

$$[\#_g, \Lambda] \vdash [\Lambda, S]$$

soit possible, ce n'est pas le cas.

II.6 SIMULATION DES N.E.D (D) PAR LES A.P.M (D)

SIMULATION DES FAMILLES N.E.D.D_I

a) Nous choisissons ici de manière naturelle de simuler la pile gauche du n.e.d par la pile de l'a.p.m. Le seul mouvement posant un problème est celui de réduction gauche, car dans ce cas il y a transformation du sommet de la pile droite dans le n.e.d ce qui reviendrai à transformer l'élément pointé du mot d'entrée dans l'a.p.m ce qui n'est pas possible. Nous mémoriserons donc cet élément dans l'état de l'a.p.m, en convenant que dans cet état il faut lire Λ. Il vient :

$$[\gamma ZX, aw] \vdash [\gamma Z, Aw]$$

dans le n.e.d A donne dans l'a.p.m M :

$$[q, X, aw] \vdash [q_A, \gamma Z, w]$$

l'état q_A indiquant que le mouvement suivant se fera par lecture de Λ, c'est-à-dire par considération du triplet (q_A, Z, Λ) dans M qui correspondra à la considération de (Z,A) dans A. Le marqueur de fond de pile droite pose un autre problème, en effet un tel marqueur ne figure pas dans les configurations initiales d'un a.p.m. Nous construirons donc un a.p.m associé qui reconnaitra, pour tout L reconnu par un n.e.d, L $\#_d$. Ceci étant possible car Ginsburg et Greibach ont montré [10] que L est déterministe si et seulement si Lc est déterministe pour $c \notin \Sigma$.

Les définitions des automates étant données par ailleurs nous pouvons tout de suite donné la définition de l'automate à pile associé à un automate à pile communicantes non expressif à droite.

b) *Définition.*

Nous dirons qu'un a.p.m $M = \langle \Sigma_M, \Gamma_M, Q_M, \delta_M, q_o, Q'_M, Z_o \rangle$ est associé à un n.e.d $A = \langle \Gamma, \Sigma, M, \delta, F \rangle$ possédent les propriétés suivantes :

. $(\Lambda, X) \in \delta(a,b) \Longrightarrow X \notin \Sigma \cup \{\#_d\}$ (1)

. Les configurations finales sont de la forme $[\Lambda, S]$ pour $S \in F$ si :

1 - $\Sigma_M = \Sigma \cup \{\#_d\}$

2 - $\Gamma_n = \Gamma$

3 - $Z_o = \#_g,\ q_o = q_\Lambda$

4 - $Q = \{q_X \ /\ \exists (a,b)$ tel que : $(\Lambda, X) \in \delta(a,b)\} \cup \{q_\Lambda\}$

5 - $Q' = \{q_S \ /\ S \in F\}$

6 - δ_M est obtenu à partir de δ de la manière suivante :

a) $\forall b \in \Sigma_{\#_d}^{(2)},\ X \in V^2 \cup \{\Lambda\}\quad Y \in \Gamma \cup \{\Lambda\}$

si $(X,Y) \in \delta(A,b)$ alors :

$(q_Y, X) \in \delta_M(q_\Lambda, A, b)$

b) $\forall B \in \Gamma \quad X \in V^2 \cup \{\Lambda\} \quad Y \in \Gamma \cup \{\Lambda\}$

si $(X,Y) \in \delta(A,B)$ alors :

$(q_Y, X) \in \delta_M(q_B, A, \Lambda)$

(1) cette restriction qui ne restreint pas la généralité, permet de distinguer le cas où nous lisons sur la pile droite une lettre du mot d'entrée où une lettre provenant d'une réduction gauche ou d'un mouvement d'analyse stationnaire.

(2) $\Sigma_{\#_d} = \Sigma \cup \{\#_d\}$

Redisons ici ce qui a été dit dans l'introduction à savoir que M est construit pour reconnaitre L $\#_d$, si L est le langage reconnu par le n.e.d A.

Nous pouvons dresser les deux tableaux groupant tout les cas possibles pour le couple (X,Y) concernant a) et b) de la construction :

$\delta(A,b)$	(xy,Λ)	(x,Λ)	(Λ,y)	(x,y)	(Λ,Λ)
$\delta_M(q_\Lambda, A, b)$	(q_Λ, xy)	(q_Λ, xy)	(q_y,Λ)	(q_y,x)	(q_Λ, Λ)

Tableau 1 pour $b \in \Sigma \cup \{\#_d\}$

$\delta(A,B)$	(xy,Λ)	(x,Λ)	(Λ,y)	(x,y)	(Λ,Λ)
$\delta_M(q_B, A, \Lambda)$	(q_Λ, xy)	(q_Λ, x)	(q_y, Λ)	(q_y, x)	(q_Λ, Λ)

Tableau 2 pour $B \in \Gamma$

Remarque :

Si des instructions du type $\delta(a,b) = \{(\alpha,\Lambda)\}$ avec $|\alpha| \geq 3$ été autorisées notons que nous pourrions aisément les simuler sur un a.p.m en écrivant :

$$\delta_M(q_\Lambda, a, b) = (q_\Lambda, \alpha)$$

ou

$$\delta_M(q_b, a, \Lambda) = (q_\Lambda, \alpha)$$

ceci reviendrait à empiler un mot de Γ^* de longueur strictement supérieur à 2.

c) Les configurations correspondantes sont les suivantes :

. initiales :

$$\begin{cases} \text{dans } A : [\#_g, w\ \#_d] \\ \text{dans } M : [q_\Lambda, \#_g, w\ \#_d] \end{cases}$$

. quelconques : il y a deux types :

t1 dans A : $[\alpha A,\ b\beta]$, $b \in \Sigma \cup \{\#_d\}$

dans M : $[q_\Lambda,\ \alpha A,\ b\beta]$

t2 dans A : $[\alpha A,\ B\beta]$, $B \in \Gamma$

dans M : $[q_B,\ \alpha A,\ \beta]$

nous laissons le soin au lecteur de constater que se trouvant dans deux configurations correspondantes M et A évoluent grâce aux règles a) et b) au travers de configurations correspondantes.

Notons que A possède la seule possibilité de configurations finales suivante :

$$[\Lambda, S] \text{ pour tout } S \in F$$

ce qui correspond dans M à :

$$[q_S,\ \Lambda,\ \Lambda]$$

ceci cadre avec la construction, $q_S \in Q'$

d) <u>Remarque</u> :

La restriction imposée dans la construction concernant la configuration finale vient de ce que si l'on autorise $[S,\Lambda]$ alors M se trouve dans la configuration : $[q_\Lambda,\ S,\Lambda]$ qui doit être finale, c'est-à-dire que q_Λ doit être considéré comme état final possible. Mais alors la configuration $[q_\Lambda,\ \alpha,\Lambda]$ dans M est finale pour $|\alpha| \geqslant 2$ alors que la configuration correspondante dans A, $[\alpha,\Lambda]$, ne l'est pas.

d) Il est immédiat que si A est déterministe il en est de même pour M. En effet nous n'avons jamias le choix entre lire une lettre de Σ ou lire Λ dans M car nous ne pouvons lire une lettre que dans l'état q_Λ et lire Λ seulement dans un état du type q_B avec $B \in \Gamma$.

Par ailleurs il est clair que :

$$\text{pour } b \in \Sigma \quad |\delta(A,b)| = 1 \Longrightarrow |\delta_M(q_\Lambda, A, b)| = 1$$

$$\text{pour } B \in \Gamma \quad |\delta(A,B)| = 1 \Longrightarrow |\delta_M(q_B, A, \Lambda)| = 1$$

nous pouvons donc énoncer le lemme.

Lemme.

Pour tout langage L sur Σ tel que $\#_d \notin \Sigma$ et tout n.e.d (resp. n.e.d det) reconnaissant L, il existe un a.p.m (resp. a.p.m det) reconnaissant L $\#_d$.

Preuve.

Les éléments de la démonstration, qui est évidente, ont été donnés dans ce § et les précédents. Nous laissons le soin au lecteur de la faire dans les détails.

e) L'article GG [10] nous permet d'énoncer le lemme :

Lemme.

Tout langage L sur Σ tel que $c \notin \Sigma$ est reconnu par un a.p.m (resp. déterministe) si et seulement si Lc est reconnu par un a.p.m (resp. déterministe).

Ces deux lemmes nous permettent d'énoncer le théorème conséquence immédiate :

Théorème.

Tout langage L reconnaissable par un n.e.d (resp. déterministe) est reconnaissable par un a.p.m (resp. déterministe).

f) Montrons maintenant que la famille n.e.d.d.$_1$ correspond à la famille Δ_1. Il suffit ici de reprendre exactement la construction proposée pour la démonstration de n.e.d (d) $\subseteq$ a.p.m (d).

Le langage L $\in$ n.e.d.d$_1$ étant reconnu sans marqueur de fond de pile droite nous avons les configurations initiales correspondantes :

dans M : $[q_\Lambda, \#, w]$

dans A : $[\#, w]$

et les configurations finales correspondantes sont :

dans A : $[\Lambda, S]$ dans M : $[q_S, \Lambda, \Lambda]$

dans A : $[S, \Lambda]$ dans M : $[q_\Lambda, S, \Lambda]$

mais pour un automate de la famille n.e.d.d$_1$, nous savons que nous pouvons nous limiter à la configuration finale $[S,\Lambda]$ ce qui démontre l'inclusion, n.e.d.d n.e.d.d$_1$ $\subseteq$ Δ_1. La configuration finale dans M étant $[q_\Lambda, S, \Lambda]$, M réinitialise sa pile.

g) La famille n.e.d.d$_2$ correspond à la famille Δ_2. Ceci est immédiat. Reprenons ce qui a été dit pour n.e.d.d$_1$ $\subseteq$ Δ_1.

Ici la seule configuration finale possible est :

$[\Lambda, S]$ dans A

ce qui correspond dans M à :

$[q_S, \Lambda, \Lambda]$.

Donc n.e.d.d$_2$ $\subseteq$ Δ_2.

h) Nous allons résumer en un théorème les résultats des parties II.4, II.5 et II.6 concernant la comparaison des a.p.m et des n.e.d.

Théorème.

La famille des langages reconnus par les automates à piles de mémoire (resp. déterministes) coïncide avec la famille des langages reconnues par les automates à piles communicantes non expensives à droite (resp. déterministe).

De plus les familles Δ_1 et Δ_2 définies § II.5 a coïncident respectivement avec les familles n.e.d.d_1 et n.e.d.d_2 définies § I.2 f, aux langages contenant la chaîne vide près, qui ne peuvent appartenir aux familles n.e.d.d_1 et n.e.d.d_2.

II.7 AUTOMATES A PILES COMMUNICANTES ET LANGAGES SIMPLES

Nous allons montrer dans cette partie que moyennant certaines restrictions, il existe un a.p.c qui reconnaît exactement la famille des langages simples.

Rappellons d'abord la définition des automates simples.

a) *Définition.*

Un automate à pile déterministe simple est un a.p.m.d avec les trois restrictions suivantes :

1 - $\forall q, \forall Z \; \delta(q,Z,\Lambda) = \emptyset$

2 - un mot w est accepté si et seulement si :

$$\exists (q_o,Z_o,w) \overset{*}{\vdash} (q,\ \Lambda,\ \Lambda) \text{ pour } q \in Q'$$

3 - l'automate n'a qu'un seul état c'est-à-dire

$|Q| = 1.$

Nous appelerons un tel automate un *automate simple*.

Définition.

Un langage est dit *langage simple* si il est reconnu par un automate simple.

Nous conviendrons, l'automate simple n'ayant qu'un seul état de ne pas le faire figurer dans les configurations, une configuration sera donc de la forme :

$$[\gamma,w] \quad \gamma \in \Gamma^* \text{ et } w \in \Sigma^*$$

une configuration finale sera de la forme :

$$[\Lambda,\Lambda].$$

De la même manière dans l'expression des règles de l'automate simple nous ne ferons pas figurer l'état une règle s'écrira donc ; l'automate étant déterministe :

$$\delta(Z,a) = (\gamma)$$

ce qui est équivalent à :

$$\delta(q,Z,a) = (q,\gamma).$$

b) *Théorème*.

La classe des automates simples est identiques à la classe des automates à piles communicantes simples.

Preuve.

Nous donnons la construction étant donné un automate simple, qui permet d'associé un a.p.c.s qui reconnaît le même langage et réciproquement.

Notons δ_M lorsqu'il s'agit de l'application de l'a.p.m et δ_A lorsqu'il s'agit de l'application de l'a.p.c.

Soit $M = <\Sigma_M, \Gamma_M, \delta_M, Z_o>$ et $A = <\Sigma_A, \Gamma_A, \#, \delta_A, A_x>$.

I Donnons tout d'abord A, connaissant M.

Soit $E = \{v / \exists a \in \Sigma, \exists Z \in \Gamma, \exists u \in \Gamma^* \text{ tels que } (vu) = \delta(Z,a)\}$ alors pour tout $\beta \in E$.

1 - $\delta_M(Z,a) = (\gamma) \Longrightarrow \delta_A((\beta Z),a) = ((\beta)\ (\gamma),\Lambda)$

2 - $\delta_M(Z,a) = (\gamma) \Longrightarrow \delta_A((\beta Z),a) = ((\beta),\Lambda)$.

Si $Z \in T_r(Z_o)$ alors :

3 - $\delta_M(Z,a) = (\Lambda) \Longrightarrow \delta_A((Z),a) = (\Lambda,A_x)$

Notons que dans la règle 2 si $\beta = \Lambda$ il vient :

$$\delta_M(Z,a) = (\Lambda) \Longrightarrow \delta_A((Z),a) = (\Lambda,\Lambda)$$

Les configurations correspondantes sont si $\# = Z_o$.
Les configurations initiales dans M : $[Z_o,w]$
dans A : $[Z_o,w]$.

Les configurations initiales sont donc absolument identiques si l'on prend $\# = Z_o$ pour l'a.p.c associé.

configurations quelconques non finales correspondantes :

dans M : $[\gamma,w]$
dans A : $[(\gamma_1)\ \ldots\ (\gamma_n),w]$ avec $\gamma_1\gamma_2 \ldots \gamma_n = \gamma$.

Dans ces conditions nous avons :

- $\Sigma_A = \Sigma_M$

- $\Gamma_A = \{(u) / \exists v \in \Gamma_M^*, \exists Z,a \text{ tq } (\gamma) = \delta_M(Z,a)\} \cup \{A_x\}$.

La règle 3 de la construction assure que lorsque M vide sa pile, c'est-à-dire lorsque : $\delta_M(Z,a) = (\Lambda)$ Z étant un symbole trace de Z_o, alors A passe dans sa configuration finale $[\Lambda, A_x]$ de laquelle aucun mouvement n'est plus possible, conformément à la définition des a.p.c.s. Les configurations finales sont donc en correspondances :

dans M $[\Lambda,\Lambda]$

dans A $[\Lambda, A_x]$.

II Partons maintenant d'un a.p.c.s A, et construisons l'a.p.m M.

Nous avons :

$$\delta_A(X,a) = (X'Y',\Lambda) \Longrightarrow \delta_M(X,a) = (X'Y')$$

$$\delta_A(X,a) = (X',\Lambda) \Longrightarrow \delta_M(X,a) = (X')$$

$$\delta_A(X,a) = (\Lambda,\Lambda) \Longrightarrow \delta_M(X,a) = (\Lambda)$$

ce qui peut se résumer en une seule règle

$$\forall \gamma \in \Gamma^* \cup \Sigma^* \cup \{\#\}^0$$

1 - $\delta_A(X,a) = (\gamma,\Lambda) \Longrightarrow \delta_M(X,a) = (\gamma)$
de plus.

2 - $\delta_A(X,a) = (\Lambda, A_x) \Longrightarrow \delta_M(M,a) = (\Lambda)$.

Posons ici $Z_o = \#$, il vient pour les configurations :

- les configurations initiales sont les mêmes $[\#,w]$
- les configurations quelconques non finales sont également identiques
- la règle 2 permet d'établir la correspondance entre les configurations finales.

Nous avons de plus :

. $\Sigma_M = \Sigma_A$
. $\Gamma_M = \Gamma_A$.

<u>cqfd</u>.

II.8 AUTOMATES A PILES COMMUNICANTES ET AUTOMATES D'ETATS FINIS

Nous montrons que les automates d'états finis coîncident avec les automates à piles communicantes strictement décroissantes.

a) Rappelons la définition d'un automate d'états fini : a.e.f

Définition.

Un automate d'états fini est un 5-uple $M = (\Sigma_M, Q, \delta_M, q_o, F_M)$.

1 - Q est un ensemble fini : ensemble des états

2 - Σ est un alphabet fini : alphabet d'entrée

3 - δ_M est appelée fonction de transition, c'est une application de $Q \times \Sigma$ dans Q

4 - $q_o \in Q$ est dit état initial

5 - $F_M \subseteq Q$ ensemble des états finaux

Le langage $T(M)$ reconnu par M est défini de la manière suivante :

- nous noterons pour tout $a \in \Sigma$, $w \in \Sigma^*$, $q \in Q$ $[q,aw] \vdash [q',w]$ ssi $\delta_M(q,a) = q'$ (1)

alors :

$$T(M) = \{w / \exists q \in F \quad] \; [q_o,w] \overset{*}{\vdash} [q,\Lambda]$$

$\overset{*}{\vdash}$ étant la clôture transitive de $\vdash$.

b) La simulation des a.e.f par les s.t.d et réciproquement et tout à fait évidente.

En effet le seul mouvement autorisé dans un a.p.c strictement décroissants, et la réduction droite à savoir :

$$\delta(X,Y) = (X',\Lambda)$$

le parallèle avec la ligne (1) est évident.

De plus nous pouvons assimiler le marqueur $\#_g$ avec l'état q_o. Notons que la présence ou non d'un marqueur de fin ne pose pas de problème. Les deux possibilités sont équivalentes.

Nous pouvons énoncer le théorème, dont nous laissons le soin de la démonstration au lecteur.

Théorème.

Les langages reconnus par les automates d'états fini sont exactement ceux reconnu par les automates à piles communicantes strictement décroissantes.

CHAPITRE III

GRAMMAIRES CANONIQUES

III.1 GRAMMAIRES CANONIQUES À STRUCTURE DE PHRASE ASSOCIEES AUX [1] AUTOMATES À PILES COMMUNICANTES

a) Dans cette partie nous allons montrer comment partant des règles de l'a.p.c, nous pouvons définir une grammaire dite canonique qui reconnait le même langage. Ceci nous permettra de retrouver les principales familles de langages, les résultats venant corroborer ceux de la première partie.

Définition.

Soit un a.p.c $A = \langle\Sigma, \Gamma, \delta, M, F\rangle$ simplifié possédant les propriétés suivantes (qui ne restreignent pas la généralité).

P1 Les marqueurs de fond de pile ne sont lus qu'une seule fois.

P2 L'ensemble F est réduit à 1 élément noté Ax qui est tel que

$$\forall X \quad \delta(Ax,X) = \delta(X,Ax) = \emptyset$$

aucun mouvement n'est plus possible à partir de Ax.

P3

1. Si $(u,v) \in \delta(X,Y)$ alors $Y \in T_r(\#_d) \Longrightarrow |v| \geq 1$ ou $u = Ax$

2. Si $(u,v) \in \delta(X,Y)$ alors $X \in T_r(\#_g) \Longrightarrow |u| \geq 1$ ou $v = Ax$

3. (Λ, Ax) ou $(Ax, \Lambda) \in \delta(X,Y) \Longrightarrow X \in T_r(\#_g)$ et $Y \in T_r(\#_d)$

P4 $\forall (X,Y) \; (u,v) \in \delta(X,Y) \Longrightarrow v \in \Gamma^*$

Nous dirons que la grammaire $G = \langle N_G, \Sigma_G, P_G, \sigma\rangle$ est la grammaire canonique associée à l'a.p.c A si :

1 $\Sigma_G = \Sigma$

2 $N_G = \{X_e \;/\; X \in \Sigma \cup \Gamma$ et $e = g$ ou $e = d\}$

(1) On trouvera les définitions concernant les grammaires dans [8]. Nous emploierons les même notations. En particulier N désignera l'ensemble des non terminaux, Σ les terminaux et nous noterons $V = N \cup \Sigma$.

3 $\sigma \equiv Ax_g$ est un axiome si $[Ax,\Lambda]$ est une configuration finale

$\sigma \equiv Ax_d$ est un axiome si $[\Lambda,Ax]$ est une configuration finale.(1)

4 P_G est obtenu à partir de δ de la manière suivante : (nous donnons les règles de P_G assortie de quelques remarques)

$$\forall X,\ X',\ Y',\ Z' \in \Sigma \cup \Gamma,\ \forall Y \in \Gamma,\ \forall a \in \Sigma$$

a)

$(X'Y',Z') \in \delta(X,Y) \Longrightarrow X'_g Y'_g Z'_d \to X_g Y_d$

$(X'Y',Z') \in \delta(X,a) \Longrightarrow X'_g Y'_g Z'_d \to X_g a$

$(X'Y',Z') \in \delta(X,\#_d) \Longrightarrow X'_g Y'_g Z'_d \to a$

$(X'Y',Z') \in \delta(X,\#_d) \Longrightarrow X'_g Y'_g Z'_d \to X_g$

b)

$(X',Y'Z') \in \delta(X,Y) \Longrightarrow X'_g Y'_d Z'_d \to X_g Y_d$

$(X',Y'Z') \in \delta(X,a) \Longrightarrow X'_g Y'_d Z'_d \to X_g a$

$(X',Y'Z') \in \delta(\#_g,a) \Longrightarrow X'_g Y'_d Z'_d \to a$

$(X',Y'Z') \in \delta(X,\#_d) \Longrightarrow X'_g Y'_d Z'_d \to X_g$

c)

$(X'Y',\Lambda) \in \delta(X,Y) \Longrightarrow X'_g Y'_g \to X_g Y_d$

$(X'Y',\Lambda) \in \delta(X,a) \Longrightarrow X'_g Y'_g \to X_g a$

$(X'Y',\Lambda) \in \delta(\#_g,a) \Longrightarrow X'_g Y'_g \to a$

$(X'Y',\Lambda) \in \delta(X,\#_d)$ cas impossible à cause de P3.1

(1) Les deux cas peuvent se produire simultanément, dans ce cas on admettra que $\sigma \to Axg$ et $\sigma \to Axd$ sont des productions de P, σ étant l'axiome de G.

d) $(\Lambda, X'Y') \in \delta(X,Y) \Longrightarrow X'_d Y'_d \to X_g Y_d$

$(\Lambda, X'Y') \in \delta(X,a) \Longrightarrow X'_d Y'_d \to X_g a$

$(\Lambda, X'Y') \in \delta(\#_g, a)$ cas impossible par P3.2

$(\Lambda, X'Y') \in \delta(X,\#_d) \Longrightarrow X'_d Y'_d \to X_g$

e) $(X', \Lambda) \in \delta(X,Y) \Longrightarrow X'_g \to X_g Y_d$

$(X', \Lambda) \in \delta(X,a) \Longrightarrow X'_g \to X_g a$

$(X', \Lambda) \in \delta(\#_g, a) \Longrightarrow X'_g \to a$

$(A_x, \Lambda) \in \delta(X,\#_d) \Longrightarrow Ax_g \to X_g$

f) $(\Lambda, X') \in \delta(X,Y) \Longrightarrow X'_d \to X_g Y_d$

$(\Lambda, X') \in \delta(X,a) \Longrightarrow X'_d \to X_g a$

$(\Lambda, X') \in \delta(\#_g, a)$ est impossible à cause de P3.2

$(\Lambda, X') \in \delta(X,\#_d) \Longrightarrow X'_d \to X_g$

g) $(\Lambda, A_x) \in \delta(\#_g,\#_d) \Longrightarrow Ax_d \to \Lambda$

$(A_x, \Lambda) \in \delta(\#_g,\#_d) \Longrightarrow Ax_g \to \Lambda$

Commentaires sur les propriétés imposées à l'a.p.c.

Nous avons supposé que les marqueurs ne soient lus qu'une seule fois, car la lecture de ceci donne lieu dans P à des règles particulières, dues au fait que nous nous trouvons en début d'analyse, ou que nous sommes pour la première fois à la fin du mot à analyser.

En effet si nous autorisons $(\#_g, \Lambda) \in \delta(\#_g, a)$, $\#_g$ ne signifierai plus nécessairement le début de l'analyse.

.La propriété des traces des marqueurs,[1] vérifiée par tout a.p.c simplifié, permet de donner un sens aux contraintes imposées aux ensembles $T_r(\#_g)$ et $T_r(\#_d)$, dans les propriétés P3.

.La propriété : $\forall X\ \delta(Ax, X) = \delta(X, Ax) = \emptyset$ et $F = \{Ax\}$ impose l'arrêt de l'analyse dès que l'unique élément de F, Ax apparait. La propriété :

$$(\Lambda, Ax) \text{ ou } (Ax, \Lambda) \in \delta(X, Y) \Longrightarrow X \in T_r(\#_g) \text{ et } Y \in T_r(\#_d)$$

indique que Ax ne peut être la réduction que d'éléments, traces des marqueurs, ce qui est légitime étant donné qu'aucun mouvement n'est plus alors possible, à cause de la propriété précédente. Ce sont deux propriétés qui de toute évidence n'atteignent ni la généralité ni un éventuel déterminisme.

.Les propriété P3.1 et P3.2 intéressent respectivement les mouvements

$$[X, Y_1 \ldots Y_n] \vdash [\Lambda, Y'Y_2 \ldots Y_n]$$

et

$$[X_1 \ldots X_n, Y] \vdash [X_1 \ldots X_{n-1}X', \Lambda]$$

mouvements qui sont en fait inutiles dans A, car il aboutissent à des bloquages dans des configurations non finales

.La propriété P4 permet de distinguer le cas ou sur la pile droite on trouve un élément de Σ du mot d'entrée ou un élément venant d'une réduction car ces deux cas donne lieu dans G a des productions différentes.

Disons enfin que toutes ces propriétés sont triviales à assurer pour tout a.p.c et qu'elle n'hypothéquent ni la généralité, ni le déterminisme.

(1) voir chapitre I, § I.3 b).

b) Nous allons reprendre l'exemple 1 donné au § I.2 g), en modifiant légèrement l'a.p.c.b proposé afin qu'il vérifie les propriétés imposées à l'a.p.c dans la définition, et indiquer la grammaire canonique G associée.

Nous avons

$$A = (\Sigma,\ \Gamma,\ \delta,\ M,\ F)$$

avec

$$\Sigma = \{a,b,c\},\quad \Gamma = \{A,\bar{A},X,S,S',M\}^{(1)},\quad M = \{\#_g,\#_d\},\quad F = \{S\}$$

la construction du § a) nous fourni le tableau :

	Règle de A			Production de G
1	$\delta(\#_g,a)$	=	(Ma,Λ) [1]	$M_g a_g \to a$
2	$\delta(a,a)$	=	(aa,Λ)	$a_g a_g \to a_g a$
3	$\delta(a,b)$	=	(A,Λ)	$A_g \to a_g b$
4	$\delta(A,b)$	=	$(\Lambda,\bar{A}A)$	$\bar{A}_d A_d \to A_g b$
5	$\delta(a,\bar{A})$	=	(A,Λ)	$A_g \to a_g \bar{A}_d$
6	$\delta(A,A)$	=	(AA,Λ)	$A_g A_g \to A_g A_d$
7	$\delta(A,c)$	=	(Λ,X)	$X_d \to A_g c$
8	$\delta(A,X)$	=	(X,Λ)	$X_g \to A_g X_d$
9	$\delta(X,c)$	=	(Λ,X)	$X_d \to X_g c$

(1) Voir bas de la page suivante.

10 $\delta(M,X) = (S',\Lambda)^{(1)}$ $\qquad S'_g \to M_g X_d$

11 $\delta(S',\#_d) = (\Lambda,S)$ $\qquad S_d \to S'_g$

12 $\delta(A,\bar{A}) = (\Lambda,\bar{A}A)$ $\qquad \bar{A}_d A_d \to A_g \bar{A}_d$

le lecteur vérifiera aisément que A vérifie toutes les conditions de la construction de § a).

Les autres éléments de la grammaire G sont donc :

$$\Sigma_G = \Sigma,\ N_G = \{M_g, a_g, A_g, \bar{A}_d, A_d, X_d, X_g, S'_g, S_d\},\ \sigma = S_d$$

Les dérivations possibles à partir de S_d, qui est l'axiome de G, et donnant un mot de Σ^*, sont les suivantes. Nous indiquons au dessus de la flèche $\Longrightarrow$, le numéro de la production ayant permi la dérivation.

$$\sigma = S_d \overset{11}{\Longrightarrow} S'_g \overset{10}{\Longrightarrow} F_g X_d = \alpha_1$$

$$\alpha_1 = F_g X_d \overset{7}{\Longrightarrow} F_g A_g c \overset{3}{\Longrightarrow} F_g a_g bc \overset{1}{\Longrightarrow} abc$$

$$\alpha_1 = F_g X_d \overset{9}{\Longrightarrow} F_g X_g c \overset{8}{\Longrightarrow} F_g A_g X_d c = \alpha_2$$

$$\alpha_2 = F_g A_g X_d c \overset{7}{\Longrightarrow} F_g A_g A_g cc \overset{6}{\Longrightarrow} F_g A_g A_d cc \overset{5}{\Longrightarrow} F_g a_g \bar{A}_d A_d cc \overset{4}{\Longrightarrow} F_g a_g A_g bbcc$$

$$\overset{3}{\Longrightarrow} F_g a_g a_g bbcc \overset{2}{\Longrightarrow} F_g a_g abbcc \overset{1}{\Longrightarrow} aabbcc$$

$$\alpha_2 = F_g A_g X_d c \overset{9}{\Longrightarrow} F_g A_g X_g cc \overset{8}{\Longrightarrow} F_g A_g A_g X_d cc = \alpha_3$$

$$\alpha_3 = F_g A_g A_g X_d cc \overset{7}{\Longrightarrow} F_g A_g A_g A_g ccc \overset{6}{\Longrightarrow} F_g A_g A_g A_d ccc \overset{6}{\Longrightarrow} F_g A_g A_d ccc$$

$$\overset{5}{\Longrightarrow} F_g ag\bar{A}_d A_d ccc \overset{12}{\Longrightarrow} F_g a_g \bar{A}_d A_d ccc \overset{4}{\Longrightarrow} F_g a_f A_g A_g bccc$$

(1) Elément modifié afin de respecter la restriction indiquant que les marqueurs ne doivent être lus qu'une seule fois, ici $\#_g$.

$$\overset{6}{\Longrightarrow} F_g a_g A_g A_d bccc \overset{5}{\Longrightarrow} F_g a_g a_g \bar{A}_d bccc \overset{4}{\Longrightarrow} F_g a_g a_g A_g bbccc$$

$$\Longrightarrow F_g a_g a_g a_g bbbccc \overset{*2}{\Longrightarrow} F_g a_g aabbbccc \overset{1}{\Longrightarrow} aaabbbccc$$

$$\alpha_3 = F_g A_g A_g X_d cc \overset{9}{\Longrightarrow} F_g A_g A_g X_g ccc \overset{8}{\Longrightarrow} F_g A_g A_g A_g X_d ccc = \alpha_4$$

il est alors facile de montrer que

$$\alpha_4 \overset{*}{\Longrightarrow} \alpha_5 = F_g A_g^4 X_d c^4 \text{ et } \alpha_4 \overset{*}{\Longrightarrow} a^4 b^4 c^4$$

et que pour tout n

$$\alpha_n \overset{*}{\Longrightarrow} \alpha_{n+1} = F_g A_g^n X_d c^n \text{ et } \alpha_n \overset{*}{\Longrightarrow} a^n b^n c^n$$

donc :

$$\{a^n b^n c^n \ / \ n \geq 1\} \subseteq L(G)$$

Il serait un peu long mais pas difficile de démontrer que les dérivations envisagées sont les seules donnant naissance à des mots de Σ^*. Et donc :

$$L(G) = T(A) = \{a^n b^n c^n \ / \ n \geq 1\}$$

c) Nous amorçons dans ce paragraphe la démonstration de l'équivalence entre la grammaire canonique et l'a.p.c.

Définition.

Nous dirons qu'un mot α de V_G^* est une expression bien formée s'il est d'une des formes suivantes :

$$Y^i, X^i \in \Gamma / \{A_x\} \text{ et } a_i \in \Sigma \text{ pour tout } i$$

1 Pour $m \geq 1,\ n \geq 0,\ p \geq 0$

$$\alpha = X_g^1 \ldots X_g^i \ldots X_g^m\, Y_d^1 \ldots Y_d^n \;\; a_1 \ldots a_k \ldots a_p$$

avec : $.\ X^1 \in T_r(\#_g)$ et $\forall i \neq 1\ X^i \notin T_r(\#_g)$

$.\ Y^n \in T_r(\#_d) \Longrightarrow p=0$ (i.e. $a_1 \ldots a_p = \Lambda$)

$.\ \forall i \neq n\ \ Y^i \notin T_r(\#_d)$

2 $\alpha = AX_d,\ \alpha = AX_g$

3 $\alpha \in \Sigma_G^*$

Donnons à la suite de cette définition deux propriétés et un corollaire utiles par la suite.

Propriété A.

Pour tout $u, v \in V_G^*\ \ X, X' \in \Gamma \cup \Sigma$

$$(X_g u \overset{*}{\Longrightarrow} X'_x v) \Longrightarrow x = g \text{ ou } (X'_x \in \Sigma \text{ et } X \in T_r(\#_g))$$

Preuve.

Nous allons procéder par récurrence.

La propriété est vraie pour une dérivation de longueur nulle.

Supposons la propriété vraie pour une dérivation de longueur n.

Soit :

$$X_g u \overset{n+1}{\Longrightarrow} X'_x v.$$

Nous pouvons décomposer cette dérivation de la façon suivante :

$$X_g u \Longrightarrow Y_y u' \overset{n}{\Longrightarrow} X'_x v$$

Deux cas :

1 y=g. Il vient la propriété étant vraie pour n :

$$x = g \text{ ou } (X'_x \in \Sigma \text{ et } Y \in T_r(\#_g)).$$

Mais $Y \in T_r(\#_g) \Longrightarrow X \in T_r(\#_g)$, en effet dans la dérivation $X_g u \Longrightarrow Y_g u'$, soit $X_g = Y_g$, soit elle est obtenu en appliquant une règle provenant de :

$$(X\alpha, \beta) \in \delta(Y, Y')$$

Donc : soit x = y

soit $X'_x \in \Sigma$ et $Y \in T_r(\#_g)$ et $X \in T_r(\#_g)$.

2 $Y_y \in \Sigma$, alors $Y_y = X'_x$ donc $X'_x \in \Sigma$

soit $Y_y = a$. $X_g u \Longrightarrow au'$ ne peut se faire que par application d'une production du type

$$X_g\alpha \to a\alpha'$$

à savoir type a3, b3, c3 ou e3, dans tout ces cas cela correspond dans A à la lecture de $\#_g$ donc $X \in T_r(\#_g)$.

Notons que $y \neq d$, c'est une conséquence directe de la forme des productions de P

cqfd.

Propriété B.

Pour tout $u, v \in V_G^*$, $X, X' \in \Gamma \cup \Sigma$
$(uX_d \overset{*}{\Longrightarrow} vX'_x) \Longrightarrow x=d$ ou $X'_x \in \Sigma$ ou $(x=g$ et $X \in T_r(\#_d))$.

<u>Preuve</u>.

Nous allons ici aussi procéderpar récurrence. La propriété est vraie pour une dérivation de longueur nulle.

Supposons la vraie pour une dérivation de longueur n.

Soit :

$$uX_d \xRightarrow{n+1} vX'_x$$

Nous pouvons décomposer cette dérivation comme suit :

$$uX_d \Longrightarrow u'Y_y \xRightarrow{n} vX'_x$$

Trois cas :

1 y=d. Nous appliquons l'hypothèse de récurrence. Il vient :

$$x=d \text{ ou } X_x \in \Sigma \text{ ou } (X=g \text{ et } \dot{Y} \in T_r(\#_d))$$

mais une argumentation semblable à cette employée pour la propriété A, permet d'affirmer que :

$$(uX_d \Longrightarrow u'Y_d) \Longrightarrow (Y \in T_r(\#_d) \Longrightarrow X \in T_r(\#_d))$$

donc si $x = g$ on a bien $X \in T_r(\#_d)$.

2 $Y_y \in \Sigma$. Alors $X'_x \in \Sigma$ de manière triviale.

3 y = g. Mais alors : $uX_d \Longrightarrow u'Y_g$ ne peut être obtenu qu'en appliquant une production du type a4, b4, d4 ou f4, qui sont issues d'un mouvement par lecture de $\#_d$ dans A ce qui implique $X \in T_r(\#_d)$.
Donc, la propriétée est vraie, quelque soit X'_x.

<u>cqfd</u>.

Corollaire A.

Pour $u \in V_G$ si u est de la forme suivante

$$u = u_1 \, X_d \, Y_g \, v_1 \text{ avec } X \notin T_r(\#_g) \text{ et } Y \notin T_r(\#_d)$$

alors $\nexists w \in \Sigma_G^*$ tel que : $u \overset{*}{\Longrightarrow} w$

Preuve.

D'après la proposition A :

si $Y_g v_1 \overset{*}{\Longrightarrow} Y'_y v'_1$, comme $Y \notin T_r(\#_d)$ on a nécessairement $y = g$.

Et d'après la proposition B :

si $u_1 X_d \overset{*}{\Longrightarrow} u'_1 X'_x$ alors soit $x = d$, soit $X_x \in \Sigma$ car $X \notin T_r(\#_g)$.

Donc :

$$\forall v \text{ tel que : } u \overset{*}{\Longrightarrow} v$$

v est de la forme :

$$u'_1 \; a \; Y'_g \; v'_1 \text{ ou } u'_1 \; X'_d \; Y'_g \; v'_1$$

avec $Y' \notin T_r(\#_g)$ car $Y \notin T_r(\#_g)$ il est immédiat dans ces conditions que :

$$\nexists w \in \Sigma_G^* \text{ tel que : } u \overset{*}{\Longrightarrow} w,$$

en effet il faudrait résoudre Y'_g en un mot terminal ce qui n'est pas possible car le symbole immédiatement à sa gauche ne pourra jamais être de la forme Z_g, ce qui est la seule possibilité de dériver Y'_g en un mot terminal.

cqfd.

d) Lemme A.

Pour toute suite $(\alpha_i)_{i=1}^{n}$ de mot de V_G^* on a :

si $\exists w \in \Sigma_G^*$ tel que : $\sigma \Longrightarrow \alpha_1 \Longrightarrow \dots \Longrightarrow \alpha_n \Longrightarrow w$

alors pour tout i, $1 \leq i \leq n$ x_i est bien formée.

Preuve.

Nous allons montrer que α_1 est bien formé, puis nous montrerons que si α_i est bien formée alors α_{i+1} est bien formé.

. α_1 est bien formée. L'axiome peut être de deux formes soit Ax_g soit Ax_d il vient 5 possibilités pour α_1 :

1) par e1[1]$Ax_g \to x_g Y_d = \alpha_1$ grâxe à $(A,\Lambda) \in \delta(X,Y)$

et P3.3 implique $X \in T_r(\#_g)$ et $Y \in T_r(\#_d)$, α_1 est donc bien formé

2) par e4 $Ax_g \to X_g = \alpha_1$ grâce à $(A,\Lambda) \in \delta(X,\#_d)$
et P3.3 implique $X \in T_r(\#_g)$, α_1 est bien formé

3) par f1 : même chose que pour e1 avec $Ax_d \to X_g Y_d = \alpha_1$

4) par f4 : même chose que pour e4 avec $Ax_g \to X_g = \alpha_1$

5) par g : $Ax_d \to \Lambda$ ou $Ax_g \to \Lambda$, $\Lambda \in \Sigma_G^*$ est bien formée.

. Supposons maintenant que α_i est bien formée.

Il suffit de répertorier les cas ou α_{i+1} n'est pas bien formée et de montrer que dans ces cas, on ne peut aboutir à un mot w de Σ_G^*.

(1) e1 fait référence, à la première ligne de e) dans la construction de la grammaire canonique, § a). De même par la suite pour e4, f1, ... etc..

Soit $\alpha_i = X_g^1 \ldots X_g^i \ldots X_g^m Y_d^1 \ldots Y_d^i \ldots Y_d^n a_1 \ldots a_k \ldots a_p$

avec

$. X^1 \in T_r(\#_g)$ et $\forall i \neq 1\ X_i \notin T_r(\#_g)$

$. Y^n \in T_r(\#_d) \Longrightarrow p = o$

$. \forall i \neq n\ \ Y^i \notin T_r(\#_d)$

Remarquons que l'application d'une des règles issues de a) ou b) implique que α_{i+1} est bien formé.

Qu'il en est de même pour les règles c) d) c) ou f) appliquées respectivement à X_g^{m-1}, X_g^m, $Y_d^1 Y_d^2$, X_g^m ou Y_d^1, en remarquant que

c3 est applicable si $X^1 \ldots X^{m-2} = \Lambda$ afin que $X^{m-1} \in T_r(\#_g)$

d4 est applicable si $Y_d^3 \ldots Y_d^n a_1 \ldots a_p = \Lambda$ afin que $Y^2 \in T_r(\#_d)$

même chose pour e3 et e4.

Envisageons à présent les cas donnant au α_{i+1} non bien formée.

Nous distinguons 6 cas :

1) Nous pouvons alors appliquer à $X_g^i X_g^{i+1}$, c1, ou à X_g^{i+1}, e1, soit :

$$X_g^i X_g^{i+1} \rightarrow A_g B_d \text{ ou } X_g^{i+1} \rightarrow A_g B_d$$

dans ces deux cas α_{i+1} est de la forme :

$$\alpha_{i+1} = \ldots X_g A_g\ B_d\ Y_g \ldots$$

qui ne peut se dériver en un mot de Σ_G^*. (voir le corollaire A).

2) Nous pouvons également appliquer à $X_g^i X_g^{i+1}$, c2, ou à X_g^{i+1}, e2, nous nous trouvons dans les deux cas avec un α_{i+1} de la forme :

$$\ldots A_g a\ Y_g \ldots$$

Or la propriété A indique que $Y_g \ldots \overset{*}{\Longrightarrow} a \ldots$, seulement si (Propriété A) $Y \in T_r(\#_g)$ ce qui n'est pas le cas, ceci étant la seule solution pour aboutir à w, α_{i+1} ne peut aboutir à w.

3) Les solutions c3, $X_g^i X_g^{i+1} \rightarrow a$ et e3, $X_g^i \rightarrow a$ sont inapplicables si $X^i \notin T_r(\#_g)$, et si $\alpha_i = X_g^i \ldots X_g^m u$ c'est-à-dire $X^i \in T_r(\#_g)$ alors α_{i+1} est de la forme :

$$\alpha_{i+1} = aY_g \ldots$$

et pour les même raison que précédemment à savoir

$$Y_g \ldots \overset{*}{\Longrightarrow} a \ldots$$

si $Y \in T_r(\#_g)$ d'après la propriété A α_{i+1} ne peut se dériver en w.

4) Nous pouvons appliquer une règle du type d1 à $Y_d^i Y_d^{i+1}$ ou du type f1 à Y_d^{i+1} ce qui nous donne dans les deux cas α_{i+1} de la forme :

$$\ldots X_d\ A_g\ B_d\ Y_d \ldots$$

Nous sommes dans le même cas que précédemment à savoir X_d et A_g consécutifs. Nous savons que la présence de ce cas ne permet pas la dérivation de α_{i+1} en w.

5) Nous pouvons appliquer une règle du type d2 à $Y_d^i\ Y_d^{i+1}$ ou du type f2 à Y_d^{i+1} ce qui nous donne dans ces deux cas un α_{i+1} de la forme :

$$\ldots X_d\ A_g\ a\ Y_d \ldots$$

même cas que précédemment, X_d et A_g sont concécutifs.

6) Les règles de type d4 et f4 sont inapplicables car $Y^{i+1} \notin T_r(\#_d)$.

Si $\alpha_i = uY_d^1 \ldots Y_d^i Y_d^{i+1}$, $Y^{i+1} \in T_r(\#_d)$ alors d4 est applicable à $Y_d^i\ Y_d^{i+1}$ et f4 à Y_d^{i+1} en donnant α_{i+1} de la forme

$$\ldots X_d Y_g$$

nous sommes ramener au cas précédent.

cqfd.

e) Nous allons maintenant définir une correspondance entre les expressions bien formées de G et les configurations dans l'a.p.c.

Définition.

Nous dirons qu'une expression bien formée α de G et une configuration K de l'a.p.c sont correspondantes, nous noterons $\alpha \longleftrightarrow K$, si nous sommes dans un des cas suivants :(1)

type 1 $\quad m \geq 1,\ n \geq 0,\ p \geq 1$

$$\alpha = X_g^1 \ldots X_g^m Y_d^1 \ldots Y_d^n a_1 \ldots a_p$$

$$K = [X^1 \ldots X^m,\ Y^1 \ldots Y^n\ a_1 \ldots a_p\ \#_d]$$

type 2 $\quad m \geq 1,\ n \geq 1$

$$\alpha = X_g^1 \ldots X_g^m Y_d^1 \ldots Y_d^n$$

21 $\quad$ Si $Y^n \in T_r(\#_d)$ $\quad K = [X^1 \ldots X^m,\ Y^1 \ldots Y^n]$

22 $\quad$ Si $Y^n \notin T_r(\#_d)$ $\quad K = [X^1 \ldots X^m,\ Y^1 \ldots Y^n\ \#_d]$

type 3 $\quad m \geq 1$ et si $m = 1$ $X^1 \neq A_X$

$$\alpha = X_g^1 \ldots X_g^m \text{ et } \qquad K = [X^1 \ldots X^m,\ \#_d]$$

(1) Rappelons que pour tout i : $X^i, Y^i \in \Gamma \,/\, \{A_X\}$ et $a_i \in \Sigma$.

type 4

41 $\alpha = Ax_d$ $K = [\Lambda, A_x]$

42 $\alpha = Ax_g$ $K = [A_x, \Lambda]$

type 5 $\alpha \in \Sigma_G^*$ $K = [\#_g, \alpha\#_d]$

Notons que cette correspondance est biunivoque, notamment grâce aux propriétés imposées à l'a.p.c.

Exemple.

Nous allons reprendre l'exemple en b), en mettant en parallèle les dérivations dans G et les mouvements dans A. Le lecteur constatera les correspondances. Nous le faisons pour w = abc.

Dans G		Dans A	Productions	Règles employées
$[\#_g, abc\#_d]$	$\overset{1}{\longleftrightarrow}$	abc		
$\top$		$\Uparrow$		1
$[Fa, bc\#_d]$	$\overset{2}{\longleftrightarrow}$	$F_g a_g b_g c$		
$\top$		$\Uparrow$		3
$[FA, c\#_d]$	$\overset{3}{\longleftrightarrow}$	$F_g A_g c$		
$\top$		$\Uparrow$		9
$[F, X\#_d]$	$\overset{4}{\longleftrightarrow}$	$F_g X_d$		
$\top$		$\Uparrow$		10
$[S', \#_d]$	$\overset{5}{\longleftrightarrow}$	S'_g		
$\top$		$\Uparrow$		11
$[\Lambda, S]$	$\overset{6}{\longleftrightarrow}$	S_d		

Les correspondances 1, 2 et 3 correspondent au type 1.
La correspondance 4 est de type 2.2 car $X \notin T_r(\#_d)$. La correspondance
La correspondance 5 est de type 3, la 6 de type 4, S étant l'axiome.

Lemme.

Pour toutes expressions bien formées α, α' et toutes configurations K, et K' de l'a.p.c telles que :

$$K \longleftrightarrow \alpha \text{ et } K' \longleftrightarrow \alpha'$$

on a :

$$(\alpha \Longrightarrow \alpha') \Longleftrightarrow (K' \vdash K)$$

Preuve.

Ce lemme est une conséquence directe de la construction et des définitions. Il est évident dans presque tout les cas, en particulier lorsque K et α d'une part et K', α' d'autre part sont en correspondances de type 1, ou de type 2 avec $n \geq 1$, nous laissons au lecteur le soin de le constater.

. Envisageons le cas ou :

$$K' = [X^1 \ldots X^m X,\ \#_d] \vdash K = [X^1 \ldots X^m, Y]$$

grâce à

$$(\Lambda, Y) \in \delta(X,\ \#_d),$$

notons que $Y \in T_r(\#_d)$ il vient :

$$p : (Y_d \to X_g) \in P$$

et

$$K \longleftrightarrow \alpha = X_g^1 \ldots X_g^m Y_d \text{ avec } Y \in T_r(\#_d)$$

$$K' \longleftrightarrow \alpha' = X_g^1 \ldots X_g^m X_g \text{ correspondance de type 3.}$$

On a bien $\alpha \Longrightarrow \alpha'$ par la règle p . Réciproquement si $\alpha' \Longrightarrow \alpha$ par p, il est facile de voir que nécessairement $(\Lambda, Y) \in \delta(X, \#_d)$, ceci est dû à la forme des règles de P, et donc $K' \vdash K$.

Nous avons ici appliquer une production du type f4, nous obtenons des cas semblables par applications de productions du type a4 b4 c4 d4 ou e4.

Tout les autres cas se traitent de la même manière, sans difficultés particulières.

cqfd.

f) Nous pouvons énoncé à la suite de cela, le corollaire qui pose l'équivalence de la grammaire canoniques et de l'automate.

Corollaire.

$$L(G) = T(A)$$

Preuve.

Ce théorème est une conséquence des 2 lemmes précédents.

En effet :

$\amalg$ Montrons que $L(G) \subseteq T(A)$

Soit :

$$\sigma \Longrightarrow \alpha_1 \Longrightarrow \alpha_2 \Longrightarrow \dots \Longrightarrow \alpha_n = w \quad w \in \Sigma_G^*$$

nous savons d'après le lemme du §c) qu'alors la suite des $(\alpha_i)_{i=1}^n$ est une suite l'expressions bien formée.

Or la correspondance définie au §d) permet d'associer à cette suite d'expression $(\alpha_i)_{i=1}^n$, une suite de configurations $(K_i)_{i=1}^n$, configurations de l'a.p.c telles que, pour tout i $K_i \longleftrightarrow \alpha_i$. Enfin le lemme du §d), nous permet d'écrire pour tout i :

$$(\alpha_i \Longrightarrow \alpha_{i+1}) \Longrightarrow (K_{i+1} \vdash K_i)$$

Donc : si $\sigma \equiv Ax_g$, $K_o = [Ax, \Lambda]$, de plus $K_n = [\#_g, w\#_d]$ et

$$K_n \vdash K_{n-1} \vdash \dots \vdash K_o = [Ax, \Lambda]$$

donc $w \in T(A)$

II $T(A) \subseteq L(G)$, ceci ce fait de la même manière.

cqfd.

Remarque :

La correspondance entre les expressions bien formées et les configurations de l'a.p.c étant biunivoque, ceci implique que si l'a.p.c est déterministe, il sera possible d'analyser tout mot de L(G), par une analyse des feuille vers la racine, et cette analyse sera unique pour tout mot.

g) Nous concluons cette partie en énonçant le théorème principal, il est une conséquence directe des § précédents.

Théorème.

A tout a.p.c A, on sait associer une grammaire à structure de phrase, donnée par la construction en a) qui reconnait le même langage.

Preuve.

Il suffit de montrer qu'à tout a.p.c on peut associer un a.p.c équivalent vérifiant les propriétés de la construction du §a). Ceci est très facile à faire. Nous laissons du lecteur le soin de la constater.

cqfd.

h) <u>Remarque</u> :

La suppression des mouvements croissants, suprime de G les productions pour lesquelles la partie gauche est de longueur strictement supérieure à celle de la partie droite, mis à part les 3^{iem} production de c) et 4^{iem} de d) dont il est facile de voir qu'elles peuvent être remplacées. Cette suppression fait de G une grammaire à contexte lié, confirmant de cette façon le résultat obtenu en II.3 en ce qui concerne le cas non déterministe.

Maintenant si nous supprimons les mouvements croissants à droite et les mouvements d'analyse droite. Nous savons que l'a.p.c ainsi obtenu reconnait les langages algébriques (remarque II.6.d) cependant G n'est pas algébrique . Mais nous pouvons faire la remarque suivante :

tout mot de L(G) est dans ce cas générable par une dérivation la plus à droite. En effet il suffit de constater qu'à droite du sommet de pile droite ne peut figurer qu'un mot de Σ^*, donc d'après la correspondance énoncer au paragraphe b), ceci implique que la dérivation dans G correspondante au mouvement dans A est la plus à droite. Nous pouvons alors utilisé le résultat énoncé par Ginsburg et Greibach dans [9] à savoir que l'ensemble L tel que

$$L = \{w \ / \ \exists \ \sigma \overset{* \, d}{\Longrightarrow} w\}$$

$\sigma \overset{d}{\Longrightarrow} w$ étant la dérivation la plus à droite, est algébrique la dérivation étant opérés dans une grammaire à structure de phrase quelconque.

Ajoutons qu'à l'aide des a.p.c et de leurs grammaires associées nous pourrions montrer ce résultat. Constatons qu'ils en donnent une interprétation tout à fait claire.

III.2 GRAMMAIRE CANONIQUE ALGEBRIQUE ASSOCIEE A TOUT AUTOMATE A PILES COMMUNICANTES NON EXPENSIVES A DROITE

Nous donnons dans cette partie la construction de la grammaire canonique algébrique associée à tout n.e.d. Puis nous montrons que la grammaire et l'automate définissent le même langage.

Nous menons de front le cas général, dans lequel le marqueur $\#_d$ est utilisé, cas où les configurations initiales sont de la forme $[\#_g, w\#_d]$ et le cas ou $\#_d$ n'est pas utilisé. Ceci afin d'obtenir les grammaires correspondantes aux familles n.e.d.d_1 et n.e.d.d_2.

a) *Définition.*

Soit un n.e.d $A = <\Sigma, \Gamma, \delta, M, F>$ simplifié (voir I.3.c) possédant en plus les propriétés suivantes :

P1 Pour tout $A, B, X \in \Sigma' \cup \Gamma \cup M$

1 $(\Lambda,X) \in \delta(A,B) \Longrightarrow X \notin \Sigma \cup \{\#_d\}$ (1)

2 $(X,\Lambda) \in \delta(A,B) \Longrightarrow X \neq \#_g$

P2 Si le marqueur $\#_d$ est utilisé :

1 $F = \{S\}$ et $S \in T_r(\#_d)$

2 Si $B \in T_r(\#_d)$, $\nexists X', Y \in \Gamma, Y' \in \Gamma \cup \{\Lambda\}$ tels que :

$$(Y'X',\Lambda) \in \delta(Y,B)$$

P3

$$(X'Y',\Lambda) \in \delta(X,Y) \Longrightarrow X'=X \quad \text{(voir I.3.d)}$$

(1) si le marqueur $\#_d$ n'est pas utilisé $X \notin \Sigma \cup \{\#_d\}$ devient $X \notin \Sigma$.

Nous dirons que $G = \langle N_G, \Sigma_G, P, \sigma \rangle$ est la grammaire algébrique associée au n.e.d A si :

1 $\Sigma'_G = \Sigma$

2 $V_G = \Sigma \cup \{(A,B) \;/\; \exists X \in \Gamma^0 \cup \{\#\} \quad Y \in \Gamma \cup \{\#\}$

$A = X_g$ et $B = Y_d$ ou $Y_g\}$

P_G est obtenu à partir de δ de la manière suivante :

$A, X, B' \in \Gamma \cup \Sigma,\ B \in \Gamma,\ a \in \Sigma,\ Y \in \Gamma \cup \{\Lambda\} \cup M$

11 $(X,\Lambda) \in \delta(A,B) \Longrightarrow \forall Y, (Y_g, X_g) \to (Y_g, A_g)\ (A_g, B_d)$

12 $(\Lambda,X) \in \delta(A,B) \Longrightarrow \forall Y, (Y_g, X_d) \to (Y_g, A_g)\ (A_g, B_d)$

13 $(AB',\Lambda) \in \delta(A,B) \Longrightarrow (A_g, B'_g) \to (A_g, B_d)$

21 $(X,\Lambda) \in \delta(A,a) \Longrightarrow \forall Y, (Y_g, X_g) \to (Y_g, A_g)\, a$

22 $(\Lambda,X) \in \delta(A,a) \Longrightarrow \forall Y, (Y_g, X_d) \to (Y_g, A_g)\, a$

23 $(AB',\Lambda) \in \delta(A,a) \Longrightarrow (A_g, B'_g) \to a$

31 $(X,\Lambda) \in \delta(\#_g, a) \Longrightarrow (\Lambda, X_g) \to a$

32 $(\Lambda,X) \in \delta(\#_g, a) \Longrightarrow (\Lambda, X_d) \to a$

33 $(\#_g A,\Lambda) \in \delta(\#_g, a) \Longrightarrow (\#_g, A_g) \to a$

41 $(X,\Lambda) \in \delta(\#_g,B) \Longrightarrow (\Lambda,X_g) \rightarrow (\#_g,B_d)$

42 $(\Lambda,X) \in \delta(\#_g,B) \Longrightarrow (\Lambda,X_d) \rightarrow (\#_g,B_d)$

43 $(\#_gA',\Lambda) \in \delta(\#_g,B) \Longrightarrow (\#_g,A'_g) \rightarrow (\#_g,B_d)$

5 $(\Lambda,X) \in \delta(A,\#_d) \Longrightarrow \forall Y,(Y_g,X_d) \rightarrow (Y_g,A_g)$

6 $(\Lambda,X) \in \delta(\#_g,\#_d) \Longrightarrow (\Lambda,X_d) \rightarrow \Lambda$

$(X,\Lambda) \in \delta(\#_g,\#_d) \Longrightarrow (X_g,\Lambda) \rightarrow \Lambda$

7 Pour tout $X \in F$

si $[\Lambda,X]$ est une configuration finale $\sigma \rightarrow (\Lambda,X_d)$

si $[X,\Lambda]$ est une configuration finale $\sigma \rightarrow (\Lambda,X_g)$

<u>Commentaire sur les propriétés.</u>

. La propriété P1.1 implique deux choses d'une part que $\#_d$ ne soit lu qu'une seule fois, d'autre part que tout terminal au sommet de la pile droite, n'est pas issu d'une réduction, mais est une lettre du mot à analyser. Ceci car, $\#_d$, comme les éléments de Σ sur la pile droite donne lieu à des productions particulières dans G (voir les cas 2i, 5 et 6).
Même commentaire pour la propriété P2.2, avec les règles 3i et 4i.

. Les propriétés P2 sont évidentes à assurer pour tout n.e.d.
Elles sont souvent utiles dans les démonstrations qui suivent.
Elles permettent d'éviter la forme d'erreur suivante :

Soit tel que :

1 $\delta(\#_g,a) = (X,\Lambda)$;

2 $\delta(X,\#_d) = (S,\Lambda)$;

3 $(S,b) = (S,\Lambda)$;

S étant un élément unique de F. On constate que T(A) = {a} et que la règle 3 est inutile, car le mouvement suivant est impossible :

$$[\#_g, a\#_d b] \vdash [X, \#_d b] \vdash [S,b] \vdash [S,\Lambda]$$

en effet $[\#_g, a\#_d b]$ n'est pas une configuration initiale possible. Cependant les règles de contruction de G nous donnent :

1 $(\Lambda, X_g) \to a$

2 $(\Lambda, S_g) \to (\Lambda, X_g)$

3 $(\Lambda, S_g) \to (\Lambda, S_g)b$

$[S,\Lambda]$ étant une configuration finale $\sigma \to (\Lambda, S_g)$ il vient la possibilité dans G :

$$(\Lambda, S_g) \Longrightarrow (\Lambda, S_g)b \overset{*}{\Longrightarrow} (\Lambda, S_g)b^* \Longrightarrow (\Lambda, X_g)b^* \Longrightarrow ab^*$$

or

$$\forall n \neq 0 \quad ab^n \notin T(A)$$

b) A ce stade nous allons reprendre l'exemple donné au § I.2 g), il permettra au lecteur de mieux suivre les démarches adoptées par la suite et justifiera en première approximation le choix des règles de P_A.

<u>Exemple</u>.

Soit le n.e.d $A = (\Sigma, \Gamma, \delta, M, F)$ telle que :

$$\Sigma = \{a,b\},\ \Gamma = \{A, B, C, \bar{C}, S\},\ M = \{\#_g, \#_d\},\ F = \{S\}.$$

Nous indiquons les valeurs de la fonction pour que :

$$T(A) = \{a^n b^n \ /\ n \geq o\} \cup \{a^n \ /\ n \geq o\}$$

et en même temps nous donnons les règles de G_A, grammaire issue de A, déduite des valeurs de δ.

	Dans A		type	Dans G_A	
1	$\delta(\#_g, a)$	$= (\#_g a,\ \Lambda)$	$\xrightarrow{33}$	$(\#_g, a_g)$	$\longrightarrow a$
2	$\delta(a, a)$	$= (aa,\ \Lambda)$	$\xrightarrow{23}$	(a_g, a_g)	$\longrightarrow a$
3	$\delta(a, b)$	$= (A, \Lambda)$	$\xrightarrow{21}$	$(\ \ , A_g)$	$\longrightarrow (\ \ , a_g)b$
4	$\delta(A, b)$	$= (\Lambda, B)$	$\xrightarrow{22}$	$(\ \ , B_d)$	$\longrightarrow (\ \ , A_g)b$
5	$\delta(a, B)$	$= (A, \Lambda)$	$\xrightarrow{11}$	$(\ \ , A_g)$	$\longrightarrow (\ \ , a_g)(a_g, B_d)$
6	$\delta(A, \#_d)$	$= (\Lambda, \bar{C})$	$\xrightarrow{5}$	$(\ \ , \bar{C}_d)$	$\longrightarrow (\ \ , A_g)$
7	$\delta(a, \#_d)$	$= (\Lambda, C)$	$\xrightarrow{5}$	$(\ \ , C_d)$	$\longrightarrow (\ \ , a_g)$
8	$\delta(a, C)$	$= (\Lambda, C)$	$\xrightarrow{12}$	$(\ \ , C_d)$	$\longrightarrow (\ \ , a_g)(a_g, C_d)$
9	$\delta(\#_g, \bar{C})$	$= (\Lambda, S)$	$\xrightarrow{42}$	(Λ, S_d)	$\longrightarrow (\#_g, \bar{C}_d)$
10	$\delta(\#_g, C)$	$= (\Lambda, S)$	$\xrightarrow{42}$	(Λ, S_d)	$\longrightarrow (\#_g, C_d)$
11	$\delta(\#_g, \#_d)$	$= (\Lambda, S)$	$\xrightarrow{6}$	(Λ, S_d)	$\longrightarrow \Lambda$

12 $[\Lambda, S]$ étant la configuration finale $\sigma \to (\Lambda, Sd)$

Les endroits laissés sans symboles sont ceux qui correspondent aux $\forall Y$ apparaissant dans les règles de construction de P_A à savoir pour les types 11 12 21 22 et 51.

Pour ne donner que les règles utiles de G_A, nous allons partir de (Λ, S_d), qui valide grâce aux règles 9 et 10 les éléments $(\#_g, \bar{C}_d)$ et $(\#_g, C_d)$, ces éléments sont bien entendus validés en partie gauche. Dececi il vient :

$6 \rightarrow 61 \quad (\#_g, \bar{C}_d) \rightarrow (\#_g, A_g)$

$7 \rightarrow 71 \quad (\#_g, C_d) \rightarrow (\#_g, a_g)$

$8 \rightarrow 81 \quad (\#_g, C_d) \rightarrow (\#_g, a_g)\ (a_g, C_d)$

Ces nouvelles règles valident en partie gauche les éléments

$$(\#_g, A_g),\ (\#_g, a_g) \text{ et } (a_g, C_d)$$

il vient :

$3 \rightarrow 31 \quad (\#_g, A_g) \rightarrow (\#_g, a_g)b$

$5 \rightarrow 51 \quad (\#_g, A_g) \rightarrow (\#_g, a_g)\ (a_g, B_d)$

$7 \rightarrow 72 \quad (a_g, C_d) \rightarrow (a_g, a_g)$

$8 \rightarrow 82 \quad (a_g, C_d) \rightarrow (a_g, a_g)\ (a_g, C_d)$

Ces règles valident les éléments nouveaux :

$$(a_g, B_d),\ (a_g, a_g)$$

il vient :

$4 \rightarrow 41 \quad (a_g, B_d) \rightarrow (a_g, A_g)b$

Cette règle valide

$$(a_g, A_g)$$

il vient :

$3 \rightarrow 32 \quad (a_g, A_g) \rightarrow (a_g, a_g)b$

$5 \rightarrow 52 \quad (a_g, A_g) \rightarrow (a_g, a_g)\ (a_g, B_d)$

Examinons les différentes possibilités de dérivation dans G_A :

Notons que conformément à la construction de G_A, l'axiome de G_A est (Λ, S_d) donc :

⌶ $(\Lambda,S_d) \underset{11}{\Longrightarrow} \Lambda$ 11 indique la règle qui a permis cette dérivation

⌶ $(\Lambda,S_d) \Longrightarrow (\#_g,C_d) \underset{72}{\Longrightarrow} (\#_g,a_g) \underset{1}{\Longrightarrow} a$

⌶ $(\Lambda,S_d) \underset{10}{\Longrightarrow} (\#_g,C_d) \underset{81}{\Longrightarrow} (\#_g,a_g)\ (a_g,C_d)$

$\overset{n}{\underset{82}{\Longrightarrow}} (\#_g,a_g)(a_g,a_g)^{n+1}(a_g,C_d) \underset{72}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)^{n+1}(a_g,a_g)$

$\overset{n}{\underset{2}{\Longrightarrow}} (\#_g,a_g)\ a^{n+2} \underset{1}{\Longrightarrow} a^{n+3}$

⌶ $(\Lambda,S_d) \underset{9}{\Longrightarrow} (\#_g,\bar{C}_d) \underset{61}{\Longrightarrow} (\#_g,A_g) \underset{51}{\Longrightarrow} (\#_g,a_g)\ (a_g,B_d)$

$\underset{41}{\Longrightarrow} (\#_g,a_g)(a_g,A_g)b \underset{52}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)(a_g,B_d)b$

$\underset{41+52}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)^{n+1}(a_g,B_d)b^{n+1}$

$\underset{41}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)^{n+1}(a_g,A_g)bb^{n+1}$

$\underset{32}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)^{n+1}(a_g,a_g)bbb^{n+1}$

$\overset{n+2}{\underset{2}{\Longrightarrow}} (\#_g,a_g)\ a^{n+2}\ b^{n+3}$

$\underset{1}{\Longrightarrow} a^{n+3}\ b^{n+3}$

⌶ $(\Lambda,S_d) \underset{9}{\Longrightarrow} (\#_g,\bar{C}_d) \underset{61}{\Longrightarrow} (\#_g,A_g) \underset{31}{\Longrightarrow} (\#_g,a_g)\ b \underset{1}{\Longrightarrow} ab$

Notons que toutes les règles de G_A ont été utilisées.

Nous donnons maintenant en parallèle les mouvements dans A, et les mouvements d'analyse "top-down" correspondant dans G_A.

Nous choisissons comme mot de Σ^* : a^3b^3

Dans A	Dans G_A
$[\#_g, aaabbb\#_d]$	$aaabbb$
$^{1}\top$	$^{1}\Uparrow$
$[\#_g a, aabbb\#_d]$	$(\#_g, a_g)aabbb$
$^{2}\top^{*}$	$^{2}\Uparrow^{*}$
$[\#_g aaa, bbb\#_d]$	$(\#_g, a_g)(a_g, a_g)(a_g, a_g)bbb$
$^{3}\top$	$^{32}\Uparrow$
$[\#_g aaA, bb\#_d]$	$(\#_g, a_g)(a_g, a_g)(a_g, A_g)bb$
$^{4}\top$	$^{41}\Uparrow$
$[\#_g, aa, Bb\#_d]$	$(\#_g, a_g)(a_g, a_g)(a_g, B_d)b$
$^{5}\top$	$^{52}\Uparrow$
$[\#_g aA, b\#_d]$	$(\#_g, a_g)(a_g A_g)b$
$^{4}\top$	$^{41}\Uparrow$
$[\#_g a, B\#_d]$	$(\#_g, a_g)(a_g, B_d)$
$^{5}\top$	$^{51}\Uparrow$
$[\#_g A, \#_d]$	$(\#_g, A_g)$
$^{6}\top$	$^{61}\Uparrow$
$[\#_g, \bar{C}]$	$(\#_g, \bar{C}_d)$
$^{9}\top$	$^{9}\Uparrow$
$[\Lambda, S]$	(Λ, S_d)

c) *Définition.*

Nous dirons qu'un mot α de V_G^* est une expression bien formée si il est d'une des formes suivantes :

$$\forall i \; X^i \in V_G,\ a_i \in \Sigma_G,\ X \in V_G,\ z \in V_G,\ n, p \geq 0$$

1 $\alpha = (\#^o_g, X^1_g)(X^1_g, X^2_g) \ldots (X^{i-1}_g, X^i_g)(X^i_g, X^{i+1}_g) \ldots (X^n_g, X_g)\ a_1 a_2 \ldots a_p$ (1)

2 $\alpha = (\#^o_g, X^1_g)(X^1_g, X^1_g) \ldots (X^{i-1}_g, X^i_g)(X^i_g, X^{i+1}_g) \ldots (X^n_g, Z_d)\ a_1 a_2 \ldots a_p$ (2)

Si le marqueur $\#_d$ est utilisé :

$$Z \in T_r(\#_d) \Longrightarrow a_1 \ldots a_p = \Lambda$$

3 $\alpha \in \Sigma_G$

Nous pouvons énoncer le lemme suivant :

<u>Lemme</u>.

Pour toute suite $(\alpha_i)_{i=1}^{n}$ telle que :

$$\xRightarrow{d} \alpha_1 \xRightarrow{d} \alpha_2 \Longrightarrow \ldots \xRightarrow{d} \alpha_n,$$

dérivation dans G_A, alors pour tout i, α_i est une expression bien formée.

<u>Démonstration</u>.

Montrons ce lemme par récurrence sur I. Nous verons que ce lemme est évident pour les familles n'utilisant pas $\#_d$ car la condition $Z \subset T_r(\#_d) \Longrightarrow a_1 \ldots a_n = \Lambda$ ne tient plus car $\#_d$ n'est pas employé pour ces familles.
Il est une conséquence de la forme des productions de G.

o Montrons que le lemme est vrai pour i = 1. On peut avoir

(1) Un cas particulier, pour n = O sera : $\alpha = (\#^o_g, X_g)a_1 \ldots a_p$ et en plus si p = O, $\alpha = (\#_g, X_g)$ ou $\alpha = (\Lambda, X_g)$.

(2) Un cas particulier pour n = O sera : $\alpha = (\#^o_g, Z_d)a_1 \ldots a_p$ ici aussi si p = O $\alpha = (\#_g, Z_d)$ ou $= (\Lambda, Z_d)$

soit $\sigma \to \alpha_1 = (\Lambda, X_d)$,

soit $\sigma \to \alpha_1 = (\Lambda, X_g)$, (point 6 de la construction de δ),

soit $\sigma \to \Lambda$ et α_1 est dans tout ces cas bien formée.

Notons que si $\#_d$ est utilisé par 2.1 on a, $X \in F$, $\alpha_1 = (\Lambda, X_d)$, $X \in T_r(\#_d)$ et l'on a bien $a_1 \ldots a_p = \Lambda$

o Supposons le lemme vrai pour i=p et soit :

$$\sigma \xRightarrow{d} \alpha_1 \xRightarrow{d} \ldots \xRightarrow{d} \alpha_p \xRightarrow{d} \alpha_{p+1}$$

nous avons donc α_p est bien formé, montrons que α_{p+1} est bien formé.

Deux cas pour α_p :

$\Pi \quad \alpha_p = (\Lambda_g, X_g^1) \ldots (X_g^n, X_g)\ a_1 \ldots a_n$

la dérivation $\alpha_p \xRightarrow{d} \alpha_{p+1}$ étant la plus à droite, α_{p+1} est obtenu à partir de α_p par dérivation de (X_g^n, X_g). Plusieurs possibilités de dériver (X_g^n, X_g)

. Dérivation par $(X_g^n, X_g) \to (X_g^n, A_g)(A_g, B_d)$ issue de $(X, \Lambda) \in \delta(A,B)$.

Pour que α_{p+1} soit bien formé il suffit que

$$B \in T_r(\#_d) \Longrightarrow a_1 \ldots a_p = \Lambda.$$

Supposons que $B \in T_r(\#_d)$, ceci est impossible par P22 car $(X, \Lambda) \in \delta(A,B)$ et $B \in T_r(\#_d)$. Donc nécessairement

$$\alpha_{p+1} = (\Lambda_g, X_g') \ldots (X_g^n, A_g)(A_g, B_d) a_1 \ldots a_n$$

est bien formée, α_p étant bien formée.

. Dérivation par $(X_g^n, X_g) \to (X_g^n, X_d')$ issue de $(X^nX, \Lambda) \in \delta(X^n, X')$, ici encore $X' \notin T_r(\#_d)$ par P22.
Il en va exactement de la même manière pour les productions issues de 41 et 43 de la construction.

. Il est immédiat par ailleurs que l'application de productions issues de 21, 23, 31 ou 33 donne un α_{p+1} bien formé.

II $\alpha_p = (\Lambda_g, X_g^1) \dots (X_g^n, Z_d)\ a_1 \dots a_p$ avec si $Z \in T_r(\#_d)$ alors $a_1 \dots a_p = \Lambda$

. Dérivation par $(X_g^n, Z_d) \to (X_g^n, X_g')(X_g', Y_d)$ issue de $(\Lambda, Z) \in \delta(X', Y)$ ce qui donne

$$\alpha_{p+1} = (\Lambda_g, X_g^1) \dots (X_g^n, X_g')(X_g', Y_d)\ a_1 \dots a_p$$

de plus $Y \in T_r(\#_d)$ implique $Z \in T_r(\#_d)$ car $(\Lambda, Z) \in \delta(X, Y)$ donc dans ce cas $a_1 \dots a_p = \Lambda$, α_{p+1} est bien formée.

. La dérivation par des règles correspondantes à 22, ou 32 ne pose pas de problèmes.

. Application de 42 à savoir $(\Lambda, Z_d) \to (\#_g, A_d)$ issue de $(\Lambda, Z) \in \delta(\#_g, A)$, n'appelle aucune remarques particulières. Si $Z \in T_r(\#_d)$ alors $A \in T_r(\#_d)$ à cause de la propriété <u>tdm</u> dans ce cas $a_1 \dots a_p = \Lambda$, il vient :

$$\alpha_p = (\Lambda, Z_d) \quad \to \alpha_{p+1} = (\#_g, A_d)$$

α_{p+1} est bien formée. Notons simplement, que ceci n'a un sens que si $Z \in F$ alors $\sigma \to (\Lambda, Z_d)$.

. Dérivation par 5 $(X_g^n, Z_d) \to (Y_g, A_g)$ ce cas ne pose aucun problème particulier, cette production est issue de $(\Lambda, Z) \in \delta(A, \#_d)$ il vient :

$$Z \in T_r(\#_d)$$

donc $a_1 \dots a_n = \Lambda$ et :

$$\alpha_p = (\Lambda_g, X'_g) \ \dots \ (X^n_g, Z_d) \Longrightarrow \alpha_{p+1} = (\Lambda_g, X'_d) \ \dots \ (Y_g, A_g)$$

α_{p+1} est bien formée.

d) Nous définissons à présent une correspondance entre les expressions bien formées et les configurations du n.e.d A.

Définition.

Nous dirons qu'une expression bien formée α et une configuration K de A sont correspondantes nous noterons $\alpha \longleftrightarrow K$. Si nous nous trouvons dans un des cas suivant :

$\forall X^i \in \Sigma \cup \Gamma$, $\forall a_i \in \Sigma$, $Y \in \Gamma$, $X \in \Sigma \cup \Gamma$, $n, p \geq 0$

1 $\alpha = (\#^o_g, X^1_g) \ \dots \ (X^{i-1}_g, X^i_g)(X^i_g, X^{i+1}_g) \ \dots \ (X^n_g, X_g) a_1 \ \dots \ a_p$

$K = [\#^o_g X^1 \ \dots \ X^{i-1} X^i X^{i+1} \ \dots \ X^n X, \ a_1 \ \dots \ a_p \ \#^o_d]$

2 $\alpha = (\#^o_g, X^1_g) \ \dots \ (X^{i-1}_g, X^i_g)(X^i_g, X^{i+1}_g) \ \dots \ (X^n_g, Y_d) \ a_1 \ \dots \ a_p$

. si $Y \notin T_r(\#_d)$

$K = [\#^o_g X^1 X^2 \ \dots \ X^{i-1} X^i X^{i+1} \ \dots \ X^n, \ Y a_1 \ \dots \ a_p \ \#^o_d]$

. si $Y \in T_r(\#_d) / \{\#_d\}$ alors $a_1 \ \dots \ a_p = \Lambda$(car α est bien formée).

$K = [\#^o_g X' \ \dots \ X^i \ \dots \ X^n, Y]$

3 Si $\alpha \in \Sigma^*_G$ alors : $K = [\#_g, \alpha \#^o_d]$

Lemme.

Pour tout mot $w \in \Sigma^*$, il y a équivalence entre :

1 $\exists\ (K_i)_{i=o}^{n}$ telle que $K_o = [\#_g, w\#d^o]$ et $K_o \vdash K_1 \vdash \dots \vdash K_n$

2 $\exists\ (\alpha_i)_{i=o}^{n}$ suite d'exp. bien formées telle que :

$$\alpha_o = w \text{ et } \Gamma \Longrightarrow \alpha_n \overset{d}{\Longrightarrow} \alpha_{n+1} \overset{d}{\Longrightarrow} \alpha_1 \overset{d}{\Longrightarrow} \alpha_o$$

De plus :

Point 1. o K_i est finale ssi α_i est l'axiome de G

Point 2. o $\forall i,\ o \leq i \leq n \quad K_i \longleftrightarrow \alpha_i$

Preuve.

Dans cette démonstration nous allons utiliser les tableaux a) b) c) d) et e).

Tout les cas possibles de configuration K de A et d'expressions bien formées α, telles que $\alpha \longleftrightarrow K$ sont présentés divisés en 5 cas, s'excluant deux à deux.

Dans chaque tableau nous avons envisagé toutes les possibilités de mouvements à partir de K dans A, et toutes les possibilités de réductions à partir de α dans G donnant α' bien formée. Nous notons K' et α' la configuration et l'expression bien formée telle que :

$$K \vdash K' \text{ et } \alpha' \overset{d}{\Longrightarrow} \alpha,$$

nous avons fait figurer K' dans la deuxième colonne et α' dans la quatrième.

Les règles de A et la production de G ayant permis ces mouvements et réductions figurent sur une même ligne dans la première et la troisième colonne. Nous avons la propriété suivante :

	$K = [\#_g, a_1 \dots a_p \#^o_d]$		$\alpha = a_1 \dots a_p$	
	Règle de A $\quad K \longmapsto K'$	K'	Production de P $\quad \alpha \overset{d}{\Longrightarrow} \alpha'$	α'
a1	$(X,\Lambda) \in \delta(\#_g, a_1)$	$[X, a_2 \dots a_p \#^o_d]$	$(\Lambda, X_g) \to a_1$	$(\Lambda, X_g) a_2 \dots a_p$
a2	$(\Lambda, X) \in \delta(\#_g, a_1)$	$[\underline{\Lambda}, X a_2 \dots a_p \#^o_d]$	$(\Lambda, X_d) \to a_1$	$(\Lambda, X_d) a_2 \dots a_p$
a3	$(\#_g A, \Lambda) \in \delta(\#_g, a_1)$	$[\#_g A, a_2 \dots a_p \#^o_d]$	$(\#_g, A_g) \to a_1$	$(\#_g, A_g) a_2 \dots a_p$

Tableau a) $1 \leq p$.

	$K = [\#^o_g X^1 \dots X^n X, a_i \dots a_p \#^o_d]$		$\alpha = (\#^o_g, X^1_g)(X^1_g, X^2_g) \dots (X^n_g, X_g) a_i \dots a$	
	Règle de A $\quad K \longmapsto K'$	K'	Production de P $\quad \alpha' \overset{d}{\Longrightarrow} \alpha$	α'
b1	$(X', \Lambda) \in \delta(X, a_i)$	$[\#^o_g X^1 \dots X^n X', a_{i+1} \dots a_p \#^o_d]$	$(X^n_g, X'_g) \to (X^n_g, X_g) a_i$	$(\#^o_g, X^1_g) \dots (X^n_g, X'_g) a_{i+1} \dots a_p$
b2	$(\Lambda, X') \in \delta(X, a_i)$	$[\#^o_g X^1 \dots X^n, X' a_{i+1} \dots a_p \#^o_d]$	$(X^n_g, X'_d) \to (X^n_g, X_g) a_i$	$(\#^o_g, X'_g) \dots (X^n_g, X'_d) a_{i+1} \dots a_p$
b3	$(XX', \Lambda) \in \delta(X, a_i)$	$[\#^o_g X^1 \dots X^n XX', a_{i+1} \dots a_p \#^o_d]$	$(X_g, X'_g) \to a_i$	$(\#^o_g, X^1_g) \dots (X_g, X_g)(X_g, X'_g) a_{i+1} \dots a_p$

Tableau b) $1 \leq i \leq p$ (1), $0 \leq n$ si $n = 0$ $X^o = \#^o_g$

(1) C'est-à-dire $a_i \dots a_p \neq \Lambda$

	$K = [\#_g, Ya_1 \dots a_n \#^o_d]$		$\alpha = (\#_g, Y_d) a_1 \dots a_n$	
	Règle de A $\quad K \longmapsto K'$	K'	Production de P $\quad \alpha' \xrightarrow{d} \alpha$	α'
d1	$(X', \Lambda) \in \delta(\#_g, Y)$	$[X', a_1 \dots a_n \#^o_d]$	$(\Lambda, X'_g) \to (\#_g, Y_d)$	$(\Lambda, X'_g) a_1 \dots a_n$
d2	$(\Lambda, X') \in \delta(\#_g, Y)$	$[\Lambda, X' a_1 \dots a_n \#^o_d]$	$(\Lambda, X'_d) \to (\#_g, Y_d)$	$(\Lambda, X'_d) a_1 \dots a_n$
d3	$(\#_g X', \Lambda) \in \delta(\#_g, Y)$	$[\#_g X', a_1 \dots a_n \#^o_d]$	$(\#_g, X'_g) \to (\#_g, Y_d)$	$(\#_g, X'_g) a_1 \dots a_n$

Tableau d).

	$K = [\#^o_g X^1 \dots X^n, Ya_i \dots a_p \#^o_d]$		$\alpha = (^o_g, X^1_g) \dots (X^{n-1}_g, X^n_g)(X^n_g, Y_d) a_i \dots a_p$	
	Règle de A $\quad K \longmapsto K'$	K'	Production de P $\quad \alpha' \xrightarrow{d} \alpha$	α'
c1	$(X', \Lambda) \in \delta(X^n, Y)$	$[\#^o_g X^1 \dots X^{n-1}, a_i \dots a_p \#^o_d]$	$(X^{n-1}_g, X'_g) \to (X^{n-1}_g, X^n_g)(X^n_g, Y_d)$	$(\#^o_g, X^1_g) \dots (X^{n-1}_g, X'_g) a_i \dots a_p$
c2	$(\Lambda, X') \in \delta(X^n, Y)$	$[\#^o_g X^1 \dots X^{n-1}, X' a_i \dots a_p \#^o_d]$	$(X^{n-1}_g, X'_d) \to (X^{n-1}_g, X^n_g)(X^n_g, Y_d)$	$(\#^o_g, X^1_g) \dots (X^{n-1}_g, X'_d) a_i \dots a_p$
c3	$(X^n X', \Lambda) \in \delta(X^n, Y)$	$[\#^o_g X^1 \dots X^n X', a_i \dots a_p \#^o_d]$	$(X^n_g, X'_g) \to (X^n_g, Y_d)$	$(\#^o_g, X^1_g) \dots (X^n_g, X'_g) \; a_i \dots a_p$

Tableau c) $\#^o_g X^1 \dots X^n \neq \#_g$

	$K = [\#^o_g X^1 \dots X^n X, \#_d]$		$\alpha = (\#^o_g, X^1_g) \dots (X^n_g, X_g)$	
	Règle de A $\quad K \longmapsto K'$	K'	Production de P $\quad \alpha' \xrightarrow{d} \alpha$	α'
e1	$(\Lambda, X') \in \delta(X, \#_d)$	$[\#^o_g X^1 \dots X^n, X']$	$(X^n_g, X'_d) \to (X^n_g, X_g)$	$(\#^o_g, X^1_g) \dots (X^n_g, X'_d)$

Tableau e).

Dans chaque tableau on a :

Pour tout $i \geq 1$, $a_i \in \Sigma$, $X^i \in \Sigma \cup \Gamma$ et $X, X' Y \in \Sigma \cup \Gamma$

Sur une même ligne K' <—> α' et la règle de A existe si et seulement si la production de P existe.

Cette propriété est évidente dans presque tout les cas. Disons que la restriction imposée par P1.1 dans la construction, nous donne dans la ligne e1 $X' \neq \#_d$, donc K' <—> α'. Que de même P1.2 assure à la ligne d1 $X' \neq \#_g$, donc K' <—> α'.

Il est immédiat qu'à toute suite $(K_i)_{i=o}^{n}$, je sais associer grâce à la correspondance une suite d'expression bien fournies $(\alpha_i)_{i=o}^{n}$ et réciproquement. De plus la correspondance étant biunivoque, la suite des α_i est unique et réciproquement.

Nous allons montrer par récurrence sur i que :

$$(K_o \vdash K_1 \vdash \dots \vdash K_i) \Longleftrightarrow (\alpha_i \Longrightarrow \alpha_{i-1} \Longrightarrow \dots \Longrightarrow \alpha_1 \Longrightarrow \alpha_o)$$

les α_i et les K_i étant en correspondance

L'équivalence est évidente pour i = 0.

Le sous-tableau a) représentant tous les cas à envisagés pour i=1, nous montrerons à part le cas i=1.

Puis pour i⩾2 nous emploierons les sous-tableaux b) c) d) e), en effet si

$$K_o \overset{i}{\vdash} K_n$$

alors :

$$i \geqslant 2 \Longrightarrow K_i \neq [\#_g, a_1 \dots a_n \#_d^o]$$

ceci est dû à la propriété P1.2, donc le sous-tableau a) n'est plus à considérer pour i ⩾ 2.

Ⅱ Montrons que l'équivalence est vrai pour i=1. $(K_o \vdash K_1) \Longleftrightarrow (\alpha_1 \Longrightarrow \alpha_o)$

Nous avons regroupé dans le sous-tableau a) tous les cas possibles de mouvements à partir de K_o et toutes les réductions possibles à partir de α_o

donnant un α_1 bien formé. En effet :

c'est tout à fait évident pour les mouvements à partir de K_o, ces trois mouvements a1 a2 a3 sont les seuls possibles dans A.

Pour les réductions à partir de α_o, nous aurions pu appliquer à $\alpha_o = a_1 \dots a_n$ une réduction par une production $(A_g, B'_g) \to a_1$ avec $A_g \neq \#^o_g$ issue nécessairement de $(A,B') \in \delta(A,a)$, qui donnerait

$$\alpha_1 = (A_g, B'_g)\, a_2 \dots a_n$$

et aucune configuration K de A serait telle que

$$K_o \vdash K \text{ et } K <\!\!-\!\!> \alpha_1$$

car

$$\alpha_1 <\!\!-\!\!> K = [A_g B'_g, a_2 \dots a_n \#^o_d] \text{ et } K_o \vdash K$$

est faux.

Mais α_1 n'est pas dans ce cas une expression bien formée car $A_g \neq \#^o_g$, ce qui est en contradiction avec l'hypothèse du lemme. Toutes les autres réductions sont envisagées dans a), car les seules autres possibilités sont l'emploie de 31, 32 ou 33 de la définition.

Nous avons vu que dans une même ligne la règle de A implique l'existence de la production de P et réciproquement ceci nous permet d'affirmer l'équivalence pour i=1 et le point 2 en constatant la correspondance de K' et α' sur une même ligne.

Reste le point 1. K_1 ne peut être finale que dans le cas des feuilles Δ_1 ou Δ_2, pour lesquelles on peut avoir

$$K_o = [\#_g, a_1] \text{ et } \alpha_o = a_1$$

il est évident que par a1) pour Δ_1 et a2) pour Δ_2 nous atteignons la configuration finale dans A et l'axiome dans G et que par 3) de la construction Δ_1 sera l'axiome si et seulement si Δ_1 est finale.

De plus si $(\sigma \to \Lambda) \in P$ il vient par 6 que $(\Lambda, S) \in \delta(\#_g, \#_d)$ donc :

$$(\sigma \to \Lambda) \Longleftrightarrow [\#_g, \#_d] \vdash [\Lambda, S]$$

II Supposons le lemme vrai pour i=p, montrons qu'alors il est vrai pour i=p+1.

Soit $K_o \vdash \ldots \vdash K_p \vdash K_{p+1}$ et $\alpha_{p+1} \overset{d}{\Longrightarrow} \alpha_p \overset{d}{\Longrightarrow} \ldots \overset{d}{\Longrightarrow} \alpha$.

Le lemme étant supposé vrai pour n=p nous avons :

$$\exists\, K_o \vdash \ldots \vdash K_p \text{ ssi } \exists\, \alpha_p \overset{d}{\Longrightarrow} \ldots \overset{d}{\Longrightarrow} \alpha_o$$

et

$$K_p \longleftrightarrow \alpha_p .$$

Il nous faut donc considérer tout les cas $K_n \vdash K_{n+1}$ et $\alpha_{n+1} \overset{d}{\Longrightarrow} \alpha_n$ avec α_n expression bien formée et $K_n \longleftrightarrow \alpha_n$. C'est ce qui est fait dans les sous-tableaux b) c) d) et e), ou nous précisons à chaque fois la règle qui a permis le mouvement dans A et la production qui a permis la réduction dans G_A. Tout les cas ont été envisagés en effet :

. Pour les configurations de A, c'est tout à fait évident (les divers cas correspondant en fait aux catégories de règles dans la définition de G_A).

. Pour les expressions bien formées de G nous avons en plus des cas envisagés dans le tableau, les cas suivants donnant lieu à un α_{p+1} qui n'est pas une expression bien formée.

o Dans le tableau b) on pourrait appliquer à (X_{n_g}, X_g) une réduction par la production $(X_{n_g}, X'_d) \to (X_{n_g}, X_g)$ issue de $(\Lambda, X') \in^g \delta(X, \#_d)$ notons que ceci implique $X' \in T_r(\#_d)$, ce qui donnerait $\alpha_{p+1} = (\Lambda, X_{1_g}) \ldots (X_{n_g}, X'_d) a_p \ldots a_{p'}$ qui n'est bien formée que si $a_p \ldots a_{p'} = \Lambda$ (cas traité dans le sous-tableau e)) ce qui est faux ici.

Dans ce cas nous avons :

$$\alpha_{p+1} \longleftrightarrow [\Lambda_g X_1 \ldots X_n, X' a_p \ldots a_p, \#_d^o] = K_{p+1}$$

il est immédiat qu'aucune règle ne peut assurer le mouvement :

$$K_p = [\Lambda_g X_1 \ldots X_n X, a_p \ldots a_p, \#_d^o] \vdash K_{p+1} = [\Lambda_g X_1 \ldots X_n, X' a_p \ldots a_p, \#_d^o]$$

ceci montre comme précédemment que

$$\alpha_{p+1} \Longrightarrow \alpha_p$$

ne correspond à un mouvement $K_p \vdash K_{p+1}$ que si α_{p+1} et α_p sont bien formées.

Toujours dans le sous-tableau b) on pourrait appliquer à a_p une réduction par la production $(A_g, X'_g) \rightarrow a_p$ issue de $(AX', \Lambda) \in \delta(A, a_p)$ avec $A \neq X$ ce qui donnerait $\alpha_{p+1} = (\Lambda_g, X_{1_g}) \ldots (X_{n-1_g})(A_g, X'_g) a_{p+1} \ldots a_p$, qui n'est pas bien formée car $X \neq A$.

Dans ce cas nous avons :

$$\alpha_{p+1} \longleftrightarrow [\Lambda_g X_1 \ldots X_n AX', a_{p+1} \ldots a_p, \#_d^o] = K_{p+1}$$

ici encore :

$$K_p = [\Lambda_g X_1 \ldots X_n X, a_p \ldots a_p, \#_d^o] \vdash K_{p+1}$$

est impossible car il faudrait que :

$$(AX', \Lambda) \in \delta(X, a_p)$$

ce qui n'est pas possible d'après P4 car $A \neq X$.

Ces deux cas, plus les trois envisagées dans le sous-tableau sont les seuls envisageables pour α de cette forme.

o Dans le tableau c) il est immédiat que toutes réductions appliquées à a_p ne peut donner α_{p+1} bien formée, de même pour le tableau d).

Tout les cas possibles ont donc été envisagées. La propriété, plus le constat de la correspondance sur chaque lignes du tableau de K' et α', démontre 1 $\Longleftrightarrow$ 2 pour n=p+1 et le point 2.

Le point 1 est aisément vérifié. K_{p+1} sera finale ou α_{p+1} sera l'axiome de G dans les cas suivants :

ligne b1) pour $K = K_p = [X,a_p]$ pour la famille n.e.d.d. 1

ligne b2) pour $K = K_p = [X,a_p]$ pour la famille n;e.d.d. 2

ligne c1) pour $K = K_p = [X_n,Y]$ pour la famille n.e.d.d. 1

ligne c2) pour $K = K_p = [X_n,Y]$ ou $[X_n,\#_d]$ pour n.e.d.d. ou n.e.d.d. 2

ligne d1) pour $K = K_p = [\#_g,Y]$ pour la famille n.e.d.d. 1

ligne d2) pour $K = K_p = [\#_g,Y]$ pour la famille n.e.d.d. ou n.e.d.d. 0

ligne e) pour $K = K_p = [X,\#_d]$ pour la famille n.e.d.d.

<u>cqfd</u>.

e) Nous pouvons donc énoncer le corollaire qui est une correspondance direct du des lemmes précédents.

<u>Corollaire</u>.

$$L(G) = T(A)$$

<u>Preuve</u>.

⌶ $L(G) \subseteq T(A)$

Soit $w \in L(G)$, alors on sait qu'il existe une dérivation la plus à droite donnant w soit :

$$\sigma \overset{d}{\Longrightarrow} \alpha_1 \Longrightarrow \alpha_2 \Longrightarrow \dots \overset{d}{\Longrightarrow} \alpha_n = w$$

d'après le lemme c) les α_i sont bien formées, on peut donc leur faire correspondre dans A, une suite de configurations $(K_i)_{i=1}^{n}$ telle que :

$$K_n \quad K_n <-> w$$

$$K_n = [\#_g, w\#_d^o] \vdash K_{n-1} \vdash \dots \vdash K_1$$

K_1 étant une configuration finale car $\sigma \to \alpha_1$ et $\alpha_1 <-> K_1$ donc $w \in T(A)$

II $T(A) \subseteq L(G)$

ce résultat est une conséquence directe du lemme du §.d.

cqfd.

f) Nous pouvons à présent conclure cette partie en énonçant le théorème.

Théorème.

La construction donnée au §-a) permet d'associer à tout n.e.d, une grammaire canonique algébrique équivalente.

Preuve.

Il suffit de montrer qu'à tout n.e.d, on peut associer un n.e.d équivalent vérifiant les propriétés imposées dans la construction. C'est ce qui est fait pour le plus difficile dans la partie I.3 du premier chapitre. Pour le reste nous laissons au lecteur le soin de le constater.

cqfd.

CHAPITRE IV

APPLICATION AUX GRAMMAIRES

B R C (L,K)

IV.1 PRESENTATION DES GRAMMAIRES BRC(1,k).

a)

Il s'agit de définir les grammaires, pour lesquelles il est possible pour tout mot du langage qu'elles engendrent, d'effectuer une analyse déterministe, c'est-à-dire unique, des feuilles vers la racine. Cette analyse s'effectuera de la gauche vers la droite du mot, elle correspondra donc dans G à la dérivation la plus à droite.

Pour lever éventuellement une ambiguité à un moment donné de l'analyse, on pourra considérer le contexte du mot qui fera l'objet de la réduction. Plus précisément :

Définition. Dérivation la plus à droite.

Soit une grammaire $G = (\Sigma, N, P, \sigma)$ (1). *Pour* $\alpha \in V^*$, $A \in N$, $\beta \in V^*$ et $(A \to w) \in P$: *nous dirons que* $\alpha A\beta$ *engendre* $\alpha w\beta$ *dans une dérivation la plus à droite et nous noterons* $\alpha A\beta \overset{d}{\underset{G}{\Longrightarrow}} \alpha w\beta$ *ssi* $\beta \in \Sigma^*$.

Définition "poignet".

Soit $\alpha \in V^*$. Toute paire (p,i) dans laquelle $p \in P$ et $i \in N$ est appellée poignet de α ssi :

pour $A \in V/\Sigma$, α', $\beta \in V^*$ et $w \in \Sigma^*$ on a

1 $S \overset{*d}{\Longrightarrow} \alpha' A w \overset{d}{\Longrightarrow} \alpha'\beta w = \alpha$

2 $p : A \to \beta$

3 $i = |\alpha'\beta|$.

Commentaires.

1 Dans une analyse "top down", il s'agit partant de $\alpha'\beta w$ de retrouver $\alpha' A w$. Il est immédiat qu'il faudra nécessairement que $w \in \Sigma^*$

(1) Nos notations sont celle utilisées dans [8]. N est l'ensemble des non-terminaux de G et nous notons : $V = N \cup \Sigma$, Σ étant l'ensemble des terminaux.

sinon il existerait $A' \in V / \Sigma$ tel que :

$$\alpha'AA'w \overset{d}{\Longrightarrow} \alpha'\beta A'w$$

ce qui est impossible pour une dérivation la plus à droite

2 La lettre A dans $\alpha'Aw$ est le non terminal le plus à droite, notons que $\forall\alpha$ le non terminal le plus à droite est nécessairement celui qui provient de la réduction précédente dans une analyse.

Rappellons la définition, donnée par Havel-Harrison dans [12], des grammaires B R C (l,k).

Définition.

Soit G une grammaire algébrique, $G = (N, \Sigma, P, \sigma)$, sans dérivations de la forme

$$S \overset{d+}{\Longrightarrow} S.$$

Soit $l, k \geq o$. Alors G est une grammaire B R C (l,k), ssi les conditions suivantes sont réalisées, pour tout $\alpha, \alpha', \beta \in V^*$, $w, w' \in \Sigma^*$, $A \in N$, $p' \subset P$ et $i' \geq lg(\alpha'\beta)$.(1)

Si $(A\to\beta, lg(\alpha\beta))$ et (p',i') sont des poignets de $\alpha\beta w$ et $\alpha'\beta w'$ respectivement, et si $\alpha^{(l)} = \alpha'^{(l)}$ et ${}^{(k)}w' = {}^{(k)}w'$ alors p' est la forme $A\to\beta$ et $i' = l_g(\alpha'\beta)$. G est B R C si elle est B R C (l,k) pour un l et un k supérieurs ou égaux à zéro.

Commentaires.

. L'interdiction des dérivations $S \overset{+d}{\Longrightarrow} S$, permet d'assurer que les grammaires B R C (l,k) sont non-ambigues. En effet la grammaire dont les seules règles sont $S \to S$ et $S \to a$ est une grammaire B R C (O,O) ambigue.

. Dans une grammaire B R C (l,k), tout mot est analysable dans une analyse des feuilles vers la racine, de la gauche vers la droite. La poignet est à chaque pas unique, la détermination est faite en consultant le contexte du mot réduit, d'une profondeur l sur sa gauche, et d'une profondeur k sur sa droite.

(1) Pour tout mot w, lg(w) est la longueur de w.

Nous allons maintenant préciser ces notions. Cette étude nous amménera à poser une nouvelle définition des grammaires BRC (1,k), dont nous montrerons qu'elle est équivalente à celle que nous venons d'énoncer.

Définition "pointer".

Pour tout mot w, de la forme $\alpha_1 X \beta_1$, avec $\alpha_1 \in V^*$, $X \in V$, $\beta_1 \in \Sigma^*$ nous dirons que pointer X c'est chercher s'il existe :

$$. \text{ Si } X \neq \Lambda \ : \alpha', \alpha' \in V \ , p \in P \text{ tels que}$$

$$\alpha_1 = \alpha'\alpha, \ p : A \to \alpha X$$

$$. \text{ Si } X = \Lambda : p \in P \text{ tel que : } p : A \to \Lambda$$

Lorsque l'on pointe un élément X on peut se trouver dans un des 4 cas suivants : pour $X \neq \Lambda$.

1 . $\exists ! p \ \exists ! \alpha$ tel que $p : A \to \alpha X$

2 . $\exists p, p'$ tq : $p : A \to \alpha X, \ p' : A' \to \alpha X$

3. $\exists \alpha, \alpha', \beta, \beta', p, p'$ tq : $\alpha_1 = \beta\alpha = \beta'\alpha'$

$$p : A \to \alpha X \quad p : A' \to \alpha' X$$

4. $\nexists \alpha \ \nexists p$ tq : $p : A \to \alpha X$

dans le 4$^{\text{ième}}$ cas aucune réduction n'est possible, dans la recherche de la poignet, nous pointerons donc le symbole immédiatement à droite de X

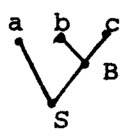
Pointer X c'est donc chercher s'il existe une poignet dont "l'extrémité" est X.

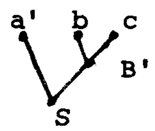
Cas $X = \Lambda$, si $p : A \to \Lambda$, à chaque instant de l'analyse cette possibilité est envisageable, elle est donc une source de non déterminisme dans une analyse "top-down".

b) Exemples.

(nous supposons dans ces exemples que la grammaire est entièrement donnée par les arbres de dérivation que nous proposons).

Ex 1.

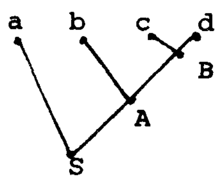


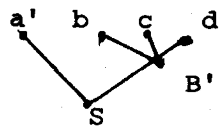


cet exemple correspond au 2e cas, lorsque nous pointons c nous avons soit B → bc soit B' → bc.

Notons que la considération du contexte gauche de bc permet de lever l'ambiguïté

Ex 2.



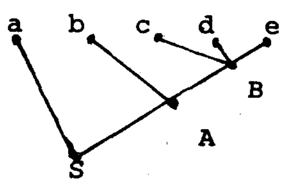


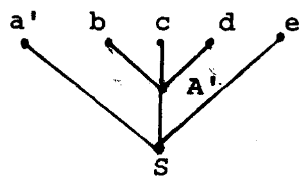
ici nous sommes dans le cas 3 lorsque l'on pointe sur d nous avons 2 poignets possibles :

B → cd   B' → bcd

ici encore le contexte gauche de bcd permet de lever l'ambiguïté

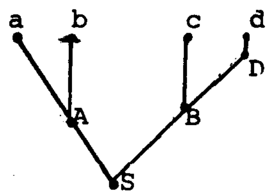
Ex 3.





dans cette exemple lorsque l'on pointe d, A' → bcd est applicable, pourtant cette réduction ne doit pas être faite si le contexte geh de bed n'est pas a'.

Ex 4.



la seule poignet dans cet exemple est (A → ab,2) elle seule correspond à la dérivation la plus à droite.

c) Ces exemples nous montrent que la détermination de la poignet permettant la réduction peut se faire dans certains cas, en considérant le contexte du mot faisant l'objet de la réduction. Nous voyons apparaître la notion de contexte dans lequel une règle est applicable, nous allons préciser cette notion en définissant une application c de P dans $P(V_\Lambda^l \times \Sigma_\Lambda^k)$ (1) l et k indiquant respectivement la profondeur du contexte gauche et celle du contexte droit qu'il est nécessaire de connaître pour lever une ambiguïté lors d'un pointage. Soit :

$$c : P \to \mathcal{P}(V_\Lambda^l \times \Sigma_\Lambda^k) \text{ (1)}.$$

Dans ces conditions lorsque l'on pointe un élément X une règle $p : A \to \alpha X$ sera c-applicable pour $w = \alpha_1 \alpha X \beta_1$ ssi

$$(\alpha_1^{(l)}, {}^{(k)}\beta_1) \in c(P) \text{ (2)}.$$

Plus précisèment :

(1) Exeptionnellement nous conviendrons des conventions suivantes :
$V_\Lambda = V \cup \{\Lambda\}$ ce qui est classique mais
$V^o = \{\Lambda\}$ ce qui est différent de la convention du 1er chapitre.

(2) Rappellons que $\alpha^{(e)}$ (resp. ${}^{(e)}\alpha$) est le suffixe (resp. préfixe) de longueur q, $q = \inf(l_g(\alpha), e)$ i.e. : $\alpha^{(o)} = \Lambda$ quelque soit α.

d) *Définition.*

c-analyse : $w' \overset{c}{\underset{G}{\Longleftarrow}} w$.

Pour toute grammaire $G = (N, \Sigma, P, \sigma)$ et toute application c

$$c : P \to \mathcal{P}(V_\Lambda^l \times \Sigma_\Lambda^k),$$

nous écrirons $w' \overset{c}{\underset{G}{\Longleftarrow}} w$ ssi

$$\exists \alpha_1 \in V^*, \exists \beta_1 \in \Sigma^*, \exists (p : A \to \beta) \in P \text{ tels que :}$$

(i) $w' = \alpha_1 A \beta_1$

(ii) $w = \alpha_1 \beta \beta_1$

(iii) $(\alpha_1^{(l)}, {}^{(k)}\alpha_1) \in c(p)$

Commentaires.

1. Examinons le cas où $l=0$. Ce cas revient lors de l'analyse de w, à ne pas consulter le contexte gauche de β. En effet, c est alors une application de P dans $\mathcal{P}(V_\Lambda^o, \Sigma_\Lambda^k)$ or $V^o = \{\Lambda\}$, de plus $\alpha^{(o)} = \Lambda$ quelque soit α, la condition (iii) devient donc :

$$(\Lambda, {}^{(k)}\beta_1) \in c(p) \subseteq \{\Lambda\} \times \Sigma_\Lambda^k$$

ceci revient a ne pas consulter le contexte gauche de β.

Le cas où $k=0$, revient à ne pas consulter le contexte droit de β.

2. Dans le cadre de notre définition, nous pouvons avoir pour un p donné :
$p : A \to \beta$

$$(\Lambda, t) \in c(p)$$

si $l=0$ nous avons vu ce qu'il en était, mais si $l \neq 0$ ceci veut dire que p n'est envisageable pour une réduction que si le contexte gauche de est vide, car $\alpha^{(l)} = \Lambda$ implique $\alpha = \Lambda$ si $l \neq 0$.
Même commentaire pour $k=0$.

3. Notons enfin que α_1, dans la définition peut être de longueur inférieur à l, en effet les éléments de V^l sont les mots formés sur V, et de longueur inférieure ou égale à l.

4. L'intérêt pour nous de la c-dérivation c'est partant de w trouver w' tel que $w' \overset{c}{\Longrightarrow} w$.

Il va de soit que nous pourrions considérer $\overset{c}{\Longrightarrow}$ tel que :

$$(w' \overset{c}{\Longleftarrow} w) \Longleftrightarrow (w' \overset{c}{\Longrightarrow} w)$$

la flèche dans un sens ou dans l'autre indiquant la même propriété, différenciant seulement la préoccupation du moment présent. L'application c dans le sens $\overset{c}{\Longrightarrow}$ voudrait signifier le désir de limiter l'application de certaines règles à un certain contexte, ce n'est pas ici notre préoccupation.

Conséquence du commentaire 1.

Lorsque $l=0$, nous dirons quelque fois que c est une application de $P \rightarrow \mathcal{P}(\Sigma_\Lambda^k)$, et lorsque $k = 0$ une application de $P \rightarrow \mathcal{P}(V_\Lambda^l)$.

Notation.

Nous noterons $\overset{*c}{\underset{G}{\Longleftarrow}}$ la clôture réflexive et transitive de $\overset{c}{\underset{G}{\Longleftarrow}}$

Définition c-analysable, c-d-analysable.

Nous dirons que w est c-analysable dans G ssi :

$$\exists S \overset{*c}{\underset{G}{\Longleftarrow}} w$$

et est c-d-analysable si $S \overset{*c}{\underset{G}{\Longleftarrow}} w$ est unique.

Définition. Grammaire c-(1,k)

Nous dirons qu'une grammaire $G = (N, \Sigma, P, \sigma)$ est c-(1,k) pour une application c de P dans $\mathcal{P}(V_\Lambda^1 \times \Sigma_\Lambda^k)$ ssi tout mot de L(G) est c-d analysable.

Nous allons donner quelques exemples.

e) Exemple de grammaire c-(1,1) pour deux applications distinctes

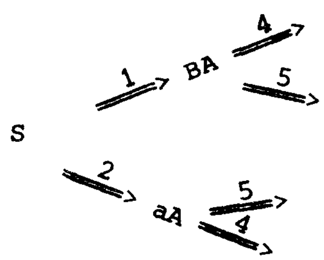
Soit $G = (V, \Sigma, P, S)$

P1 : $S \to Ba$ P2 : $S \to \alpha A$ P3 : $B \to b$

P4 : $A \to \Lambda$ P5 : $A \to a$.

Nous avons les dérivations les plus à droites suivantes :

au dessus des flèches nous indiquons le numéro de la production appliquée :



$B \overset{3}{\Longrightarrow} b$

$Ba \overset{3}{\Longrightarrow} ba$

a

aa

de ceci on en déduit une application c_1 telle que :

$c_1(P1) = \{(\Lambda,\Lambda)\}$ $c_1(P2) = \{(\Lambda,\Lambda)\}$

$c_1(P3) = \{(\Lambda,\Lambda), (\Lambda,a)\}$

$c_1(P4) = c_1(P5) = \{(B,\Lambda), (a,\Lambda)\}$

il suffit pour cela de noter dans quels contextes sont dérivées les parties gauches des règles.

Les c_1-analyses de b, ba, a, et aa sont les suivantes :

les notations sont :

. $\alpha\underset{\uparrow}{X}\,\beta$ ou $\alpha X\underset{\uparrow}{\Lambda}\beta$ la flèche indique le symbole pointé

. $\alpha X\underset{\uparrow}{Y}\beta \xRightarrow{c} \alpha X\underset{\uparrow}{\Lambda}Y \xRightarrow{c} \alpha X\underset{\uparrow}{Y}\beta$

ceci indique qu'aucune règle n'était c-applicable à X et à Λ, on pointe donc Y.

. $\alpha\underset{\uparrow}{\underline{\alpha' X}}\,\beta \xRightarrow{c} \alpha A\beta$

$\alpha'X$ souligné indique le mot qui a été réduit il vient :

. $\underset{\uparrow}{\Lambda}b \xRightarrow{c_1} \underset{\uparrow}{b}$ en effet p4 n'est pas applicable dans le contexte (Λ,b) :

$$b = \underset{\uparrow}{\Lambda}b \xRightarrow{c_1} \underset{\uparrow}{\underline{b}} \xRightarrow[P3]{c_1} \underset{\uparrow}{B} \xRightarrow{c_1} B\underset{\uparrow}{\underline{\Lambda}} \xRightarrow[p4]{c_1} \underset{\uparrow}{\underline{BA}} \xRightarrow[p1]{c_1} S$$

$$ba = \underset{\uparrow}{\Lambda}ba \xRightarrow{c_1} \underset{\uparrow}{\underline{b}}a \xRightarrow[p3]{c_1} \underset{\uparrow}{B}a \xRightarrow{c_1} B\underset{\uparrow}{\Lambda}a \xRightarrow{c_1} B\underset{\uparrow}{\underline{a}} \xRightarrow[p5]{c_1} \underset{\uparrow}{\underline{BA}} \xRightarrow[P1]{c_1} S$$

$$a = \underset{\uparrow}{\Lambda}a \xRightarrow{c_1} \underset{\uparrow}{a} \xRightarrow{c_1} a\underset{\uparrow}{\underline{\Lambda}} \xRightarrow[p4]{c_1} \underset{\uparrow}{\underline{aA}} \xRightarrow[p2]{c_1} S$$

$$aa = \underset{\uparrow}{\Lambda}aa \xRightarrow{c_1} \underset{\uparrow}{a}a \xRightarrow{c_1} a\underset{\uparrow}{\Lambda}a \xRightarrow{c_1} \underset{\uparrow}{\underline{aa}} \xRightarrow[P5]{c_1} aA \xRightarrow[P2]{c_1} S$$

tout les mots de L(G) sont c_1 analysables donc G est $c_1(1,1)$ car :

$$c_1 : P \to \mathcal{P}(V_\Lambda \times \Sigma_\Lambda).$$

Il y a d'autres applications c pour lesquelles G est c-analysable. En effet soit : c_2 telle que :

$$c_2(P1) = c_2(P2) = c_2(P3) = V_\Lambda \times \Sigma_\Lambda$$

ceci veut dire lorsque le membre droit de P1,P2 ou P3 est considéré au cours d'une c_2-analyse, les applications P1, P2 ou P3 sont appliquées pour la réduction, sans considération du contexte.

Peut-on se limiter pour P4 et P5 afin d'obtenir soit $l=0$ soit $k=0$.

Si $c_2(p4) = c_2(P5) = \{a,b\} \times \Sigma_\Lambda$ c'est-à-dire $k=0$ il vient :

$$\underset{\uparrow}{\Lambda}aa \overset{c_2}{\Longrightarrow} \underset{\uparrow}{a}a \overset{c_2}{\Longrightarrow} a\underline{\Lambda}a \overset{c_2}{\Longrightarrow} aAa$$

ce qui n'est pas correcte, le mot aa sera refusé.

Si $c_2(P4) = c_2(P5) = \{\Lambda\} \subseteq \Sigma_\Lambda$ c'est-à-dire $l=0$ il vient :

$$\underset{\uparrow}{\underline{a}} \overset{c_2}{\Longrightarrow} \underset{\uparrow}{\Lambda} \overset{c_2}{\Longrightarrow} a\underline{A} \overset{c_2}{\Longrightarrow} A\underset{\uparrow}{\Lambda} \overset{c_2^*}{\Longrightarrow} AA\ldots A$$

analyse qui ne conduit pas à l'axiome.

Donc :

$$c_2(P4) = c_2(P5) = \{(B,\Lambda),(a,\Lambda)\}$$

G est $c_2 - (1,1)$.

Nous constatons donc c_1 et c_2 permettent de la même manière l'analyse "top-down" des mots de L(G), et ceci avec la même efficacité. Cependant différent lorsqu'il s'agit d'analyser des mots n'appartenant pas à L(G) et conséquent de les rejetter.

Exemple :

$$\underset{\uparrow}{\Lambda}bb \overset{c_1}{\Longrightarrow} \underset{\uparrow}{b}b \overset{c_1}{\Longrightarrow} b\underset{\uparrow}{\Lambda}b \overset{c_1}{\Longrightarrow} b\underset{\uparrow}{b} \overset{c_1}{\Longrightarrow} bb\underset{\uparrow}{\Lambda}$$

$$\underset{\uparrow}{\Lambda}bb \overset{c_2}{\Longrightarrow} \underset{\uparrow}{\underline{b}}b \overset{c_2}{\Longrightarrow} \underset{\uparrow}{B}b \overset{c_2}{\Longrightarrow} B\underset{\uparrow}{\Lambda}b \overset{c_2}{=\!=} B\underset{\uparrow}{\underline{b}} \overset{c_2}{\Longrightarrow} B\underset{\uparrow}{B} \overset{c_2}{\Longrightarrow} BB\underset{\uparrow}{\Lambda}$$

c_1 différencie plus les règles de P, et contient donc plus d'information que c_2, elle est donc plus efficace pour l'analyse.

Exemple

Grammaire pour laquelle $\nexists c$, l, k telles que G est $c-(l,k)$.

. Donnons l'exemple de grammaire citée dans [7] montrant une grammaire pour laquelle il n'existe pas d'application c.

Soit :

P1 : $S \rightarrow ABc$

P2 : $S \rightarrow aBd$

P3 : $A \rightarrow a$

P4 : $B \rightarrow (B)$

P5 : $B \rightarrow *$

nous obtenons :

$$S \begin{cases} \xRightarrow{1} ABc \xRightarrow{4} A(B)c \xRightarrow{4^{n-1}} A(^nB)^nc \xRightarrow{5} A(^n*)^nc \xRightarrow{3} a(^n*)^nc \\ \xRightarrow{2} ABd \xRightarrow{4} a(B)d \xRightarrow{4^{n-1}} a(^nB)^nd \xRightarrow{5} a(^n*)^nd \end{cases}$$

Soit à analyser $a(^n*)^nd$, lorsque l'on pointe a il n'est pas possible de savoir s'il faut réduire par P3, il faudrait pour décider avoir accès à la fin du mot afin de savoir s'il se termine par c ou d, les mots de L(G) n'étant pas de longueur finie, l'application c n'existe pas .

Floyds fait remarquer qu'en analysant les mots de L(G) de la droite vers la gauche, l'analyse devient possible.

<u>Exemple de Grammaire c-(2,0)</u>.

Soit $G = (u, \Sigma, S, P)$

P1 : $S \rightarrow AY$

P2 : $Y \rightarrow Aa$

P3 : $A \rightarrow \Lambda$

Nous avons une unique dérivation possible dans G

$$S \Longrightarrow AY \Longrightarrow AAa \Longrightarrow Aa \Longrightarrow a$$

de cette dérivation on peut déduire une application c_1 telle que :

$$c_1(P1) = (\Lambda,\Lambda) \quad c_1(P2) = (A,\Lambda) \quad c_1(P3) = \{(A,a),(\Lambda,a)\}$$

a n'est pas c_1-analysable en effet :

$$\underset{\uparrow}{\underline{\Lambda}}a \xRightarrow[c_1]{} \underset{\uparrow}{A}a \xRightarrow[c_1]{} A\underset{\uparrow}{\underline{\Lambda}}a \xRightarrow[c_1]{} A\underset{\uparrow}{A}a \xRightarrow[c_1]{} AA\underset{\uparrow}{\underline{\Lambda}}a \xRightarrow[c]{} AAAa \xRightarrow[c]{} \dots$$

on constate donc qu'il n'est pas possible d'aboutir à l'axiome.

Soit alors c_2 telle que :

$$c_2(P1) = c_2(P2) = V_\Lambda \times \Sigma_\Lambda$$

et

$$c_2(P3) = \{(\Lambda A),(\Lambda)\} \times \Sigma_\Lambda$$

il vient :

$$\underset{\uparrow}{\underline{\Lambda}}a \xRightarrow[c_2]{} \underset{\uparrow}{A}a \xRightarrow[c_2]{} A\underset{\uparrow}{\underline{\Lambda}}a \xRightarrow[c_2]{} A\underset{\uparrow}{A}a \xRightarrow[c_2]{} AA\Lambda a$$

à ce stade P3 n'est pas applicable, car le contexte gauche de Λ est AA donc :

$$AA\underset{\uparrow}{\Lambda}a \xRightarrow[c_2]{} \underset{\uparrow}{\underline{AA}}a \xRightarrow[c_2]{} \underset{\uparrow}{\underline{AY}} \xRightarrow[c_2]{} S$$

donc : G est c-(2,0).

f) Montrons à présent l'équivalence des deux définitions :

<u>Proposition.</u>

Toute grammaire G est BRC(l,k) ssi elle est c-(l,k) pour une application c.

Preuve.

∐ Montrons que si la grammaire G est c-(1,k) alors elle est BRC (1,k). Nous allons le montrer par l'absurde.

Supposons donc G non BRC (e,k) ceci veut dire :

$$\exists w,\ \alpha_1,\ \beta_1,\ \alpha'_1,\ \beta'_1,\ A,\ A',\ \beta,\ \beta' \text{ tels que :}$$

$$S \overset{*d}{\Longrightarrow} \alpha_1\ A\ \beta_1 \overset{d}{\Longrightarrow} w = \alpha_1\,\beta\,\beta_1 \qquad (\alpha)$$

et

$$S \overset{*d}{\Longrightarrow} \alpha'_1\ A'\ \beta'_1 \overset{d}{\Longrightarrow} w = \alpha'_1\ \beta'\ \beta'_1 \qquad (\beta)$$

ce qui signifie que le mot w a deux poignets :

$$(A \rightarrow \beta,\ lg\,(\alpha_1\beta))$$

et

$$(A' \rightarrow \beta',\ lg(\alpha'_1\beta')).$$

Or toute application c devra être telle que :

$$(\alpha_1{}^{(1)},\ {}^{(k)}\beta_1)\ \in\ c(A \rightarrow \beta)$$

et

$$(\alpha'_1{}^{(1)},\ {}^{(k)}\beta'_1)\ \in\ c(A' \rightarrow \beta')$$

sinon il existerait des mots de L(G) qui ne seraient pas c-analysable, ceux qui utiliseraient l'une ou l'autre des dérivations (α) ou (β).

∐ Réciproquement, montrons que toute grammaire BRC (1,k) est c-(1,k) pour une application c.

Il nous faut construire cette application soit c définie par : pour toute production p : A → β

$$c(P) = \{(v,t) \mid |v| \leqslant 1 \ |t| \leqslant k, \exists v_1, t_1 \text{ tels que : } S \overset{*d}{\Longrightarrow} v_1 vAtt_1 \overset{d}{\Longrightarrow} v_1 v\beta tt_1\}$$

l'ensemble c(P) est fini et constructible. Il est immédiat, par construction même de l'application c, que si G n'était pas c-(1,k), elle ne serait pas BRC (1,k).

cqfd.

IV.2. APPLICATION.

$$\text{BRC } (1,1) \supsetneq \text{n.e.d.d}$$

$$\text{BRC } (1,0) \supseteq \text{n.e.d.d}_1$$

a) Pour montrer cette propriété il nous suffit de montrer que la grammaire canonique contexte libre associée à un n.e.d.d, est une grammaire BRC (1,1).

La propriété d'une telle grammaire est de rendre possible une analyse des feuilles vers la racine, analyse faite de manière déterministe moyennant la considération du contexte du mot réduit par une règle de G.

Or la grammaire canonique d'un n.e.d.d permet d'établir un parallèle étroit entre les configurations du n.e.d.d et les mots non nécessairement terminaux dérivés de l'axiome dans G. Comme la reconnaissance d'un mot dans le n.e.d.d se fait des feuilles vers la racine, nous allons en déduire dans G une analyse des feuilles vers la racine.

Nous montrerons en même temps que pour rendre cette analyse déterministe, nous n'aurons jamais besoin de considérer plus d'une lettre à gauche et plus d'une lettre à droite du mot réduit par une règle G lors d'une analyse, ceci permettant d'affirmer que la grammaire est BRC(1,1).

b) Nous allons reprendre l'exemple du § III.2.b et montrer qu'il existe une application $c : \Sigma^o \times \Gamma^o$ telle que tout mot de $L(G_A)$ soit c-d analysable.

Rappellons les productions de G_A

P1 $(\#_g, a_g) \rightarrow a$

P2 $(a_g, a_g) \rightarrow a$

P31 $(\#_g, A_g) \rightarrow (\#_g, a_g)b$

P32 $(a_g, A_g) \rightarrow (a_g, a_g)b$

P41 $(a_g, B_d) \rightarrow (a_g, A_g)b$

P51 $(\#_g, A_g) \rightarrow (\#_g, a_g)(a_g, B_d)$

P52 $(a_g, A_g) \rightarrow (a_g, a_g)(a_g, B_d)$

P61 $(\#_g, \bar{C}_d) \rightarrow (\#_g, A_g)$

P71 $(\#_g, C_d) \rightarrow (\#_g, a_g)$

P72 $(a_g, C_d) \rightarrow (a_g, a_g)$

P81 $(\#_g, C_d) \rightarrow (\#_g, a_g)(a_g, C_d)$

P82 $(a_g, C_d) \rightarrow (a_g, a_g)(a_g, C_d)$

P9 $(\Lambda, S_d) \rightarrow (\#_g, \bar{C}_d)$

P10 $(\Lambda, S_d) \rightarrow (\#_g, C_d)$

P11 $(\Lambda, S_d) \rightarrow \Lambda$

P12 $\sigma \rightarrow (\Lambda, S_d)$

σ étant l'axiome de G_A et $\Sigma_A = \{a,b\}$ nous avions :

$$L(G_A) = \{a^n b^n \ / \ n \geq 0\} \cup \{a^n \ / \ n \geq 0\}$$

A présent nous allons donner les valeurs de c. Nous montrons dans le § d), que toute production n'exige pas la connaissance du contexte, nous avons dans ce cas particulier :

$$C(P31) = C(P32) = C(P41) = C(P51) = C(P52) = C(P81) = C(P82)$$

$$= C(P9) = C(P10) = C(P12) = (\Sigma_{G_A} \cup \{\Lambda\} \times (V_{G_A} \cup \{\Lambda\})$$

c'est-à-dire que lorsque nous rencontrons la partie droite d'une de ces productions au cours d'une analyse, nous pouvons la réduire en la partie gauche, quelque soit le contexte.

Nous obtenons pour les autres productions :

1- 1 $(\#_g, a_g) \rightarrow a$ $c(P1) = \{\Lambda\} \times \Gamma^0$

2 $(a_g, a_g) \rightarrow a$ $c(P2) = \{(., a_g)\} \times \Gamma^0$ (1)

seul le contexte gauche est utilisé pour lever l'indétermination.

2- 72 $(a_g, c_d) \rightarrow (a_g, a_g)$ $c(P72) = \Sigma^0 \times \{\Lambda\}$

cette règle est source d'ambiguïté dans une analyse top-down, car lorsque l'on pointe un élément (a_g, a_g) nous devons décider entre appliquer cette production ou bien pointer le symbole suivant. De même :

3- 61 $(\#_g, \bar{C}_d) \rightarrow (\#_g, A_g)$ $c(P61) = \Sigma^0 \times \{\Lambda\}$

4- 71 $(\#_g, c_d) \rightarrow (\#_g, a_g)$ $c(P71) = \Sigma^0 \times \{\Lambda\}$

5- 11 $(\Lambda, S_d) \rightarrow \Lambda$ $c(P11) = \{\Lambda\} \times \{\Lambda\}$

P11 n'est applicable que si le contexte de Λ est vide.

(1) $\{(., a_g)\}$ désigne l'ensemble des éléments de la forme (X, a_g)

en fait on montrera dans la section IV.3 que seule les types de construction de règles 23 31 34 et 44 exigent éventuellement la connaissance du contexte gauche et que seules 5 et 6 demandent la connaissance du contexte droit. Remarquons tout de suite l'importance du marqueur $\#_d$, nous montrerons que la suppression de ce marqueur permettra de construire c telle que G_A soit BRC (1,0).

Dans le cas de notre exemple nous pouvons donc dire que si tout mot de $L(G_A)$ est c-analysable alors G_A est BRC (1,1).

La suppression de ce marqueur permettra de construire c telle que G_A soit BRC(1,0).

Dans le cas de notre exemple nous pouvons donc dire que si tout mot de $L(G_A)$ est c-analysable alors G_A est BRC(1,1)

Donnons un exemple de c-analysable dans G_A. Pour simplifier nous n'envisagerons pas à chaque fois la possibilité d'analyser Λ par P11 : $(\Lambda,S_d) \rightarrow \Lambda$, ayant remarqué que cette production n'est applicable que dans le contexte $\{\Lambda\} \times \{\Lambda\}$.

Soit à analyser a^3b^3

$$\underset{\uparrow}{a}aabbb \underset{c}{\Longrightarrow} (\#_g,\underset{\uparrow}{a_g})\ aabbb \underset{c}{\Longrightarrow} (\#_g,a_g)\ \underset{\uparrow}{a}abbb$$

$$\underset{c}{\Longrightarrow} (\#_g,a_g)(a_g,\underset{\uparrow}{a_g})\ abbb \underset{c}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)\ \underset{\uparrow}{a}bbb$$

$$\underset{c}{\Longrightarrow} (\#_g,a_g)(\underset{\uparrow}{a_g},a_g)(a_g,a_g)bbb \underset{c}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)\underset{\uparrow}{\underline{(a_g,a_g)}}bbb$$

$$\underset{c}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)(a_g,A_g)bb \underset{c}{\Longrightarrow} (\#_g,a_g)(a_g,a_g)\underset{\uparrow}{\underline{(a_g.A_g)}}\ bb$$

$$\underset{c}{\Longrightarrow} (\#_g,a_g)\underset{\uparrow}{\underline{(a_g,a_g)(a_g,B_d)}}b \underset{c}{\Longrightarrow} (\#_g,a_g)(a_g,A_g)b$$

$$\underset{c}{\Longrightarrow} (\#_g,a_g)\underset{\uparrow}{\underline{(a_g,A_g)b}} \underset{c}{\Longrightarrow} \underset{\uparrow}{\underline{(\#_g,a_g)(a_g,B_d)}} \underset{c}{\Longrightarrow} \underset{\uparrow}{\underline{(\#_g,A_g)}}$$

$$\underset{c}{\Longrightarrow} \underset{\uparrow}{\underline{(\#_g,\bar{c}_d)}} \underset{c}{\Longrightarrow} (\Lambda,S_d)$$

c) Nous allons tout d'abord donner deux propriétés qui sont des conséquences directes des lemmes et définitions donnés dans le chapitre III.

Lemme 4.1.

La grammaire canonique contexte libre G associée à part n.e.d.d A est telle que tout mot w de L(G) est dérivable de manière unique par une dérivation la plus à droite à partir de l'axiome de G.

Démonstration.

Ce résultat est une conséquence directe des lemmes 1 et 2 et du fait que la correspondance défini p du chapitre est biunivoque.
En effet supposons qu'il existe deux dérivation la plus à droite distinct pour un mot w soit :

$$\sigma \overset{d}{\Longrightarrow} \alpha_1 \overset{d}{\Longrightarrow} \alpha_2 \cdots \overset{d}{\Longrightarrow} \alpha_n \overset{d}{\Longrightarrow} w$$

$$\sigma \overset{d}{\Longrightarrow} \alpha'_1 \overset{d}{\Longrightarrow} \alpha'_2 \cdots \overset{d}{\Longrightarrow} \alpha'_p \overset{d}{\Longrightarrow} w$$

d'après le lemme 1 les α_i sont bien formées on peut donc leur appliquer le lemme 2. On peut associer à ces deux dérivations deux analyses dans A. Si les deux dérivations sont distinctes on a donc $\exists_i$ tel que :

$$\alpha_i \neq \alpha'_i$$

si une seule analyse dans A correspond à ces deux dérivations soit : n=p

$$K_o = [\#_g, w\#_d^o] \vdash K_1 \vdash \cdots \vdash K_n$$

mais alors pour i on a :

$$\alpha_i \neq \alpha'_i \text{ et } \alpha_i <\!-\!> K_i \text{ et } \alpha'_i <\!-\!> K_i$$

ceci n'est pas possible car la correspondance est biunivoque.

Donc aux deux dérivations les plus à droites distinctes dans G corresponds deux analyse distinctes dans A, ce qui n'est pas possible si A est déterministe.

cqfd.

Lemme 4.2.

Dans la grammaire canonique contexte libre G associée à un n.e.d on a : pour tout $w \in L(G)$

$$\exists \sigma \xRightarrow{*d} w \text{ ssi } \exists (\alpha_i)_{i=1}^{n} \text{ suite d'expressions bien formées}$$

$$\text{tq } \sigma \Longrightarrow \alpha_1 \Longrightarrow \ldots \Longrightarrow \alpha_n = w.$$

Ce lemme est en fait un corollaire du lemme 4.1. On peut appliquer ce lemme 4.1 à $\alpha_n \in \Sigma^*$ car tout mot de Σ^* est une expression bien formée.

La condition est suffisante car :

$$(\exists \sigma \xRightarrow{d} w) \xRightarrow{\text{lemme4.1}} (\exists \sigma \Longrightarrow \alpha_1 \Longrightarrow \ldots \Longrightarrow \alpha_u = w)$$

suite $(\alpha_i)_{i=1}^{n}$ bien formée.

La condition est nécessaire de manière évidente, elle est due à la forme des expressions bien formées, si on n'opère pas la dérivation la plus à droite l'expression obtenu n'est pas bien formée.

d) Nous allons à présent montrer le lemme le plus important du chapitre.

Lemme 4.3.

A toute grammaire canonique contexte libre G associé à un n.e.d.d correspond une application c :

$$c : P_G \to \mathcal{P}(V_\Lambda \times \Sigma_\Lambda)$$

telle que tout mot w de L(G) est c-d-analysable.

Démonstration.

Rappellons qu'une c-d-analyse est une analyse des feuilles vers la racine, effectuée de la gauche vers la droite du mot, telle qu'à chaque pas pour lever une éventuelle ambiguïté quant à la réduction à appliquer, c permet de consulter le contexte du mot faisant l'objet de la réduction.

Cette analyse doit être unique, pour tout mot du langage, cette condition est assurée ici par le lemme 4.1.

Nous avons vu d'après le lemme 4.2 qu'étant donnée une expression bien formée α telle que :

$$\sigma \overset{*d}{\Longrightarrow} \alpha \overset{*d}{\Longrightarrow} w, \; w \in \Sigma^*$$

la recherche de α' telle que :

$$\sigma \overset{d}{\Longrightarrow} \alpha' \overset{d}{\Longrightarrow} \alpha$$

répondait à un critère simple à savoir : il suffit de déterminer α' telle que α' soit bien formée (le lemme 4.1 nous assurant que cette analyse sera unique).

Nous savons que les expressions bien formées sont de trois types.

Rappellons les :

$$\forall i \; X^i \in V_G, \; a_i \in \Sigma_G^o, \; X \in V_G$$

1 pour $n \geq 0$

$$\alpha = (\Lambda_g, X_g^1) \; \dots \; (X_g^{n-1} X_g^n)\,(X_g^n, X_g)\, a_1 a_2 \; \dots \; a_p$$

2 pour $n \geq 0$

$$\alpha = (\Lambda_g, X_g^1) \; \dots \; (X_g^{n-1}, X_g^n)\,(X_g^n, Z_d)\, a_1 a_2 \; \dots \; a_p$$

avec $Z \in T_r(\#_d) \Longrightarrow a_1 a_2 \dots a_p = \Lambda$

$3\ \alpha \in \Sigma_G^*$

pour chacun de ces trois types nous pouvons appliquer dans le sens d'une réduction les règles de G.

Nous allons donc considérer chacun des types de règle de G en indiquant à chaque fois la valeur de l'application c, de manière à ce que α' tel que :

$$\alpha' \overset{d}{\Longrightarrow} \alpha$$

soit bien formée.

La recherche des valeurs de c sur P est tout à fait facile étant donné ce critère.

Il vient :

. L'application des réductions par les types de productions :

11 12 13 21 22 41 42 43

ne demande pas la consultation du contexte du membre droit de ces productions, dans α, ceci est évident, leur application donne automatiquement α' bien formée.

. Pour tout $A, B \in \Gamma \cup \Sigma$, $a \in \Sigma$ tout $p \in P$ telles que :

$p : (A_g, B_g) \to a$ (type 23)

on a :

$$c(P) = \{(X_g, A_g) \ / \ \forall X \in (\Gamma \cup \Sigma \cup M)^* \} \times \Sigma^0$$

il s'agit ici de respecter le fait que les éléments de Γ_G sont chaînés dans une expression bien formée.

Notons que pour ce type de production, on ne consulte qu'un seul élément sur la gauche du mot faisant objet de la réduction.

. Pour tout $X \in \Gamma \cup \Sigma$, tout $a \in \Sigma$ tout $p \in P$ telle que :

$$P : (\Lambda, X_g) \rightarrow a \quad \text{(type 31)}$$

ou

$$P : (\Lambda, X_g) \rightarrow a \quad \text{(type 32)}$$

ou

$$P : (\#_g, A_g) \rightarrow a \quad \text{(type 33)}$$

on a $c(P) = \{\Lambda\} \times \Sigma^o$

en effet les éléments telles que (Λ, X_g), (Λ, X_d) ou $(\#_g, A_g)$ ne peuvent être placés qu'au début de toutes expressions bien formées. En effet dans la définition des expressions bien formées on a $X_i \in \Gamma \cup \Sigma$ donc $X_i \neq \Lambda$. Ici encore seule le contexte gauche est consulté.

. Pour tout $Y \in \Gamma \cup \Sigma \cup M \cup \{\Lambda\}$ tout X, $A \in \Gamma \cup \Sigma$ par p telle que :.

$$P : (Y_g, X_d) \rightarrow (Y_g, A_g) \quad \text{(type 5)}$$

on a :

$$c(P) = V^o \times \{\Lambda\}$$

en effet ce type de règle implique nécessairement que $X \in T_r(\#_d)$, il faut donc pour que α' soit bien formée que le contexte droit de (Y_g, A_g), $a_1 \ldots a_p$ soit la chaine vide donc : $a_1 \ldots a_p = \Lambda$, sinon α' n'est pas bien formée.

. Le dernier cas est celui provenant de l'existence d'une règle de type 6, à savoir $\delta(\#_g, \#_d) = (\Lambda, X)$ ou $\delta(\#_g, \#_d) = (X, \Lambda)$, il vient :

$$P : (\Lambda, X_d) \rightarrow \Lambda$$

ou

$$P : (\Lambda, X_g) \rightarrow \Lambda$$

on a dans les deux cas :

$$c(P) = \{\Lambda\} \times \{\Lambda\}$$

Remarque 1.

Ces deux derniers types de règle, 5 et 6 sont les seuls qui exigent la connaissance du contexte droit.

Nous constatons donc que : $c : P \to \mathcal{P}(V_G^o \times \Sigma_G^o)$.

cqfd.

De ce qui précède nous déduisons étant donnée la remarque 1 précédente : que la supression du marqueur $\#_d$ amenant la supression du type de production 5 et 6 rend pour toutes productions la consultation du contexte droit inutile. Nous pouvons donc énoncer le corollaire.

e) Corollaire 4.1.

A toutes grammaires algébriques canoniques à un n.e.d.d acceptant un langage de la famille n.e.d.d_1, c'est-à-dire sans utilisation du marqueur droit, correspond une application :

$$c : P_G \to \mathcal{P}(V^o)$$

telle que tout mot de L(G) soit c-analysable.

f) Nous pouvons à la suite de ces résultats énoncer le théorème :

Théorème.

Pour tout langage de la famille n.e.d.d, il existe une grammaire le reconnaissant qui est BRC (1,1).

Pour tout langage de la famille n.e.d.d_1, il existe une grammaire le reconnaissant qui est BRC (1,0).

-BIBLIOGRAPHIE-

1 CHOMSKY N.
"On certain formal properties of grammars"
Inf and Control 2 137-165 (1959)

2 CHOMSKY N. and MILLER G.A.
"Finite state languages"
Inf and Control 1 91.112 (1958)

3 COLMERAUER A.
"Total precedence relation"
JACM 17 1 (Jan. 1970) 14-30.

4 COURCELLE B.
"Le problème de l'équivalence"
Rapport de recherche n°48 IRIA (1974)

5 EILENBERG S.
Communication privé
1975 VIC-SUR-CERE

6 FLOYD R.W.
"Syntactic Analysis and operator precedence"
JACM 10,3 (July 1963) 316-333

7 FLOYD R.W.
"Bounded context syntactic analysis"
Comm. A.C.M. 7,2 (Feb. 1964) 62.67

8 GINSBURG S.
"The mathematical theory of Context Free Languages"
Mc Graw-Hill, New York, 1966

9 GINSBURG S. and GREIBACH S.
"Mappings which preserves context sensitive languages"
Inf and Control 9 563-582

10 GINSBURG S. and GREIBACH S.
"Deterministic context free languages"
Inf and Control 9 620-648

11 HARRISON M.A. and HAVEL
"Strict deterministic Grammars"
JCSS 7 237-277 (1973)

12 HARRISON M.A. and HAVEL
"On the parsing of deterministic languages"
JACM vol 21 n°4 Oct 1974 p : 525-548

13 KNUTH D.E.
"On the translation of languages from left to right"
Inf and Control 8 (1965) 607-639

14 KORENJAK HOPCROFT
"Simple deterministic languages"
I.E.E.E 1966

15 KURODA S.Y.
"Classes of languages and linear bounded Automata"
Inf and Control 7 207-222 (1964)

16 COURCELLE B.
Doctorat d'Etat. Paris

17 AZRA-JAULIN
"Récursivité"
Collection "Programmation" Gauthier - Villars

18 GREIBACH S.A.
"A new normal from theorem for context free phrase grammars"
(1965) JACM 12 42.52