Informatique théorique 2
notes de cours

Géraud Sénizergues

Année 2010-2011 [1]

[1] dernière mise à jour : 29 mars 2011

Table des matières

1 Langages rationnels de mots **9**
1.1 Mots, monoïdes 9
1.2 Automates finis 11
1.3 Lemme d'itération 15
1.4 Propriétés de clôture 17
1.5 Automate minimal 21
1.6 Expressions rationnelles 31

2 Langages algébriques de mots **35**
2.1 Arbres planaires ordonnés 35
2.2 Grammaires algébriques 36
2.3 Arbres de dérivation 42
2.4 Formes normales pour les grammaires algébriques 45
2.5 Lemme d'itération 54
2.6 Propriétés de clôture 55
2.7 Automates à pile 60

3 Langages rationnels de termes **67**
3.1 Termes, F-algèbres 67
3.2 Automates finis 72
3.3 Lemme d'itération 75
3.4 Propriétés de clôture 78
3.5 Automate minimal 86
3.6 Expressions rationnelles 89

4 Logique Monadique du Second Ordre **93**
4.1 Syntaxe et sémantique 93
4.2 Ensembles de mots définissables en MSO 97
4.3 Ensembles de termes definissables en MSO 102

5 Langages récursivement énumérables de mots **105**
5.1 Grammaires contextuelles 105
5.2 Grammaires à structure de phrase 107
5.3 Hiérarchie de Chomsky 112
5.4 Un problème 113

Introduction

Que va-t-on étudier ? Le module d'Informatique Théorique 2 traite de la théorie des *Langages Formels*. Le mot "langages" est certainement connu du lecteur ; l'épithète "formels" précise que nous étudions les mots des langages du point de vue de leur "forme" et non de ce qu'ils pourraient évoquer à un locuteur de ce langage. Cette notion de forme s'oppose à celle du " sens" des mots et des phrases, qui est certes essentielle, et fait l'objet de la " sémantique" des langages. On se concentre ici sur la forme, non par mépris du sens (ce qui serait absurde), mais par ce que nous appliquons une stratégie " diviser pour régner" dans le domaine de l' étude des langages : cette démarche s'avère souvent efficace tant pour le chercheur que pour le pédagogue et l' étudiant. En fait, on s' apercevra en cours de route qu'une bonne compréhension de la forme fournit des outils pertinents pour essayer de déduire le sens (des mots, des phrases, du discours) de l'analyse de leur forme.

Pourquoi ? Historiquement, deux grandes préoccupations ont conduit à s'intéresser aux langages formels.

P1 Les mathématiciens de la seconde moitié du 19ième siècle ont cherché (et réussi) à donner aux mathématiques une forme plus rigoureuse, donc soulevant moins de controverses ; la clé de cette rigueur est la définition d'un langage artificiel, beaucoup plus pauvre que la langue naturelle (au moyen de laquelle on écrivait les mathématiques à cette époque) mais, en revanche, définie de façon beaucoup plus précise. Nous appellerons ce langage la " langue mathématique formelle". En particulier, on peut aisément vérifier si un énoncé prétendûment écrit dans cette langue est effectivement un énoncé

correctement formé de cette langue. On peut également vérifier, en principe du moins, si une suite d'énoncés (correctement formés) est réellement une *démonstration* dans une axiomatique donnée, ou non.

P2 Depuis le 18ième siècle au moins, les linguistes s'efforcent de décrire les langues naturelles. Au début du 20ième siècle, ils se sont préoccupés d'inventer des *modèles mathématiques* de leur objet d'étude : les langues. Ainsi les AB-grammaires, apparues dans les années 1930 décrivent certaines phrases simples des langues naturelles. Les grammaires algébriques (qui sont apparues dans les années 1950 et que nous étudierons ici) , les grammaires catégorielles (qui datent de la même époque) sont des évolutions des AB-grammaires. Depuis lors, de nombreux autres modèles ont vu le jour et continuent à être inventés ou perfectionnés par les spécialistes de *linguistique computationnelle*.

Une fois la théorie lancée par ces deux motivations, des liens fructueux (et prometteurs, car rien n'est terminé) sont apparus avec les domaines suivants :

- la conception des Langages de Programmation : i.e. la construction de langues artificielles qui permettent d'exprimer des calculs ; de tels langages doivent être suffisamment riches pour satisfaire les désirs des utilisateurs (qui souhaitent réaliser des calculs compliqués) et doivent aussi pouvoir être compris des ordinateurs, dont le langage natif, le " langage machine", est très primitif. La traduction des programmes évolués vers les langages machine est l'objet de la *compilation* ;
- la Théorie des modèles : i.e. l'étude des structures ensemblistes qui rendent vraie (ou fausse) une formule logique ou un ensemble de formules logiques ; ce sujet est abordé à la section 4.
- La Théorie de la complexité des algorithmes : un algorithme peut être défini comme une machine ou bien un programme d'une machine ; on définit alors une notion de complexité liée au type de machines utilisé. La notion informellement utilisée dans les enseignements d'ASD est souvent celle associée aux machines dites " R.A.M." (Random Access Memory machines).
- La Théorie des Jeux : initialement il s'agissait de modéliser le comportement des acteurs d'une économie de marché ; il s' avère que les concepts-clés de cette théorie (notion de stratégie gagnante, de jeu déterminé) sont très pertinents pour traiter de problèmes d'automates.

- l'Algèbre : il s'agit ici de l'étude des structures telles que les groupes, les semi-groupes, les monoïdes, etc ... La notion de langage rationnel est très liée à celle de monoïde fini par exemple ; des liens subtils entre les expressions rationnelles et les monoïdes conduisent à des algorithmes que l'on aurait du mal à concevoir directement sur les automates finis.
- La Combinatoire : c'est l'art (et la science) du dénombrements d'objets discrets définis par leurs propriétés ou bien par des récurrences. Il s'avère que les récurrences d'un côté, et les grammaires, de l'autre, sont souvent très liées.
- La Biologie moléculaire : une molécule d'ADN (ou d'ARN) est souvent modélisée comme un mot sur un alphabet de quatre lettres (A,C,D,G). Il s' agit de mots très longs et la conception d'algorithmes efficaces de traitement de telles molécules obtenues expérimentalement passe parfois par la modélisation par des grammaires, des automates, des systèmes de réécriture ou d'autres systèmes formels plus complexes.

Quelles autres UE sont concernées ?

En "amont" :

L'Unité d' Enseignement intitulée "Informatique Théorique 1" fournit les bases indispensables concernant les automates finis, les expressions rationnelles et plus généralement les mathématiques discrètes.

L'Algorithmique, qu'elle soit générale et associée aux " Structures de Données", ou bien focalisée sur les Graphes, est très utile pour concevoir des algorithmes portant sur les langages formels.

En "aval" :

Les concepts, propriétés et algorithmes que nous introduisons ici sont
- essentiels pour les UE d' Analyse Syntaxique (L3) et de Compilation (M1)
- utiles pour l'UE de Modèles de Calcul (M1)
- utiles pour l'UE de Logique (M1)
- essentiels pour l'UE de Linguistique Computationnelle (M2)

Chapitre 1

Langages rationnels de mots

1.1 Mots, monoïdes

Définition 1.1.1 *Un monoïde est un triplet $\mathcal{M} = (M, \cdot, e)$, où M est un ensemble M, $\cdot$ est une opération associative sur M et e est l' élément neutre de M pour cette loi.*

Soit X un ensemble. Un *mot* sur l'alphabet X est, par définition, une application partant d'un intervalle $[1, n]$ de $\mathbb{N}$ et arrivant dans l'ensemble X : $u : [1, n] \to X$.
Le mot particulier correspondant à l'application vide est dénommé le "mot vide" et noté ϵ.
On dénote par X^*, l'ensemble des mots écrits sur l'alphabet X.
Soient $u : [1, n] \to X, v : [1, m] \to X$ deux mots. Le *produit de concaténation* de u par v est le mot w obtenu en juxtaposant u et v. Formellement, $w = u \cdot v$ est défini par :

$$\mathrm{dom}(w) = [1, n + m], \quad \forall i \in [1, n], w[i] = u[i]; \quad \forall j \in [1, m], w[i + j] = v[j].$$

Remarquons que, pour tout $u \in X^*$

$$u \cdot \epsilon = \epsilon \cdot u = u.$$

Définition 1.1.2 *Pour tout alphabet X, $(X^*, \cdot, \epsilon)$ est un monoïde. On l'appelle le "monoïde libre engendré par X".*

Définition 1.1.3 *Soit $\mathcal{M} = (M, \cdot, e)$. Un* sous-monoïde *de $\mathcal{M}$ est un triplet $\mathcal{M}' = (M', \odot, e')$ tel que :*
– $M' \subseteq M$
– $e' = e$
– $\forall m, m' \in M', m \odot m' = m \cdot m'$.

Si I est un ensemble d'indices et si $\forall i \in I, \mathcal{M}_i = (M_i, \cdot, e)$ est un sous-monoïde de $\mathcal{M}$, alors $(\bigcap_{i\in I} M_i, \cdot, e)$ est un sous-monoïde de $\mathcal{M}$.

Définition 1.1.4 *Soit Y une partie d'un monoïde M. On appelle* sous-monoïde engendré par Y, *le plus petit sous-monoïde de $\mathcal{M}$ contenant Y. On le note Y^*.*

D'après ce qui précède, Y^* est l'intersection de tous les sous-monoïdes de $\mathcal{M}$ qui contiennent Y.

Exemple 1.1.5 *Soit $\mathcal{N} = (\mathbb{N}, +, 0)$. Soit A l'ensemble des nombres pairs et B l' ensemble des nombres impairs. $(A, +, 0)$ est le sous-monoïde de $\mathcal{N}$ engendré par $\{2\}$ tandis que $(B, +, 0)$ n' est pas un sous-monoïde de $\mathcal{N}$.*

Définition 1.1.6 *Soient $\mathcal{M} = (M, \cdot, e)$,$\mathcal{M}' = (M', \times, e')$ des monoïdes. On appelle* homomorphisme *(de monoïde) allant de $\mathcal{M}$ dans $\mathcal{M}'$, toute application h de M dans M' verifiant :*
– $h(e) = e'$
– $\forall m, m' \in \mathcal{M}, h(m \cdot m') = h(m) \times h(m')$

Exemple 1.1.7 *L'application qui à un mot u associe sa longueur $|u|$ est un homomorphisme de $(X^*, \cdot, \varepsilon)$ dans $(\mathbb{N}, +, 0)$.*

L'application $n \mapsto 2^n$ est un homomorphisme de $(\mathbb{N}, +, 0)$ dans $(\mathbb{N}, \times, 1)$.

Théorème 1.1.8 *Soit X un alphabet. Soit $(M, \cdot, e)$ un monoïde et soit h une application de X dans M. Il existe un et un seul homomorphisme $\tilde{h} : X^* \longrightarrow M$ tel que $\forall x \in X, \tilde{h}(x) = h(x)$.*

Le monoïde X^* est appelé le monoïde *libre* engendré par X. La propriété énoncée dans le théorème 1.1.8 est la *propriété universelle* du monoïde libre sur X. On peut démontrer qu'elle caractérise (à isomorphisme près) le couple (X, X^*).
Preuve : Existence : Posons $\tilde{h}(\varepsilon) = e$ et $\tilde{h}(x_1...x_n) = h(x_1) \cdot ... \cdot h(x_n)$. Il est facile de voir que $\tilde{h}$ est bien un homomorphisme.

Unicité :Soient g et $g^{'}$ deux homomorphismes de X^* dans M tels que $\forall x \in X, g(x) = g'(x)$. Alors $g(\varepsilon) = g^{'}(\varepsilon) = e$ et pour tout mot $u = x_1...x_n$,

$$g(u) = g(x_1) \cdot ... \cdot g(x_n) = g^{'}(x_1) \cdot ... \cdot g^{'}(x_n) = g^{'}(u). \tag{1.1}$$

C'est à cause de ce théorème que le monoïde X^* est appelé le monoïde *libre* engendré par X. □

On appelle *langage* sur l'alphabet X, toute partie de X^* .

1.2 Automates finis

Définition 1.2.1 *Un* automate *est un 5-uplet,* $\mathcal{A} =< X, Q, D, A, \delta >$ *où*
– X *est un ensemble fini,l'alphabet d' entrée*
– Q *est un ensemble, l' ensemble des états*
– $D \subset Q$ *est l' ensemble des etats de départ*
– $A \subset Q$ *est l' ensemble des etats d'arrivée*
– $\delta \subset Q \times X \times Q$ *est l' ensemble des transitions*

Définition 1.2.2 *Un automate* fini *est un automate* $\mathcal{A} =< X, Q, D, A, \delta >$ *dont l'ensemble d' états,* Q*, est fini.*

Exemple 1.2.3 *Posons*

$X = \{a, b\}, Q_1 = \{1, 2\}, D_1 = \{1\}, A_1 = \{2\}, \delta_1 = \{(1, a, 1), (1, b, 2), (2, b, 2)\}.$

$\mathcal{A}_1 =< X, Q_1, D_1, A_1, \delta_1 >$ *est un automate fini.*

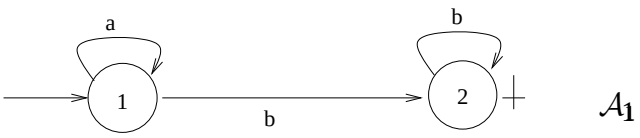


FIG. 1.1 – representation graphique.

Exemple 1.2.4 *Soit*

$$Q_2 = \{1, 2, 3, 4, 5\}, D_2 = \{1, 4\}, A_2 = \{3, 5\},$$

$$\delta_2 = \{(1, a, 1), (1, b, 1), (1, a, 2), (2, a, 3), (4, a, 5), (5, b, 4)\}.$$

$\mathcal{A}_2 =< X, Q_2, D_2, A_2, \delta_2 >$ *est un automate fini.*

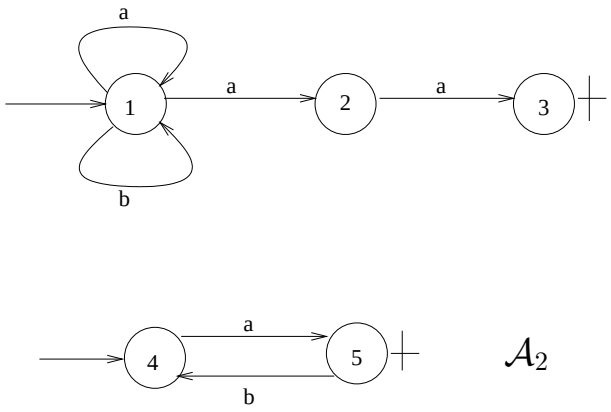


FIG. 1.2 – representation graphique.

Soit $\mathcal{A} =< X, Q, D, A, \delta >$ un automate fini.
Un mot w de δ^* est un *chemin* dans $\mathcal{A}$ ssi il s'écrit

$$w = (q_1, x_1, q_1^{'})(q_2, x_2, q_2^{'})...(q_i, x_i, q_i^{'})...(q_n, x_n, q_n^{'}) \tag{1.2}$$

et il vérifie la condition :

$$w = \varepsilon \text{ ou } \forall i \in [1, n-1], q_i^{'} = q_{i+1}$$

– n est la longueur du chemin

– ce chemin mène de l'état q_1 à l'état q_n
– on convient que ε mène de q à q (pour tout $q \in Q$)

Définition 1.2.5 *On appelle* trace *l'homomorphisme* $t : \delta^* \longrightarrow X^*$ *tq*

$$\forall (q, x, q^{'}) \in \delta, t((q, x, q^{'})) = x$$

Définition 1.2.6 *Un mot* $u \in X^*$ *est accepté (ou reconnu) par* $\mathcal{A}$*, ssi il existe un chemin* w *dans* $\mathcal{A}$ *menant d'un état* $q_d \in D$ *à un état* $q_a \in A$ *et tel que* $t(w) = u$*.*

Définition 1.2.7 *On appelle langage accepté (ou reconnu) par* $\mathcal{A}$*, noté* $L_{\mathcal{A}}$*, l'ensemble des mots acceptés (ou reconnnus) par* $\mathcal{A}$*.*

Exemple 1.2.8 *(suite des exemples 1.2.3,1.2.4).*

$$L_{\mathcal{A}_1} = \{a^n b^m | n \geq 0, m \geq 1\}; \;\; L_{\mathcal{A}_2} = \{uaa | u \in \{a, b\}^*\} \cup \{a(ba)^n | n \geq 0\}$$

Notation :

$$\mathcal{A} < p, q >=< X, Q, \{p\}, \{q\}, \delta >$$

$$L_{\mathcal{A}} = \cup_{(p,q) \in D \times A} L_{\mathcal{A}<p,q>}$$

□

Définition 1.2.9 *Un langage* $L \subset X^*$ *est dit* reconnaissable *ssi, il existe un automate fini* $\mathcal{A}$ *sur* X*, tq* $L = L_{\mathcal{A}}$*. On note* $Rec(X^*)$ *l'ensemble des langages reconnaissables sur* X^**.*

Exemple 1.2.10 $\mathcal{A}_1$ *est un automate deterministe (localement il n'a qu'un choix par lettre). Ce n'est pas le cas de* $\mathcal{A}_2$*.*

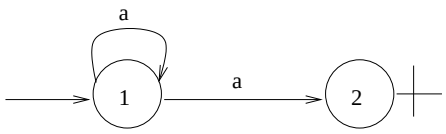


FIG. 1.3 – 2 choix possibles dans l'état 1 pour lire a

Définition 1.2.11 *Un automate* $\mathcal{A} =< X, Q, D, A, \delta >$ *est* déterministe *ssi*

- *(1)* $Card(D) = 1$
- *(2)* $\forall q \in Q, \forall x \in X, Card(\{q^{'} \in Q, (q, x, q^{'}) \in \delta\}) \leq 1$

$\mathcal{A}$ *est dit* déterministe complet *ssi*
- *(1)* $Card(D) = 1$
- *(2)* $\forall q \in Q, \forall x \in X, Card(\{q' \in Q | (q, x, q') \in \delta\}) = 1$

Remarque 1.2.12 *La simulation d'un automate fini déterministe complet par un programme est très simple.*

Notation :
Lorsque $\mathcal{A}$ est déterministe, on utilise la notation fonctionnelle

$$\delta : Q \times X \longrightarrow Q$$

La fonction δ s'étend en une unique fonction $\tilde{\delta} : Q \times X \longrightarrow Q$ vérifiant :
- $\forall q \in Q, \tilde{\delta}(q, \epsilon) = q$
- $\forall q \in Q, \forall u \in X^*, \forall x \in X, \tilde{\delta}(q, u, x) = \delta(\tilde{\delta}(q, u), x)$

Si $\mathcal{A}$ est déterministe complet, alors $\tilde{\delta}$ est une application.

Proposition 1.2.13 *Un langage est reconnaissable ssi il est reconnu par un automate fini déterministe complet.*

Preuve : Soit $L = L_{\mathcal{A}}$ où $\mathcal{A} =< X, Q, D, A, \delta >$. Construisons un automate fini déterministe complet $\mathcal{B}$ tel que $L_{\mathcal{A}} = L_{\mathcal{B}}$. Posons

$$\mathcal{B} =< X, \mathcal{P}(Q), \{D\}, \mathcal{R}, \triangle >$$

où $\mathcal{R} = \{P \in \mathcal{P}(Q), P \cap A \neq \emptyset\}$

$$\triangle : \mathcal{P}(Q) \times X \longrightarrow \mathcal{P}(Q)$$

$$(P, x) \longrightarrow \{q \in Q, \exists p \in P, (p, x, q) \in \delta\}$$

On démonte par récurrence sur $|u|$ que : $\forall u \in X^*, \forall P \in \mathcal{P}(Q)$,

$$\tilde{\triangle}(P, u) = \{q \in Q, \exists p \in P, \exists w \in \delta^*, t(w) = u \text{ et } w \text{ mène de } p \text{ à } q\}$$

$$\begin{array}{rcl} u \in L_{\mathcal{A}} & \Leftrightarrow & \exists w \in \delta^*, \exists q_d \in D, \exists q_a \in A, t(w) = u \text{ et } w \text{ est un chemin qui mène de } q_d \text{ à } q_a \\ & \Leftrightarrow & \exists q_a \in A, \tilde{\triangle}(D, u) \ni q_a \\ & \Leftrightarrow & \tilde{\triangle}(D, u) \cap A \neq \emptyset \\ & \Leftrightarrow & \tilde{\triangle}(D, u) \in \mathcal{R}. \end{array}$$

$\mathcal{B}$ est déterministe complet car $\triangle$ est une application. □

Remarque 1.2.14 *Si $\mathcal{A}$ a n états, alors $\mathcal{B}$ a 2^n états*

1.3 Lemme d'itération

Question : Les langages suivants sont-ils rationnels ?

$$\begin{array}{rcl} L_1 & = & \{a^n b^n | n \geq 1\}, \\ L_2 & = & \{f \in \{a, b, c\}^* || f|_a = |f|_b\} \\ L_3 & = & \{f \in \{a, b\}^* || f|_a = |f|_b \text{ et pour tout } g \leq f, |g|_a \geq |g|_b\}. \end{array}$$

Supposons que $\mathcal{A} =< X, Q, D, A, \delta >$ reconnaît L_1.

Soit $N = Card(Q)$, soit w un calcul de $\mathcal{A}$ sur le mot $a^N b^N$

$$w = (q_0, x_1, q_1)...(q_{i-1}, x_i, q_i)...(q_{2N-1}, x_{2N}, q_{2N})$$
$$\exists i, j \in \mathbb{N}, 0 \leq i < j \leq N, q_i = q_j.$$

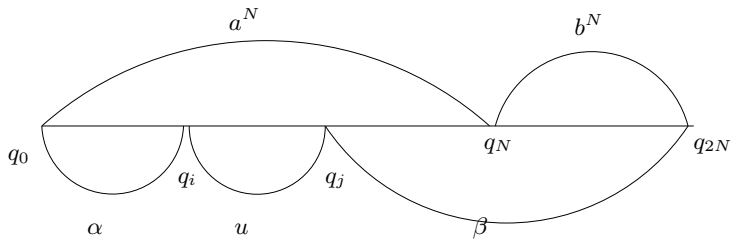


FIG. 1.4 – q_i est répété

$a^N b^N = \alpha u \beta$ où $q_0 \xrightarrow{\alpha}_{\mathcal{A}} q_i$, $q_i \xrightarrow{u}_{\mathcal{A}} q_j$, $q_j \xrightarrow{\beta}_{\mathcal{A}} q_{2N}$ On en déduit que : $\forall k \in \mathbb{N}, \alpha u^k \beta \in L_{\mathcal{A}}$, en vertu du calcul :

$$q_1 \xrightarrow{\alpha}_{\mathcal{A}} q_i, q_i \xrightarrow{u^k}_{\mathcal{A}} q_i = q_j, q_j \xrightarrow{\beta}_{\mathcal{A}} q_{2N}.$$

Mais $\alpha u^2 \beta \notin L_{\mathcal{A}}$, car $|\alpha u^2 \beta|_a > |\alpha\beta|_b$. L'existence de $\mathcal{A}$ conduit donc à une contradiction.

Lemme 1.3.1 *(de l'étoile) Soit L un langage rationnel.Alors il existe une constante $N > 0$, t.q. pour tout mot $v \in L$,si $|v| \geq N$ alors il existe des mots $\alpha, u, \beta \in X^*$ tq.*
- *(1) $v = \alpha u \beta$*
- *(2) $0 < |u| < N$*
- *(3) $\alpha u^* \beta \subseteq L$*

Preuve : Supposons que $\mathcal{A} =< X, Q, D, A, \delta >$ reconnaît L. Soit $N = Card(Q)$. Soit $v \in L, |u| \geq N$. Soit w un calcul réussi de $\mathcal{A}$ sur le mot v :
- $w = (q_0, x_1, q_1)...(q_{i-1}, x_i, q_i)...(q_{n-1}, x_n, q_n)$ où $n = |u|$
- w est un chemin
- $q_0 \in D, q_n \in A$
- $t(w) = v$.

Comme $n + 1 \geq N + 1 = Card(Q) + 1$, $\exists i, j \in \mathbb{N}, 0 \leq i < j \leq n$ tels que $q_i = q_j$. Soit $(k, \ell) \in [0, n]^2, k < \ell$, tels que $q_k = q_\ell$ et $\forall (i, j) \in [0, n]^2$
Si $i < j$ et $q_i = q_j$ alors $\ell - k < j - i$. On a $\ell - k < N$. Posons :

$$\alpha = t((q_0, x_1, q_1)...(q_{k-1}, x_k, q_k))$$

$$u = t((q_k, x_{k+1}, q_{k+1})...(q_{\ell-1}, x_\ell, q_\ell))$$

$$\beta = t((q_\ell, x_{\ell+1}, q_{\ell+1})...(q_{n-1}, x_n, q_n))$$

On vérifie alors que $\alpha u \beta = t(w) = v$, i.e. le point (1) est vrai.
$\ell > k \Rightarrow |u| > 0, \ell - k < N \Rightarrow |u| < N$, ce qui prouve le point (2).
Pour tout entier $m \geq 0$:

$$q_0 \xrightarrow{\alpha}_{\mathcal{A}} q_k, q_k \xrightarrow{u^m}_{\mathcal{A}} q_k = q_\ell, q_\ell \xrightarrow{\beta}_{\mathcal{A}} q_n.$$

Donc $\alpha u^* \beta \subseteq L$, i.e. (3). □

1.4 Propriétés de clôture

Proposition 1.4.1 *Si $L \subseteq X^*$ est reconnaissable alors $X^* - L$ est reconnaissable.*

Preuve : Soit $L = L_{\mathcal{A}}$, où

$$\mathcal{A} =< X, Q, d, A, \delta >$$

(δ application).
Posons

$$\mathcal{B} =< X, Q, d, Q - A, \delta > .$$

Montrons que $X^* - L_{\mathcal{A}} = L_{\mathcal{B}}$ Soit $u \in X^*$. On a la suite d' équivalences :
$u \in X^* - L_{\mathcal{A}} \Leftrightarrow \tilde{\triangle}(d, u) \notin A \Leftrightarrow \tilde{\triangle}(d, u) \in Q - A$. □

Proposition 1.4.2 *Si $L \subseteq X^*$ est fini, alors L est reconnaissable.*

Preuve : Soit $L \subset X^*, Card(L) < +\infty$ Soit $\mathcal{A} =< X, Q, d, A, \delta >$ où

$$Q = Pref(L) = \{u \in X^*, \exists v \in X^*, u \cdot v \in L\}$$

$$\delta : Q \times X \longrightarrow Q$$

$$(u, x) \mapsto ux$$

si $ux \in Pref(L)$, sinon (u, x) n'a pas d'image par δ.

$$d = \varepsilon$$

$$A = L$$

On a bien :

$$\forall u \in X^*, \tilde{\delta}(\varepsilon, u) \in L \Leftrightarrow u \in L.$$

□

Exemple 1.4.3 *Lorsque* $L = \{a, aa, aab, ab, ba, bba\}$, $\mathcal{A}$ *est l'automate représenté sur la figure 1.5.*

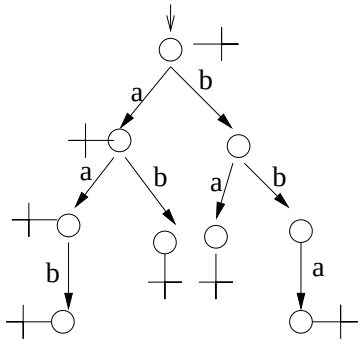


FIG. 1.5 –

Proposition 1.4.4 *Rec*(X^*) *est fermé par union.*

Preuve : Soient $L_1 = L_{\mathcal{A}_1}, L_2 = L_{\mathcal{A}_2}$

$$\mathcal{A}_1 =< X, Q, D_1, A_1, \delta_1 >, \mathcal{A}_2 =< X, Q, D_2, A_2, \delta_2 >$$

On suppose que $Q_1 \cap Q_2 = \emptyset$. Posons :

$$\mathcal{A} =< X, Q_1 \cup Q_2, D_1 \cup D_2, A_1 \cup A_2, \delta_1 \cup \delta_2 > .$$

On vérifie que $L_{\mathcal{A}} = L_{\mathcal{A}_1} \cup L_{\mathcal{A}_2}$. □

Proposition 1.4.5 *$Rec(X^*)$ est fermé par produit.*

Idée naïve : supposons que $L_1 = L_{\mathcal{A}_1}, L_2 = L_{\mathcal{A}_2}$. On construit l'union des deux automates $\mathcal{A}_1, \mathcal{A}_2$, dans laquelle on fusionne l'état final de $\mathcal{A}_1$ avec l'état initial de $\mathcal{A}_2$.

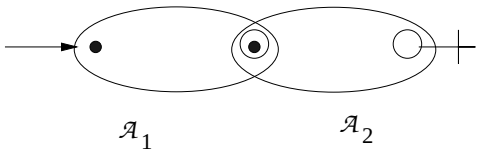


FIG. 1.6 – L' idée ...

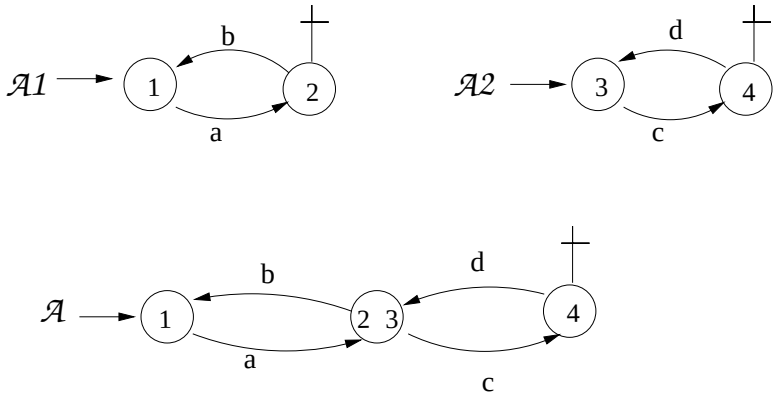


FIG. 1.7 – ... et son contre-exemple !

Dans l'exemple de la figure 1.7, on obtient un automate $\mathcal{A}$ tel que :

$$acdbac \in L_{\mathcal{A}} - L_{\mathcal{A}_1} \cdot L_{\mathcal{A}_2}$$

Il faut donc améliorer cette idée de départ.

Preuve de la proposition 1.4.5 : Considérons des automates finis $\mathcal{A}_1$, $\mathcal{A}_2$ tels que $L_1 = L_{\mathcal{A}_1}, L_2 = L_{\mathcal{A}_2}$.

$$\mathcal{A}_1 =< X, Q_1, D_1, A_1, \delta_1 >$$
$$\mathcal{A}_2 =< X, Q_2, D_2, A_2, \delta_2 >$$

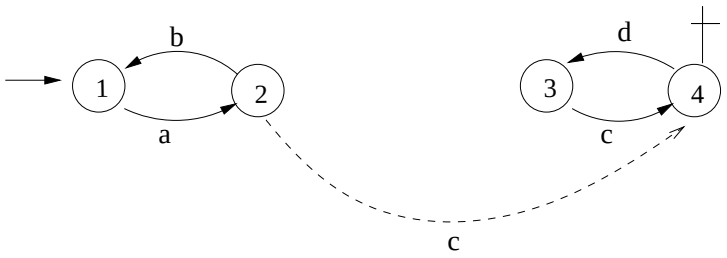


FIG. 1.8 – Exemple de produit (de concaténation).

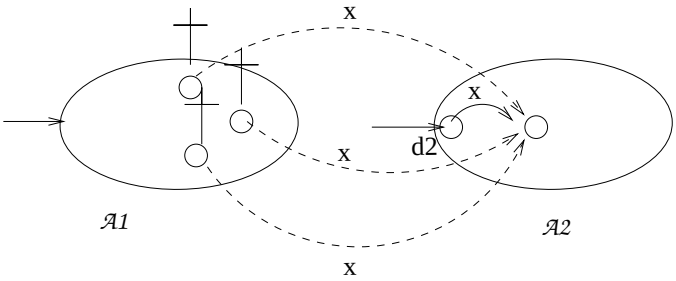


FIG. 1.9 – Produit (de concaténation)

$$\mathcal{A} =< X, Q_1 \cup Q_2, D_1, A, \delta >$$

$$\delta = \delta_1 \cup \delta_2 \cup \{(q, x, q') \in A_1 \times X \times Q_2 | \exists d_2 \in D_2, (d_2, x, q') \in \delta_2\}$$

$$A = A_2 \text{ si } D_2 \cap A_2 = \emptyset; \quad A = A_1 \cup A_2 \text{ si } D_2 \cap A_2 \neq \emptyset.$$

On vérifie que $L_{\mathcal{A}} = L_{\mathcal{A}_1} \cdot L_{\mathcal{A}_2}$. □

Définition 1.4.6 *Soit $L \subseteq X^*$. On note $L^* = \bigcup_{n \geq 0} L^n$.*

Remarquons que L^* est bien le sous-monoïde de X^* engendré par L (voir la définition 1.1.4).

Proposition 1.4.7 *$Rec(X^*)$ est fermé par l'opération *.*

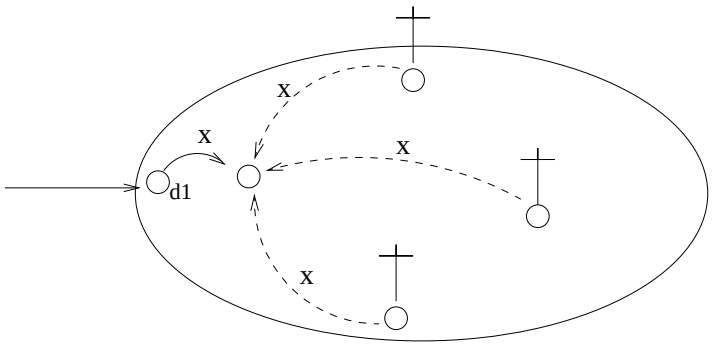


FIG. 1.10 – Opération étoile

Preuve : Soit $L = L_{\mathcal{A}_1}$:

$$\mathcal{A}_1 =< X, Q_1, D_1, A_1, \delta_1 >$$

$$\mathcal{A} =< X, Q_1, \{d_1\}, A, \delta >$$

$$\delta = \delta_1 \cup \{(q, x, q^{'}) \in A_1 \times X \times Q_1 | \exists d_1 \in D_1, (d_1, x, q^{'}) \in \delta_1\}$$

$$A = A_1 \cup D_1.$$

On vérifie que $L_{\mathcal{A}} = (L_{\mathcal{A}_1})^*$. □

Remarque 1.4.8 *Si $X \subseteq Y$ alors $Rec(X^*) \subseteq Rec(Y^*)$.*

1.5 Automate minimal

Notation : Soient $u \in X^*, L \subset X^*$

$$u^{-1}L = \{v \in X^* | uv \in L\}$$

$$Lu^{-1} = \{v \in X^* | vu \in L\}$$

On appelle "résiduel"du langage L, tout langage de la forme $u^{-1}L$.

$$\mathbb{Q}(L) = \{u^{-1}L | u \in X^*\}$$

Théorème 1.5.1 *(Myhill-Nerode) : $L \in Rec(X^*) \Leftrightarrow \mathbb{Q}(L)$ est fini.*

Preuve : 1)Supposons que $L \in Rec(X^*)$. Soit $\mathcal{A} =< X, Q, \{d\}, A, \delta >$ un automate déterministe complet reconnaissant L. Considérons les trois applications :

$$\varphi : u \mapsto u^{-1}L; \;\; \tau : u \mapsto \tilde{\delta}(d, u); \;\; \bar{\varphi} : q \mapsto L_{\mathcal{A}<q,A>}$$

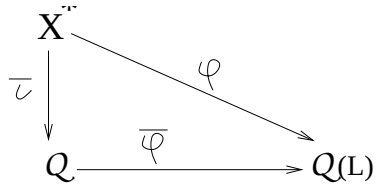


FIG. 1.11 – Diagramme 1

Le diagramme précédent est commutatif. Comme φ est surjective, $\bar{\varphi}$ l'est aussi. Donc $\mathbb{Q}(L) = \text{im}(\bar{\varphi})$ et $\text{Card}(\mathbb{Q}(L)) \leq \text{Card}(Q)$.

2)Supposons que $\mathbb{Q}(L)$ est fini.

Considérons l'automate fini déterministe, complet : $\mathcal{M} =< X, \mathbb{Q}(L), \{L\}, R, \theta >$ défini par

$$R = \{P \in \mathbb{Q}(L) | \epsilon \in P\} \;\text{ et } \theta(P, x) = x^{-1}P.$$

Pour tout $u \in X^*, \tilde{\theta}(L, u) = u^{-1}L$. Donc $L_{\mathcal{M}} = L$. □

Définition 1.5.2 *Soient $\mathcal{A} =< X, Q, \{d\}, A, \delta >$,$\mathcal{A}' =< X, Q', \{d'\}, A', \delta' >$deux automates déterministes,complets. Un* homomorphisme *de $\mathcal{A}$ dans $\mathcal{A}'$ est une application $h : Q \to Q'$ vérifiant les propriétés :*
- *(H1) $h(d) = d'$*
- *(H2) $\forall q \in Q, \forall x \in X, h(\delta(q, x) = \delta'(h(q), x)$*
- *(H3) $q \in A \Leftrightarrow h(q) \in A'$*

Propriété 1.5.3 *Si $h : \mathcal{A} \to \mathcal{A}'$ est un homorphisme, alors $L_{\mathcal{A}} = L_{\mathcal{A}'}$.*

Preuve : On prouve, à partir de $(H2)$, par récurrence sur $|u|$, que

$$\forall q \in Q, \forall u \in X^*, h(\tilde{\delta}(q,u)) = \tilde{\delta'}(h(q),u) \tag{1.3}$$

□

Théorème 1.5.4 *Soit $L \in Rec(X^*)$. Il existe un a.f déterministe complet $\mathcal{M}$ vérifiant :*
- *(MIN1) $L_{\mathcal{M}} = L$*
- *(MIN2) pour tout a.f.déterministe complet $\mathcal{A}$, si $L_{\mathcal{A}} = L$, alors il existe un homomorphisme surjectif $h : \mathcal{A} \to \mathcal{M}$.*

Cet automate $\mathcal{M}$ est unique, à isomorphisme près. On l'appelle l'automate minimal *de L.*

Preuve : **Existence**
Soit $\mathcal{M}$ l'automate défini dans la preuve du Théorème 1.5.1.
(MIN1)est vraie.
(MIN2) : montrons que $\bar{\varphi}$ est un homomorphisme d'automates :
$\bar{\varphi}(d) = L$, donc (H1) est vraie.
Soient $q \in Q, x \in X$. Alors

$$\bar{\varphi}(\delta(q,x)) = L_{\mathcal{A}<\delta(q,x),A>} = x^{-1}L_{\mathcal{A}<q,A>} = \theta(L_{\mathcal{A}<q,A>}, x)$$

donc (H2) est vraie.
$q \in A \Leftrightarrow \epsilon \in L_{\mathcal{A}<q,A>} \Leftrightarrow \epsilon \in \bar{\varphi}(q) \Leftrightarrow \bar{\varphi}(q) \in R$, donc (H3) est vraie.
Unicité
Soit $\mathcal{M} = \mathcal{M}_1$ et soit $\mathcal{M}_2$ un automate déterministe, complet, vérifiant (MIN1),(MIN2).
On considère les applications $\tau_i : u \mapsto \tilde{\delta}_i(d_i, u)$ et les homorphismes surjectifs $h_i : \mathcal{M}_{3-i} \to \mathcal{M}_i$, fournis par la propriété (MIN2). Le fait que h_i est un homorphisme entraîne que :

$$h_2 \circ \tau_1 = \tau_2 \text{ et } h_1 \circ \tau_2 = \tau_1,$$

(d'après l' équation (1.3)). On en déduit que, pour tout $u \in X^*$:

$$h_1 \circ h_2 \circ \tau_1(u) = h_1 \circ \tau_2(u) = \tau_1(u) \tag{1.4}$$

Mais τ_1 est surjective (voir la définition de $\mathcal{M} = \mathcal{M}_1$). De (1.4) on tire donc que $h_1 \circ h_2 = \mathrm{Id}_{Q_1}$.

Comme h_2 et τ_1 sont surjectives, τ_2 (qui est leur composée) l'est aussi. On peut donc utiliser le même raisonnement que ci-dessus pour montrer que $h_2 \circ h_1 = \mathrm{Id}_{Q_2}$. Les applications h_i sont donc des isomorphismes réciproques. Donc $\mathcal{M}_1 \approx \mathcal{M}_2$. □

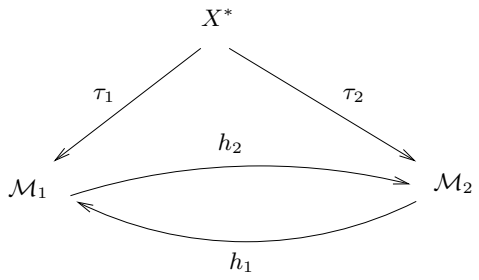


FIG. 1.12 –

Congruence d'automate :
Une relation d'équivalence $\sim$ sur Q est une *congruence* de $\mathcal{A}$ ssi,
– (C2) $\forall q, q^{'} \in Q, \forall x \in X, q \sim q' \Rightarrow \delta(q,x) \sim \delta(q',x)$
– (C3) $\sim$ sature A.
Automate quotient :
Soit $\sim$ une congruence sur l'automate d.c. $\mathcal{A} =< X, Q, \{d\}, A, \delta >$. L'automate *quotient* de $\mathcal{A}$ par la congruence $\sim$ est défini par

$$\mathcal{A}/\sim =< X, Q/\sim, \{[d]_\sim\}, A/\sim, \bar{\delta} >$$

où

$$A/\sim = \{[a]_\sim | a \in A\} \text{ et } \bar{\delta}([q]_\sim, x) = [\delta(q,x)]_\sim.$$

La projection $\Pi : Q \to Q/\sim$ est définie par :

$$\forall q \in Q, \Pi(q) = [q]_\sim.$$

Proposition 1.5.5 *1-* $\Pi : \mathcal{A} \to \mathcal{A}/\sim$ *est un homomorphisme d'automates.*
2- $L_{\mathcal{A}} = L_{\mathcal{A}/\sim}$

Lemme 1.5.6 (Factorisation des homomorphismes)
Soient $\mathcal{A} =< X, Q, \{d\}, A, \delta >$ *et* $\mathcal{B} =< X, Q', \{d'\}, A', \delta' >$ *deux automates déterministes complets. Soit* $h : \mathcal{A} \longrightarrow \mathcal{B}$ *un homorphisme d'automates surjectif. Définissons une relation d' équivalence* Ker$h$ *sur* Q *par :*

$q(\mathrm{Ker}h)r \Leftrightarrow h(q) = h(r)$.
Définissons une application $\bar{h} : Q/\mathrm{Ker}h \to Q'$ *par :* $\bar{h}([q]_{\mathrm{Ker}h}) = h(q)$. *Alors* $h = \bar{h} \circ \Pi$ *et* $\bar{h}$ *est un isomorphisme d'automates.*

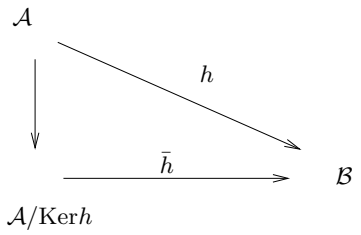


FIG. 1.13 – Factorisation de h

Définition 1.5.7 *Soit* $\mathcal{A} =< X, Q, \{d\}, A, \delta >$ *un a.d.c. La congruence de Nerode de* $\mathcal{A}$, *notée* $\equiv$, *est définie par :*

$$q \equiv r \Leftrightarrow L_{\mathcal{A}<q,A>} = L_{\mathcal{A}<r,A>}.$$

Proposition 1.5.8 *L'automate* $\mathcal{A}/\equiv$ *est isomorphe à l'automate* $\mathcal{M}$.

Preuve : Dans le diagramme 1, $\mathrm{Ker}\bar{\varphi} = \equiv$. Par le lemme de factorisation, $\mathcal{A}/\equiv$ est isomorphe à $\mathcal{M}$. □

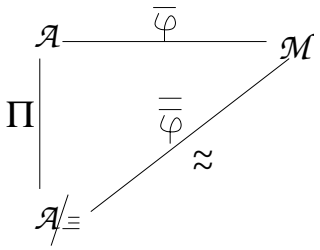


FIG. 1.14 – factorisation de $\bar{\varphi}$

Construction de l'automate minimal : Soit $\mathcal{A} =< X, Q, \{d\}, A, \delta >$ un automate fini d.c. On définit une suite d'approximations de la congruence de Nerode, $(\equiv_i)_{i\geq 0}$ par : $q \equiv_i r$ ssi

$$\forall u \in X^*, |u| \leq i \Rightarrow [\tilde{\delta}(q,u) \in A \Leftrightarrow \tilde{\delta}(r,u) \in A].$$

Lemme 1.5.9 $\forall i \in \mathbb{N}, \equiv_{i+1} \subseteq \equiv_i$.

Lemme 1.5.10 $\equiv = \bigcap_{i\geq 0} \equiv_i$.

Lemme 1.5.11 *Si* $\equiv_k = \equiv_{k+1}$ *alors* $\equiv = \equiv_k$.

Preuve :

$$q \equiv_{i+1} r \Leftrightarrow (\forall x \in X \cup \{\epsilon\}, \delta(q,x) \equiv_i \delta(r,x)).$$

Donc

$$\equiv_k = \equiv_{k+1} \Rightarrow \forall \lambda \geq 1, \equiv_k = \equiv_{k+\lambda} .$$

Ce qui entraîne que $\bigcap_{i\geq 0} \equiv_i = \equiv_k$ et donc $\equiv = \equiv_k$. □

Lemme 1.5.12 *Il existe un entier* $k \leq \mathrm{Card}(Q) - 1$, *tel que* $\equiv_k = \equiv_{k+1}$

Preuve : D'après le lemme 1.5.9 on a les inégalités :

$$1 \leq \ldots \leq \mathrm{Card}(Q/\equiv_i) \leq \mathrm{Card}(Q/\equiv_{i+1}) \leq \ldots \leq \mathrm{Card}(Q).$$

Mais la suite d'inégalités obtenues ne peuvent être strictes plus de $\mathrm{Card}(Q) - 1$ fois. □
Algorithme :

```
∼ : = ≡ _0 ;      FINI := faux;
tant que (not FINI) faire
debut
soit ≃ définie par:

 q ≃ r  ssi [∀ x ∈ X ∪{ϵ}, δ̃(q,x) ∼ δ̃(r,x)]
si ≃ = ∼  alors FINI := vrai;
fin
```

Théorème 1.5.13 *L'algorithme précédent calcule pour tout a.f.d.c $\mathcal{A}$, son équivalence de Nerode $\equiv$.*

– l'algorithme précédent est polynomial
– Il existe un algorithme en O(nlog(n)), "l'algorithme de Hopcroft", voir [BBC94, th.6.16 p.333]

Exemple 1.5.14 *Considérons l'automate fini d.c. $\mathcal{A} =< X, Q, \{d\}, A, \delta >$ où $X = \{a, b\}, Q = \{1, 2, 3, 4, 5, 6\}, d = 1, A = \{3, 4, 6\}$ et δ est donné par la table :*

δ	a	b
1	2	4
2	3	2
3	3	2
4	5	1
5	6	2
6	3	5

On obtient les approximations successives de la congruence de Nerode :

$$\begin{aligned} \equiv_0 &= \{\{1,2,5\},\{3,4,6\}\} \\ \equiv_1 &= \{\{1\},\{2,5\},\{3,6\},\{4\}\} \\ \equiv_2 &= \{\{1\},\{2,5\},\{3,6\},\{4\}\} \end{aligned}$$

Comme $\equiv_1 = \equiv_2$ on conclut que $\equiv = \{\{1\},\{2,5\},\{3,6\},\{4\}\}$. L'automate minimal équivalent à $\mathcal{A}$ est donc l'automate $\mathcal{A}/\equiv =< X, Q/\equiv, \{[1]_\equiv\}, \{[3]_\equiv, [4]_\equiv\}, \bar{\delta} >$ où $\bar{\delta}$ est donné par la table :

δ	a	b
1	25	4
25	36	25
36	36	25
4	25	1

(ici 25 dénote la classe $\{2,5\}$ et 36 dénote la classe $\{3,6\}$). Voir la figure 1.15.

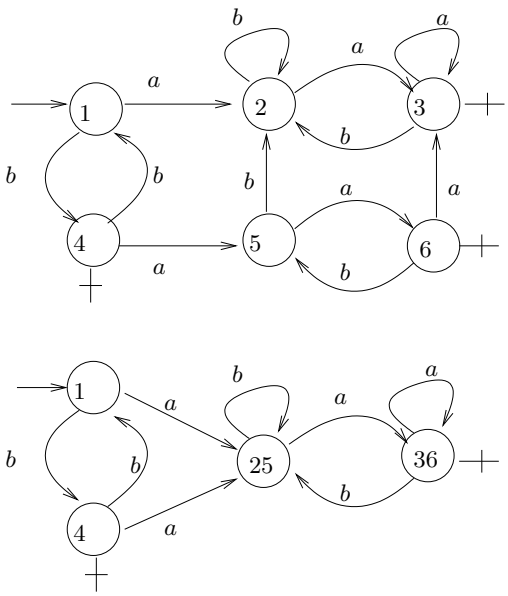


FIG. 1.15 – Minimisation

Définition 1.5.15 *Soit $\mathcal{M} = (M, \cdot, e)$ un monoïde. On appelle* congruence à droite *(resp. à gauche) sur $\mathcal{M}$, toute relation d' équivalence R sur M telle que, $\forall u, v, \beta \in M$,*

$$uRv \Rightarrow u \cdot \beta \ \ R \ \ v \cdot \beta$$

(resp. $uRv \Rightarrow \beta \cdot u \ \ R \ \ \beta \cdot v$).
R est une congruence *si et seulement si, R est à la fois une congruence à droite et à gauche.*
On appelle index *d'une congruence à droite (resp. à gauche) R, le cardinal de M/R.*

Définition 1.5.16 *On appelle congruence syntaxique à droite de L la relation $\equiv_L$ définie par :*

$$u \simeq_L v \Leftrightarrow [\forall \beta \in X^*, u\beta \in L \Leftrightarrow v\beta \in L]$$

Théorème 1.5.17 *Soit $L \in X^*$.*
(1) L est rationnel ssi $\simeq_L$ est d'index fini.

(2) $\mathcal{M} \approx < X, X^*/\simeq_L, \{[\epsilon]_{\simeq_L}\}, R, \delta >$ *où*

$$R = \{[u]_{\simeq_L} | u \in L\}; \;\; \delta([u]_{\simeq_L}, x) = [ux]_{\simeq_L}.$$

Preuve : Reprenons l'application $\varphi : X^* \to \mathbb{Q}(L)$ définie dans la preuve du théorème 1.5.1. Considérons l'automate déterministe complet (infini) $\mathcal{X} =< X, X^*, \{\epsilon\}, L, \delta_X >$ où

$$\forall u \in X^*, \forall x \in X, \delta_X(u, x) = u \cdot x.$$

φ est un homorphisme surjectif de l'automate $\mathcal{X}$ dans $\mathcal{M}$. Par le lemme de factorisation, $\mathcal{X}/\text{Ker}\varphi$ est isomorphe à $\mathcal{M}$. Mais Kerφ n'est autre que l' équivalence syntaxique à droite $\simeq_L$. Donc $\mathcal{X}/\simeq$ est isomorphe à $\mathcal{M}$, ce qui prouve le point (2).
D'après le point (2), $\text{Card}(X^*/\simeq) = \text{Card}(\mathbb{Q}(L))$. Donc L est rationnel ssi $\text{Card}(\mathbb{Q}(L)) < \infty$ (par le théorème 1.5.1) ssi $\text{Card}(X^*/\simeq) < \infty$ (par l' égalité ci-dessus). Le point (1) est prouvé. □

Définition 1.5.18 *On appelle* congruence syntaxique *de L la relation $\equiv_L$ définie par :*

$$u \equiv_L v \Leftrightarrow [\forall \alpha \in X^*, \forall \beta \in X^*, \alpha u \beta \in L \Leftrightarrow \alpha v \beta \in L]$$

Proposition 1.5.19 *Soit $L \subseteq X^*$. Les conditions suivantes sont équivalentes.*
(1) il existe un monoïde fini N, un homomorphisme $\varphi : X^* \to N$ et $P \subseteq N$, tels que $L = \varphi^{-1}(P)$
(2) il existe un monoïde fini N, un homomorphisme $\varphi : X^* \to N$ tels que $L = \varphi^{-1}(\varphi(L))$
(3) il existe une congruence d'index fini $\equiv$ sur X^* qui sature L,
(4) $\equiv_L$ est d'index fini.
Dans ce cas on dit que L est m-reconnaissable.

Preuve : (1) ⇒ (2) : selon l'égalité (1),

$$\varphi^{-1}(\varphi(L)) = \varphi^{-1}(\varphi(\varphi^{-1}(P))) = \varphi^{-1}(P) = L.$$

(2) $\Rightarrow$ (3) : considérons la congruence $\equiv = Ker\varphi$. Un "lemme de factorisation" analogue au lemme 1.5.6, mais où la structure d'automate est remplacée par la structure de monoïde, reste valide. En appliquant ce lemme 1.5.6, "version monoïdes", à $\varphi : X^* \to \text{im}\varphi \subseteq N$, on obtient que $X^*/\text{Ker}\varphi$ est isomorphe à $\text{im}\varphi$. Donc $\text{Card}(X^*/\text{Ker}\varphi) \leq \text{Card}(N) < \infty$.

(3) $\Rightarrow$ (4) : comme $\equiv$ sature L, on a l'inclusion $\equiv \subset \equiv_L$. L'application $h : X^*/\equiv \to X^*/\equiv_L$ définie par $h([u]_\equiv) = [u]_{\equiv_L}$ est bien définie (c'est en fait un homomorphisme) et elle est surjective.
Donc $\text{Card}(X^*/\equiv_L) \leq \text{Card}(X^*/\equiv) < \infty$.

(4) $\Rightarrow$ (1) : posons $N = X^*/\equiv$, $\varphi : u \mapsto [u]_{\equiv_L}$ et $P = \varphi(L)$. On a bien $L = \varphi^{-1}(P)$. □

Théorème 1.5.20 *Soit $L \subseteq X^*$. L est reconnaissable $\Leftrightarrow$ L est m-reconnaissable.*

Preuve : (1)Supposons que L est m-reconnaissable. Soit $\varphi : X^* \to N$, $P \subseteq N$ tels que $L = \varphi^{-1}(P)$.
Définissons $\mathcal{A} =< X, N, \{1_N\}, P, \delta >$ où

$$\delta(n, x) = n \cdot \varphi(x).$$

Cet automate fini (d'ailleurs déterministe et complet) reconnaît L.
(2)Supposons que L est reconnaissable. Soit $\mathcal{A} =< X, Q, \{d\}, A, \delta >$ un automate fini déterministe complet, le reconnaissant. Définissons $\mathcal{N} = (Q^Q, \otimes, I)$ où $\otimes$ dénote la composition des applications, notée "à l'envers" :

$$\forall q \in Q, \quad (f \otimes g)(q) = g(f(q)).$$

Soit $\varphi : X^* \to Q^Q$ définie par $u \mapsto (q \mapsto \tilde{\delta}(q, u))$ et soit
$P = \{f \in Q^Q | f(d) \in A\}$.
On vérifie que $L = \varphi^{-1}(P)$. □

1.6 Expressions rationnelles

Définition 1.6.1 *$Rat(X^*)$ est la plus petite partie de $\mathcal{P}(X^*)$ contenant les langages finis et stable par les opération $\cup, \cdot, *$.*

Notion d'expression rationnelle :

$$\mathcal{X} = Y \cup X \cup \{\varepsilon\}$$

où

$$Y = \{(,), \cdot, +, *\}$$

On suppose $X \cap (Y \cup \{\varepsilon\}) = \emptyset$

L'ensemble ER (X) des expressions rationnelles sur X est le plus petit langage sur $\mathcal{X}^*$ verifiant :

– (0) $\epsilon \in$ ER (X)
– (1) $\forall x \in X, x \in$ ER (X)
– (2) si $e_1, e_2 \in$ ER (X) alors $(e_1 \cdot e_2) \in$ ER (X)
– (3) si $e_1, e_2, \in$ ER (X) alors $(e_1 + e_2) \in$ ER (X)
– (4) si $e \in$ ER (X) alors $e* \in$ ER (X).

Le fait qu'il existe un *plus petit* langage sur $\mathcal{X}^*$ vérifiant ces conditions n'est pas tout à fait évident ; nous en donnons une justification, sous une forme plus générale, dans [Sén02, définition 4.2.1].
Le langage dénoté par une expression rationnelle est défini récursivement par :

$$\Pi : \text{ ER } (X) \longmapsto \mathcal{P}(X^*)$$

$$\epsilon \longmapsto \{\epsilon\}$$

$$x \in X \longmapsto \{x\}$$

$$(e_1 \cdot e_2) \longmapsto \Pi(e_1) \cdot \Pi(e_2)$$

$$(e_1 + e_2) \longmapsto \Pi(e_1) \cup \Pi(e_2)$$

$$e* \longmapsto \Pi(e*) = (\Pi(e))^*$$

Là aussi, l'existence et l'unicité d'une application Π satisfaisant ces conditions n'est pas tout à fait évidente ; elle repose sur le fait qu'un mot de

ER (X) a une décomposition *unique* sous l'une des formes (0),(1),(2),(3) ou (4). Nous en donnons une justification, sous une forme plus générale, dans [Sén02] après la définition 4.2.5.

Remarque 1.6.2 *Un langage rationnel est décrit par un infinité d'expressions rationnelles. Par exemple :*

$$(a+b)^* = (a^*b)^*a^*$$

Proposition 1.6.3 *: Soit X un alphabet fini. Alors $Rat(X^*)$ est exactement l' ensemble des langages sur X qui sont de la forme $\Pi(e)$ pour une expression rationnelle $e \in$ ER (X).*

Théorème 1.6.4 *(Kleene) : Soit X un alphabet fini,alors $Rat(X^*) = Rec(X^*)$*

Preuve : 1- $Rec(X^*)$ contient les parties finies et est fermée par $\cup, \cdot, *$. Donc $Rat(X^*) \subseteq Rec(X^*)$.
2-Soit $\mathcal{A} =< X, Q, D, A, \delta >$ un automate fini tel que $L = L_{\mathcal{A}}$.

$$\mathcal{A} =< X, Q, D, A, \delta >$$

$$Q = [1, n]$$

Pour tout calcul w de l'automate $\mathcal{A}$, on note $I(w)$ l'intérieur de ce calcul : si $w = (q_1, x_1, q_1')(q_2, x_2, q_2')...(q_i, x_i, a_i')...(q_n, x_n, q_n')$ alors $I(w) = \{q_2, \ldots, q_n\}$.
Pour tous $i, j, k \in [1, n]$, on pose alors :

$$L_{i,j}^{(k)} = \{u \in X^* | \exists w \in \delta^*, \text{ chemin de } i \text{ à } j, I(w) \subseteq [1, k]\}.$$

Les formules suivantes sont vraies :

$$\begin{aligned} L_{i,j}^{(0)} &= \{x \in X, (i, x, j) \in \delta\}(\text{si } i \neq j) \\ L_{i,j}^{(0)} &= \{x \in X, (i, x, j) \in \delta\} \cup \{\epsilon\}(\text{si } i = j) \\ L_{i,j}^{(k)} &= L_{i,j}^{(k-1)} \cup L_{i,k}^{(k-1)}(L_{k,k}^{(k-1)})^* L_{k,j}^{(k-1)}. \end{aligned}$$

On prouve ainsi par récurrence sur k :

$$\forall i, j \in [1, n], L_{i,j}^{(k)} \in Rat(X^*).$$

D'où

$$L_{\mathcal{A}} = \bigcup_{(i,j)\in D\times A} L_{\mathcal{A}<i,j>} = \bigcup_{(i,j)\in D\times A} L_{i,j}^{(n)} \in Rat(X^*).$$

□

Chapitre 2

Langages algébriques de mots

2.1 Arbres planaires ordonnés

Soit Y un alphabet et E un ensemble inclus dans Y^*. On appelle *arbre planaire ordonné étiqueté sur* E toute application $T : \mathbb{N}^* \to E$ telle que
(i) $dom(T)$ est clos inférieurement pour l'ordre préfixe : si $u \in dom(T)$ et $v \preceq u$ alors $v \in dom(T)$
(ii) $dom(T)$ est clos par "frère gauche" : $u(i+1) \in dom(T) \Rightarrow ui \in dom(T)$
On utilise alors le vocabulaire suivant :
• Nœud : élément de $dom(T)$.
• Chemin : suite de nœuds $u_1, u_2, \cdots, u_n$ telle que, u_{i+1} est un successeur de u_i pour l'ordre préfixe.
• Racine : ε
• Feuille : $u \in dom(T)$, maximal pour $\preceq$
• Nœud interne : $u \in dom(T)$, non maximal pour $\preceq$
• Hauteur : h(T) := max{l(c) | c chemin dans T} (i.e. le nombre maximum d'arcs sur une branche)

$$l(c_0, c_1, c_2, \cdots, c_n) = n$$

• Norme(T) : $\|T\|$:= Card(dom(T)).

•

$$\begin{aligned} \mathrm{r}(T) &= T(\varepsilon) \\ \mathrm{fr}(T) &= T(y_1)T(y_2)\cdots T(y_n) \end{aligned}$$

où $y_1 <_{lex} y_2 <_{lex} y_3 \cdots <_{lex} y_n$ est la liste de toutes les feuilles de T.

• Sous-arbre enraciné en $\alpha \in \mathbb{N}^*$: T/α

$$dom(T/\alpha) = \alpha^{-1}dom(T), \ \ \forall u \in \alpha^{-1}dom(T), (T/\alpha)(u) = T(\alpha u).$$

Exemple :
`une description et une figure`

2.2 Grammaires algébriques

Définition 2.2.1 *Une grammaire algébrique est un quadruple* $G =< X, V, P, S >$ *où*
- *X est un alphabet dit alphabet terminal*
- *V est un alphabet disjoint de X, dit alphabet non-terminal*
- *P est un sous-ensemble fini de $V \times (X \cup V)^*$*
- *S est un élément désigné de V, appelé l'axiome de la grammaire*

Les éléments de X sont appelés lettres terminales ou terminaux, ceux de V sont appelés lettres non-terminales ou variables ou non-terminaux, ceux de P sont des règles ou productions de la grammaire. Si $(S, m) \in P$, S est le membre gauche et $m \in (X \cup V)^*$ est le membre droit de cette règle.

Exemple :

$G_1 =< X_1, V_1, P_1, S >$ où $X_1 = \{a, b\}$, $V_1 = \{S\}$, $P_1 = \{S \to aSbS+bSaS+\varepsilon\}$

Remarques et conventions
Par abus de langage, on appelera parfois grammaire algébrique un triple $< X, V, P >$ en se permettant de ne préciser qu'après coup quel élément de

V est choisi comme axiome. Cet abus sera systématique lorsque l'alphabet non-terminal V ne contient qu'un seul élément.

Une règle (S, m) de la grammaire sera notée $S \to m$ et plusieurs règles ayant le même membre gauche sont notées en écrivant à droite du symbole $\to$ les différents membres droits séparés par le symbole $+$. Ainsi, dans l'exemple ci-dessus, $P_1 = \{(S, aSbS), (S, bSaS), (S, \varepsilon)\}$ Il est bien entendu que ni le symbole $\to$, ni le symbole $+$ n'appartiennent à $X \cup V$.

Exemples :

$$G_2 =< X_2, V_2, P_2 > \text{ où } X_2 = \{a, b\}, V_2 = \{S\}, P_2 = \{S \to aSb + aS + a\}$$

$$G_3 =< X_3, V_3, P_3, S > \text{ où } X_3 = \{a, b, c\}, V_3 = \{S, T\},$$

$$P_3 = \{S \to aSb + T, T \to Tb + b\}$$

$$G_4 =< X_4, V_4, P_4 > \text{ où } X_4 = \{a, b\}, V_4 = \{S\}, P_4 = \{S \to aSS + b\}$$

Définition 2.2.2 *Soit $G =< X, V, P >$ une grammaire algébrique. Un mot $u \in (X \cup V)^*$ se dérive directement en un mot $v \in (X \cup V)^*$ selon G si et seulement si $u = u_1\ T\ u_2$, $v = u_1\ m\ u_2$ et $(T, m) \in P$.*

On note $\to_G$ (ou $\to_P$) la relation "se dérive directement en", et lorsqu'il n'y a pas d'ambiguité sur G, on l'omet : $u \to v$. Ceci est cohérent avec la notation pour les règles puisque si $S \to m$ est une règle, S se dérive directement en m avec des contextes vides.

La relation "se dérive en", notée $\overset{*}{\to}$ est la fermeture réflexive et transitive de la relation "se dérive directement en" : $u \overset{*}{\to} v$ si et seulement si $\exists k \in \mathbb{N}$ tel que $\exists u = u_0, u_1, u_2, ..., u_k = v$ et $\forall i < k\ u_i \to u_{i+1}$.
$(u_0, u_1, ..., u_k)$ s'appelle une dérivation de u en v que l'on note $u_0 \to u_1 \to u_2 \to ... \to u_k$; k s'appelle l'ordre de la dérivation et l'on note $u \overset{k}{\to} v$.

Exemple :

$$G_5 =< X_5, V_5, P_5 > \text{ où } X_5 = \{a, b, c\}, V_5 = \{S, T, U\},$$

$$P_5 = \{S \to aSbT + cU, T \to cT + c, U \to Uc + c\}$$

$$SbS \to aSbTbS \to aSbcTbS \to aSbccbS \to aSbccbcU$$

est une dérivation d'ordre 5 de SbS en $aSbccbcU$ selon G_5.

Exercices :

- Trouver une dérivation selon G_5 de S en $aaaSbcbcbccc$
- Trouver deux dérivations distinctes selon G_5 de TU en ccc
- Ecrire tous les mots qui se dérivent de S à un ordre inférieur ou égal à 7 selon G_1, puis selon G_4.

Définition 2.2.3 *Soit $G =< X, V, P, S >$ une grammaire algébrique.*
1- Pour toute variable $T \in V$, on appelle langage engendré par G à partir du mot $u \in (X \cup V)^*$, le langage :

$$L_G(u) := \{v \in X^* | u \overset{*}{\to} v\}.$$

2- on appelle langage engendré par G, que l'on note L_G, le langage :

$$L_G = L_G(S).$$

3- on appelle langage élargi engendré par G à partir du mot $u \in (X \cup V)^*$, le langage :

$$\hat{L}_G(u) = \{v \in (X \cup V)^* | u \overset{*}{\to} v\}$$

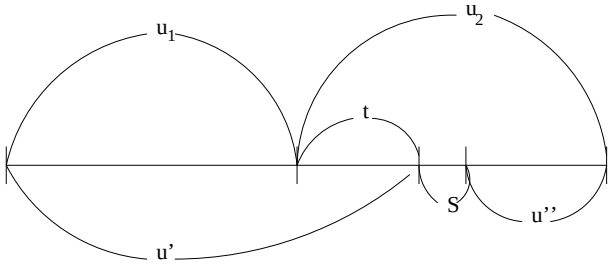
Au vu de la grammaire G_2 donnée en exemple ci-dessus, on devine que $L_{G_2} = \{a^m b^n \mid m > n \geq 0\}$.
Pour le *prouver*, nous avons besoin de l'outil suivant :

Lemme 2.2.4 *(lemme fondamental) Soient $u_1, u_2, v \in (X \cup V)^*$. Si $u_1 u_2 \overset{k}{\to} v$ alors, il existe $v_1, v_2 \in (X \cup V)^*$ tels que*

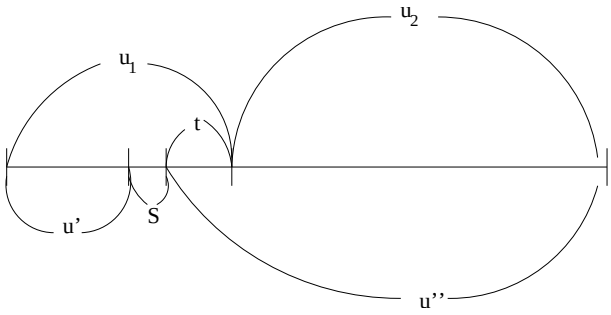
$$v = v_1 v_2, u_1 \overset{k_1}{\to} v_1, u_2 \overset{k_2}{\to} v_2 \text{ et } k_1 + k_2 = k.$$

Preuve : Montrons ce lemme par récurrence sur k.

Base 0 : $k = 0$.
Le lemme est évident.
Base 1 :$k = 1$: $u_1u_2 \to v$. Par définition $u_1u_2 = u'Su''$, $v = u'mu''$ et $S \to m \in P$. Deux cas sont à envisager :
Cas 1 :$|u'| \geq |u_1|$.
Dans ce cas on a $u' = u_1t$ et $u_2 = tSu''$
On pose $v_1 = u_1$ et $v_2 = tmu''$. On a bien $u_1 \overset{0}{\to} v_1$ et $u_2 \overset{1}{\to} v_2$, $v = v_1v_2$



et $0 + 1 = 1$.
Cas 2 :$|u'| < |u_1|$.
On a, symétriquement, $u_1 = u'St$ et $u'' = tu_2$
On pose $v_1 = u'mt$ et $v_2 = u_2$. On a bien $u_1 \overset{1}{\to} v_1$ et $u_2 \overset{0}{\to} v_2$, $v = v_1v_2$



et $1 + 0 = 1$.
Etape d'induction : Soit $k \geq 2$. Supposons le lemme vrai pour toutes les dérivations d'ordre strictement inférieur à k. Supposons $u_1u_2 \overset{k}{\to} v$.
Alors $u_1u_2 \overset{k-1}{\to} w \to v$
Par hypothèse de récurrence, $w = w_1w_2$ avec $u_1 \overset{h_1}{\to} w_1$, $u_2 \overset{h_2}{\to} w_2$,

$h_1 + h_2 = k - 1$ et $w_1 w_2 \to v$.
D'après la démonstration faite pour l'ordre 1, $v = v_1 v_2$ avec $w_1 \stackrel{\ell_1}{\to} v_1$,

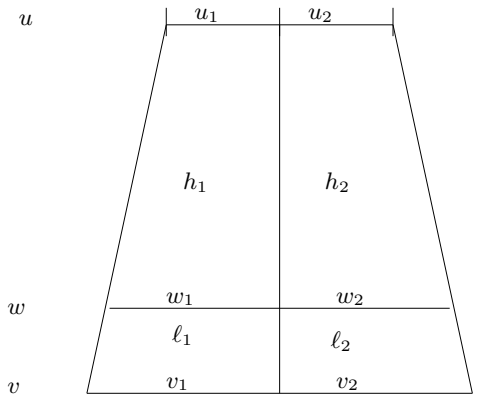


FIG. 2.1 – l'étape d'induction

$w_2 \stackrel{\ell_2}{\to} v_2$, $\ell_1 + \ell_2 = 1$.
On a donc $u_1 \stackrel{h_1+\ell_1}{\to} v_1$, $u_2 \stackrel{h_2+\ell_2}{\to} v_2$, $h_1 + \ell_1 + h_2 + \ell_2 = k$. □

Partant du lemme 2.2.4, on démontre aisément les versions plus générales qui suivent.

Lemme 2.2.5 *(lemme fondemental, version 2) Pour tous* $u_1, u_2 \in (X \cup V)^*$, $\hat{L}_G(u_1 u_2) = \hat{L}_G(u_1) \cdot \hat{L}_G(u_2)$.

Lemme 2.2.6 *(lemme fondemental, version 3) Pour tous* $p \geq 2, u_1, u_2, \ldots, u_p \in (X \cup V)^*$, *si* $u_1 u_2 \cdots u_p \stackrel{k}{\to} v$ *alors* $v = v_1 v_2 \cdots v_p$ *avec* $\forall i \in [1,p], u_i \stackrel{k_i}{\to} v_i$ *et* $\sum_{i=1}^{p} k_i = k$.

Lemme 2.2.7 *(lemme fondemental, version 4) Si* $u = w_0 S_{i_1} w_1 S_{i_2} \cdots S_{i_p} w_p \stackrel{*}{\to} v$ *avec* $\forall j \in [0,p], w_j \in X^*$, $\forall j \in [1,p], S_{i_j} \in V$, *alors* $v = w_0 m_1 w_2 \cdots m_p w_p$ *avec* $\forall j \in [1,p], m_j \in L_G(S_{i_j})$.

Remarque :
Notons, pour tout mot $u \in (X \cup V)^*$, $D(u) := \{v \in (X \cup V)^* \mid u \rightarrow_G v\}$.
D'après le lemme 2.2.4, $D(fg) = D(f) \cdot g + f \cdot D(g)$
D'où la terminologie "dérivation".

Définition 2.2.8 *Un langage est* algébrique *si et seulement si il existe une grammaire algébrique qui l'engendre.*

Exemple 2.2.9 *Montrons que* $L_{G_2}(S) = \{a^n b^m \mid n > m \geq 0\}$.
Notons $S_>$ *le langage* $\{a^n b^m \mid n > m \geq 0\}$.
1- Montrons que $S_> \subseteq L_{G_2}(S)$.
Soient $n > m \geq 0$. *En appliquant la règle* $S \rightarrow aSb$ m *fois on obtient la dérivation*

$$D1 : S \overset{m}{\rightarrow} a^m S b^m$$

puis, en appliquant la règle $S \rightarrow aS$ $n - m - 1$ *fois on obtient la dérivation*

$$D2 : a^m S b^m \overset{n-m-1}{\rightarrow} a^{n-1} S a^m$$

et en appliquant la règle $S \rightarrow a$ 1 *fois on obtient la dérivation*

$$D3 : a^{n-1} S a^m \rightarrow a^n a^m$$

En enchaînant D1,D2,D3 on obtient une dérivation

$$D : S \overset{*}{\rightarrow} a^n a^m.$$

2- Montrons que $L_{G_2}(S) \subseteq S_>$.
Prouvons par récurrence sur $k \in \mathbb{N}$ *que,*

$$\forall u \in \{a, b\}^*, S \overset{k}{\rightarrow} u \Rightarrow u \in S_>.$$

Base *:*$k = 0$. *Dans ce cas,pour tout mot terminal* u *le membre gauche de l' implication est faux, ce qui rend vraie l' implication.*
Étape d'induction *:soit* $k \geq 0$, $u \in \{a, b\}^*$ *tels que ,* $S \overset{k+1}{\rightarrow} u$. *Cette dérivation se décompose en une séquence de deux dérivations*

$$S \overset{1}{\rightarrow} w \overset{k}{\rightarrow} u.$$

Comme (S, w) *est une règle de la grammaire* G_2, *trois cas sont possibles.*
cas 1 *:* $w = a$

Dans ce cas $k = 0$ et $u = w = a$, donc $u \in S_>$.
cas 2 *: $w = aSb$*
Par le lemme fondamental (version 4), u est de la forme : $u = au'b$ avec $S \xrightarrow{k} u'$. Comme u' est facteur de u, $u' \in \{a,b\}^*$ et par hypothèse de récurrence $u' \in S_> : \exists n > m \geq 0, u' = a^n b^m$. Donc $u = a^{n+1}b^{m+1}$, ce qui montre que $u \in S_>$.
cas 3 *:$w = aS$*
Par le lemme fondamental (version 4), u est de la forme : $u = au'$ avec $S \xrightarrow{k} u'$. Comme u' est facteur de u, $u' \in \{a,b\}^*$ et par hypothèse de récurrence $u' \in S_> : \exists n > m \geq 0, u' = a^n b^m$. Donc $u = a^{n+1}b^m$, ce qui montre que $u \in S_>$.

2.3 Arbres de dérivation

Définition 2.3.1 *Soit $G =< X, V, P, \sigma >$ une grammaire algébrique.*
On appelle arbre de dérivation *de G tout arbre T étiqueté sur $X \cup V \cup \{\varepsilon\}$ vérifiant, pour tout nœud x :*
*(**AD1**)si x a exactement ℓ fils alors $T(x) \to_P T(x0)T(x1)\cdots T(x\cdot(\ell-1))$*
*(**AD2**) si $\ell \geq 2$ alors $\forall i \in [0, \ell-1]$ $T(xi) \neq \varepsilon$*
On dit que T est un arbre de dérivation terminal *si $fr(T) \in X^*$*

Exemple 2.3.2 *Soit $G =< \{a,b\}, \{S,A,B\}, P, S >$*

$$P : \left[\begin{array}{lcl} S & \to & aAB + bBa \\ A & \to & SA + ab \\ B & \to & \varepsilon \end{array} \right.$$

*La figure 2.2 représente des arbres planaires étiquetés T_1, T_2, T_3. T_1 n'est pas un arbre de dérivation car il ne vérifie pas la condition (**AD1**) : $(S, AA) \notin P$.*
T_2 est un arbre de dérivation.
*T_3 n'est pas un arbre de dérivation car il ne vérifie pas la condition (**AD2**) : $T_3(1) = \varepsilon$ alors que ce noeud a au moins un frère.*

Définition 2.3.3 *Soit $G =< X, V, P, \sigma >$ une grammaire algébrique.*
On définit la relation $\to_{g\ G}$ (dérivation gauche immédiate) par

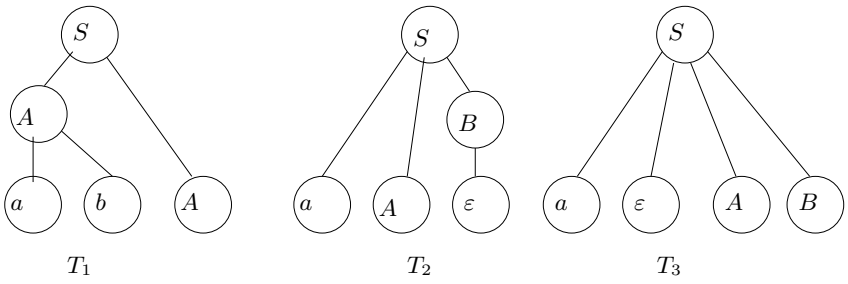


FIG. 2.2 – Quelques arbres

$\forall f \in (X \cup V)^*, \forall g \in (X \cup V)^*, f \rightarrow_{g\ G} g$ *ssi* $\exists \alpha \in X^*, \exists \beta \in (X \cup V)^*,$
$\exists (v, m) \in P, f = \alpha v \beta$ *et* $g = \alpha m \beta$
On définit alors $\rightarrow^*_{g\ G}$ *(dérivation gauche) comme la clôture réflexive et transitive de la relation* $\rightarrow_{g\ G}$.
On appelle dérivation gauche de f *en* g *une suite* $(f_1, f_2, \cdots, f_n)$ *telle que* $f_1 = f$, $f_n = g$ *et* $\forall i \in [1, n-1]$, $f_i \rightarrow_{g\ G} f_{i+1}$.
Une dérivation gauche $(f_1, \cdots, f_i, \cdots, f_n)$ *est dite terminale si* $f_n \in X^*$

Considérons d : {arbre de dérivation terminal}	$\rightarrow$	{dérivation gauche terminale}
T	$\mapsto$	la dérivation obtenue en parcourant T de gauche à droite

Exemple 2.3.4 *A partir de l'arbre de dérivation* T *de la figure 2.3 on obtient*

$$d(T) = (S, aAB, aSAB, abBaAB, abaAB, abaabB, abaab)$$

Théorème 2.3.5 *L'application* d *est une* bijection *de l'ensemble des arbres de dérivation terminaux de racine* σ *dans l'ensemble des dérivations gauches terminales partant de* σ.

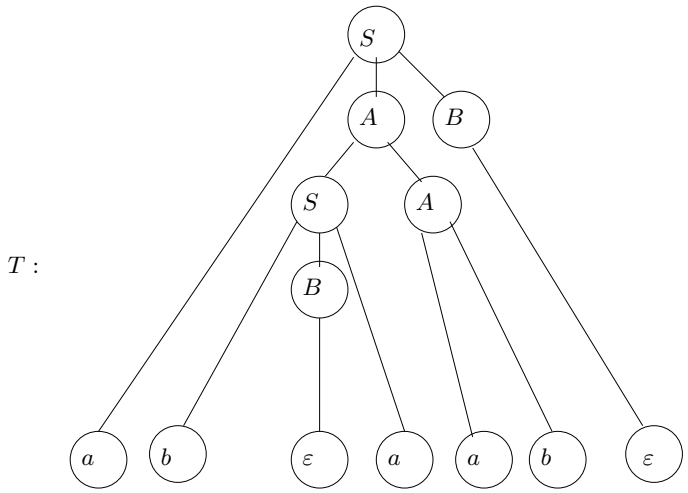


FIG. 2.3 – Un arbre de dérivation

Remarque : Soit $f \in X^*$

$$\begin{aligned} f \in L(G) &\iff \exists T, \text{ arbre de dérivation de racine } \sigma \\ &\quad \text{et de frontière } f \\ &\iff \text{il existe une dérivation gauche de } \sigma \text{ en } f \\ &\iff \text{il existe une dérivation de } \sigma \text{ en } f \end{aligned}$$

Définition 2.3.6 *Une grammaire algébrique $G =< X, V, P, \sigma >$ est dite ambiguë ssi, il existe un mot $f \in X^*$, et il existe au moins deux arbres de dérivations distincts T_1, T_2 de racine σ et de frontière f.*

G est dite non-ambiguë si elle n'est pas ambiguë.

Exemple 2.3.7 *$S \to aSb + ab$ est non-ambiguë*

Exemple 2.3.8 *$S \to aSb + bSa + SS + \varepsilon$ est ambiguë.*

2.4 Formes normales pour les grammaires algébriques

Définition 2.4.1 *Soit $G =< X, V, P, \sigma >$ une grammaire algébrique.*
- *G est dite réduite ssi :*
  (1) $\forall S \in V, L_G(S) \neq \emptyset$
  (2) $\forall S \in V, \exists u_1, u_2 \in (X \cup V)^*, \sigma \xrightarrow{*} u_1 S u_2$
- *G est dite propre ssi :*
  (3) *P ne contient pas de règle de la forme $S \to \varepsilon$*
  (4) *P ne contient pas de règle de la forme $S \to S'$*
  $(S, S' \in V)$.

Introduisons ici une notion qui sera utile pour prouver que toute grammaire est réductible à une forme réduite et propre.

Définition 2.4.2 *Etant donnés deux alphabets X, Y, on appelle* substitution *de $\mathcal{P}(X^*)$ dans $\mathcal{P}(Y^*)$ toute application*

$$\Phi : (\mathcal{P}(X^*), \cup, \cdot) \to (\mathcal{P}(Y^*), \cup, \cdot)$$

vérifiant les 3 propriétés suivantes :
(S0) $\Phi(\emptyset) = \emptyset$ *et* $\Phi(\{\varepsilon\}) = \{\varepsilon\}$
(S1) $\forall A, B \in \mathcal{P}(X^*), \Phi(A \cdot B) = \Phi(A) \cdot \Phi(B)$
(S2) *pour toute suite A_n d'éléments de $\mathcal{P}(X^*)$, $\Phi(\bigcup_{n\in\mathbb{N}} A_n) = \bigcup_{n\in\mathbb{N}} \Phi(A_n)$.*

Remarquons que Φ est entièrement déterminée par $(\Phi(\{x\}))_{x\in X}$.

Proposition 2.4.3 *Pour toute grammaire algébrique G, telle que $L_G \neq \emptyset$, il existe une grammaire algébrique réduite équivalente G_2, que l'on peut construire à partir de G.*

Preuve : Soit $G =< X, V, P, \sigma >$ telle que $L_G(\sigma) \neq \emptyset$.
Première étape : Posons $W = \{S \in V | L_G(S) \neq \emptyset\}$

Posons $W_1 = \{S \in V | \exists m \in X^*, (S,m) \in P\}$
et, pour tout $i \geq 1$

$$W_{i+1} = \{S \in V | \exists m \in (X \cup W_i)^*, (S,m) \in P\} \cup W_i$$

Fait 2.4.4 $W = \bigcup_{i \geq 1} W_i$

Il suffit de prouver par récurrence : si $S \stackrel{i}{\rightarrow} f \in X^*$ alors $S \in W_i$.

Fait 2.4.5 *si* $W_i = W_{i+1}$ *alors* $W = W_i$ *et*
il existe $i \leq |V|$ *tel que ,* $W_i = W_{i+1}$.

On peut donc calculer W.
Soit $G_1 =< X, V, P_1, \sigma >$ où $P_1 = P \cap (W \times (X \cup W)^*)$. On a :
*

$$\forall S \in W, \text{ si } (S,m) \in P_1, (S,m) \in P$$

donc $\forall S \in W$, $L_{G_1}(S) \subseteq L_G(S)$
* Si $S \stackrel{*}{\rightarrow}_G f \in X^*$, alors toutes les règles utilisées sont dans P_1, donc $S \in W$ et $L_G(S) \subseteq L_{G_1}(S)$.
* Comme $L_G \neq \emptyset$, on a $\sigma \in W$.

Soit $G_2 =< X, W, P_1, \sigma >$ on a encore $G_1 \sim G_2$ et G_2 vérifie (1).

Exemple 2.4.6 $G =< \{a,b\}, \{S_1, S_2, S_3, S_4\}, P, S_1 >$
avec

$$P = \left[\begin{array}{lcl} S_1 & \rightarrow & aS_2 + bS_2S_3 + abS_2S_1 + S_3S_4 \\ S_2 & \rightarrow & aS_3 + bS_2 + a \\ S_3 & \rightarrow & aS_3 + bS_3 \\ S_4 & \rightarrow & aS_2 + bS_1 + a \end{array} \right.$$

On obtient $W = \{S_1, S_2, S_4\}$

$$G_1 : P_1 = \left[\begin{array}{lcl} S_1 & \rightarrow & aS_2 + bS_2\emptyset + abS_2S_1 + \emptyset S_4 \\ S_2 & \rightarrow & a\emptyset + bS_2 + a \\ S_3 & \rightarrow & \emptyset \\ S_4 & \rightarrow & aS_2 + bS_1 + a \end{array} \right.$$

$G_2 =< \{a,b\}, \{S_1, S_2, S_4\}, P_1, S_1 >$

$$\left[\begin{array}{lcl} S_1 & \rightarrow & aS_2 + abS_2S_1 \\ S_2 & \rightarrow & bS_2 + a \\ S_4 & \rightarrow & aS_2 + bS_1 + a \end{array}\right.$$

Deuxième étape : Soit $G_2 =< X, V, P, \sigma >$ une grammaire vérifiant (1).
Posons $U = \{S \in V | \exists m_1, m_2 \in (X \cup V)^*, \sigma \stackrel{*}{\rightarrow} m_1 S m_2\}$.
Posons $U_0 = \{\sigma\}$ et, pour tout $i \geq 0$,

$$U_{i+1} = U_i \cup \{S \in V | \exists S' \in U_i, \exists m_1, m_2 \in (X \cup V)^*, S' \rightarrow_P m_1 S m_2\}$$

Fait 2.4.7 $U = \bigcup_{i \geq 0} U_i$

Fait 2.4.8
si $U_i = U_{i+1}$ *alors* $U = U_i$ *et*
il existe $i \leq |V| - 1$ *tel que* $U_i = U_{i+1}$

On peut donc calculer U.
Soit

$$G_3 :=< X, U, P_3, \sigma >$$

où $P_3 = P \cap U \times (X \cup U)^*$.
Remarquons que :
$*$ $\sigma \in U$
$*$ si $S \in U$ et $(S, m) \in P$, alors $m \in (X \cup U)^*$.
Donc $\forall S \in U$, $L_{G_2}(S) = L_{G_3}(S)$.
On a bien : $\forall S \in U, L_{G_3}(S) \neq \emptyset$
Donc G_3 vérifie (1) et (2) et $G \sim G_3$ □

Exemple : on obtient $U = \{S_1, S_2\}$

$$G_3 =< \{a,b\}, \{S_1, S_2\}, P_3, S_1 >$$

$$P_3 = \left[\begin{array}{lcl} S_1 & \rightarrow & aS_2 + abS_2S_1 \\ S_2 & \rightarrow & bS_2 + a \end{array}\right.$$

Proposition 2.4.9 *Pour toute grammaire algébrique G, il existe une grammaire algébrique G' vérifiant* (3) *(i.e. sans ε-règle) et telle que $L_{G'} = L_G - \{\varepsilon\}$*

Preuve : Soit $G =< X, V, P, \sigma >$.
Soit $V_\varepsilon = \{S \in V | S \ \stackrel{*}{\to}_P \ \varepsilon\}$.
Posons $V_1 = \{S \in V | S \ \stackrel{1}{\to}_P \ \varepsilon\}$ et, pour tout $i \geq 1$

$$V_{i+1} = \{S \in V | \ \exists m \in V_i^*, S \ \stackrel{1}{\to}_P \ m\} \cup V_i$$

On a encore :

- $\exists i \leq |V|, V_i = V_{i+1}$
- si $V_i = V_{i+1}, V_\varepsilon = V_i$

Soit $G' =< X, V, P', \sigma >$ où $P' = \{(T, m') \mid T \in V, m' \in (V \cup X)^+, \exists m \in (V \cup X)^+, (T, m) \in P$ et $m' \in s(m)\}$
et $s : \mathcal{P}((X \cup V)^*) \to \mathcal{P}((X \cup V)^*)$ est la substitution définie par

$$\begin{array}{lcll} \{x\} & \mapsto & \{x\} & \text{pour } x \in X \\ \{v\} & \mapsto & \{v, \varepsilon\} & \text{pour } v \in V_\varepsilon \\ \{v\} & \mapsto & \{v\} & \text{pour } v \in V - V_\varepsilon \end{array}$$

Prouvons que $\forall S \in V$, $L_{G'}(S) = L_G(S) - \{\varepsilon\}$.

Fait 2.4.10 $\stackrel{*}{\to}_{P'} \subseteq \stackrel{*}{\to}_P$,

Donc $L_{G'}(S) \subseteq L_G(S)$. De plus $P' \subseteq V \times (X \cup V)^+$, donc $L_{G'}(S) \subseteq L_G(S) - \{\varepsilon\}$

Fait 2.4.11 $\forall S \in V, \forall f \in X^+, \forall k \in \mathbb{N}, S \ \stackrel{k}{\to}_P \ f \ \Rightarrow \ S \ \stackrel{*}{\to}_{P'} \ f$

On prouve ce fait par récurrence sur k.
$k = 1$: facile
$k + 1$:

$$S \to \overbrace{u_0 S_1 u_1 S_2 \cdots S_p u_p}^{m} \ \stackrel{k}{\to} \ f$$

Donc

$$\left[\begin{array}{l} f = u_0 f_1 u_1 f_2 \cdots f_p u_p \\ S_i \stackrel{k_i}{\rightarrow}_P f_i \\ \text{avec } k_i \leq k \end{array}\right.$$

Notons $I = \{i \in [1,p], f_i = \varepsilon\}, J = \{i \in [1,p], f_i \neq \varepsilon\}$
et posons $m' = u_0 T_1 u_1 T_2 \cdots T_p u_p$ où

$$T_i = \left\{\begin{array}{l} \varepsilon \text{ si } i \in I \\ S_i \text{ si } i \in J \end{array}\right.$$

Par hypothèse de récurrence : $\forall i \in J,\ T_i = S_i \stackrel{*}{\rightarrow}_{P'} f_i$
D'autre part, $\forall i \in I,\ T_i \stackrel{0}{\rightarrow}_{P'} f_i$. Donc

$$S \rightarrow_{P'} m' \stackrel{*}{\rightarrow}_{P'} f$$

□

Proposition 2.4.12 *Pour toute grammaire algébrique G vérifiant (3), il existe une grammaire algébrique G' équivalente vérifiant les conditions* (3)(4).

Preuve : Soit $G =< X, V, P, \sigma >$, une grammaire vérifiant (3) (i.e. sans ε-règle). Définissons, pour toute variable $S \in V$ l'ensemble de variables :

$$C(S) := \{T \in V \mid S \stackrel{*}{\rightarrow}_P T\}$$

Nous définissons une suite $(C_i(S))_{i \in \mathbb{N}}$ d'ensembles de variables par :

$$C_0(S) := \{S\};\ \ C_{i+1}(S) := \{T \in V \mid \exists T' \in C_i(S), T' \rightarrow_P T\}.$$

Fait 2.4.13 *Pour toute variable $S \in V$, $C(S) = \bigcup_{i \geq 0} C_i(S)$.*

Fait 2.4.14 *Pour toute variable $S \in V$,*
si $C_i(S) = C_{i+1}(S)$ alors $C(S) = C_i(S)$
il existe un entier $i \leq |V| - 1$ tel que $C_i(S) = C_{i+1}(S)$.

En calculant les ensembles $C_i(S)$ (pour $S \in V, i \in [0, |V| - 1]$), on obtient donc les ensembles $C(S)$, pour toutes les variables $S \in V$. Posons alors

$$G' := \langle X, V, P', \sigma \rangle$$

où $P' := \{S \to m \mid S \in V, \exists T \in C(S), T \to_P m, m \notin V\}$.
Il est clair que G' vérifie les conditions (3)(4) et on peut vérifier que $\mathrm{L}_G = \mathrm{L}_{G'}$. □

Exemple 2.4.15

$$X = \{a, b, c\}\ V = \{S_0, S_1, S_2, S_3, S_4, S_5\}\ G =< X, V, P >$$

$$P = \left[\begin{array}{lcl} S_0 & \to & S_1S_2 + S_3S_4 + S_5 \\ S_1 & \to & S_1S_1 + S_4 \\ S_2 & \to & aS_2 + S_3 \\ S_3 & \to & S_2 + S_4 + S_5aS_3 \\ S_4 & \to & cS_4 + \varepsilon \\ S_5 & \to & S_4 + b \end{array}\right.$$

Cherchons à supprimer les règles décroissantes (au sens large) dans cette grammaire. On applique la transformation de la preuve de la Proposition 2.4.9.

$$V_1 = \{S_4\}\ V_2 = \{S_4, S_1, S_5\}\ V_3 = \{S_0, S_4, S_1, S_5, S_3\}\ V_4 = V$$

$$D'où\ P' = \left[\begin{array}{lcl} S_0 & \to & S_1S_2 + S_1 + S_2 + S_3S_4 + S_3 + S_4 + S_5 \\ S_1 & \to & S_1S_1 + S_1 + S_4 \\ S_2 & \to & aS_2 + a + S_3 \\ S_3 & \to & S_2 + S_4 + S_5aS_3 + aS_3 + S_5a + a \\ S_4 & \to & cS_4 + c \\ S_5 & \to & S_4 + b \end{array}\right.$$

On applique maintenant la transformation de la preuve de la Proposition 2.4.12.

$$C(S_0) = V, C(S_1) = \{S_1, S_4\}, C(S_2) = \{S_2, S_3, S_4\}, C(S_3) = \{S_3, S_2, S_4\}, C(S_4) = \{S_4\},$$

$C(S_5) = \{S_5, S_4\}$.

$$D'où\ P'' = \left[\begin{array}{lcl} S_0 & \to & S_1S_2 + S_3S_4 + S_1S_1 + aS_2 + a + S_5aS_3 + aS_3 + S_5a + cS_4 + c + b \\ S_1 & \to & S_1S_1 + cS_4 + c \\ S_2 & \to & aS_2 + a + S_5aS_3 + aS_3 + S_5a + cS_4 + c \\ S_3 & \to & S_5aS_3 + aS_3 + S_5a + a + aS_2 + cS_4 + c \\ S_4 & \to & cS_4 + c \\ S_5 & \to & cS_4 + b + c \end{array}\right.$$

Proposition 2.4.16 *Soit $G =< X, V, P, S >$ une grammaire algébrique.*
1- On peut tester si $L_G = \emptyset$.
2- On peut tester si $L_G = \{\varepsilon\}$.
3- Si $L_G \neq \emptyset$ et $L_G \neq \{\varepsilon\}$, alors on peut transformer (de façon effective) G en $G' =< X, V', P', S' >$ telle que $L_{G'} = L_G - \{\varepsilon\}$ et G' est propre et réduite.

Preuve : 1- On calcule l'ensemble W (preuve de la prop. 2.4.3) ; si $S \in W$ alors $L_G = \emptyset$, sinon $L_G \neq \emptyset$.
3- Supposons que $L_G \neq \emptyset$. On transforme G en G_1, qui est sans $S \to \varepsilon$, selon la méthode de la proposition 2.4.9 ; on obtient $L_{G_1} = L_G - \{\varepsilon\}$. On peut tester (point 1) si $L_{G_1} = \emptyset$.
3.1- Si $L_{G_1} = \emptyset$ alors $L_G = \{\varepsilon\}$.
3.2- Supposons que $L_{G_1} \neq \emptyset$.
G_1 est transformée en G_2, qui est propre, selon la méthode de la proposition 2.4.12 ; (on vérifie que G_2 est sans $S \to \varepsilon$) ;
G_2 est transformée en G_3, qui est réduite, selon la méthode de la proposition 2.4.3 ; de plus G_3 reste propre, car les transformations utilisées ne font que *supprimer* des règles. Finalement $L_{G_3} = L_{G_2} = L_{G_1} = L_G - \{\varepsilon\}$.
2- L' algorithme donné pour le point (3) ci-dessus résout aussi le problème $L_G = \{\varepsilon\}$? (voir point 3.1). □

Proposition 2.4.17 *Soit G une grammaire algébrique et w un mot. On peut décider si $w \in L_G$.*

Preuve : Soit $G =< X, V, P, S >$ et $w \in X^*$. Si $w = \varepsilon$ il suffit de construire V_ε et tester si $S \in V_\varepsilon$.

Supposons que $w \neq \varepsilon$. On construit une grammaire propre $G' =< X, V', P', S' >$ telle que $L_{G'} = L_G - \{\varepsilon\}$. On a alors : $S' \overset{*}{\rightarrow}_{P'} w$ ssi $S' \overset{k}{\rightarrow}_{P'} w$ avec $k \leq 2|w| - 1$
En effet : posons, pour tout $f \in (X \cup V)^*$, $\|f\| = |f|_V + 2|f|_X$. $\rightarrow_{P'}$ est strictement croissante pour $\| * \|$ et $\|w\| - \|S'\| = 2|w| - 1$.
On peut construire toutes les dérivations de longueur $\leq 2|w| - 1$ issues de S'. Si l'une d'entre elles aboutit à w , alors $w \in L_G$, sinon, $w \notin L_G$. □

Remarque 2.4.18 *Cet algorithme fonctionne en temps exponentiel par rapport à $|w|$. Il existe en fait des algorithmes fonctionnant en temps $O(|w|^3)$: l'algorithme de Earley et l'algorithme de Coke-Kasami-Younger sont les plus connus. On ne connaît pas d'algorithme fonctionnant en temps $O(|w|^2)$ et qui s'appliquerait à toute grammaire algébrique. C'est pourquoi on a défini diverses sous-classes de grammaires et langages, notamment les grammaires LL(k) et LR(k), pour lesquelles on connaît des algorithmes testant l'appartenance en temps $O(|w|)$. Ce sujet sera approfondi dans l'UE intitulée "analyse syntaxique" (l'analyse syntaxique consiste, étant donnés une grammaire G et un mot w , à tester si $w \in L_G$ et, lorsque la réponse est positive, à produire un arbre de dérivation pour G de frontière w).*

Définition 2.4.19 *Soit $G =< X, V, P, S >$ une grammaire algébrique. La grammaire G est en* forme normale de Chomsky *ssi , toute règle de P est de l'une des trois formes suivantes :*
(1) $T \rightarrow T_1T_2$ *où* $T \in V, T_1, T_2 \in V - \{S\}$
(2) $T \rightarrow x$ *où* $T \in V, x \in X$
(3) $S \rightarrow \varepsilon$.

Proposition 2.4.20 *Pour toute grammaire algébrique $G =< X, V, P, S >$, il existe une grammaire algébrique $G' =< X, V', P', S' >$,telle que $L_G = L_{G'}$ et G' est en forme normale de Chomsky.*

Preuve : Soit $G =< X, V, P, S >$. Construisons la grammaire demandée G' en plusieurs étapes.
Étape 1 : On applique la proposition 2.4.16 et on obtient une grammaire $G' =< X, V', P', S >$ qui est propre, réduite, et qui engendre $L_G - \{\varepsilon\}$. On

pose $V_1 := V' \cup \{\sigma\}$ et on pose $P_1 := P \cup \{(\sigma, m) \mid (S, m) \in P'\}$ (si $\varepsilon \notin L_G$), $P_1 := P \cup \{(\sigma, m) \mid (S, m) \in P'\} \cup \{(\sigma, \varepsilon)\}$ (si $\varepsilon \in L_G$). On obtient ainsi $G_1 =< X, V_1, P_1, \sigma >$ équivalente à G et dont les règles sont de l'une des trois formes suivantes :
(1) $T \to A_1 A_2 \cdots A_p$ où $p \geq 2, T \in V_1, A_1, \ldots, A_p \in X \cup (V_1 - \{\sigma\})$
(2) $T \to x$ où $T \in V_1, x \in X$
(3) $\sigma \to \varepsilon$.
Étape 2 : On considère un alphabet W_X en bijection avec X et disjoint de $X \cup V_1$: $W_X = \{S_x \mid x \in X\}$. On construit P_2 en partant de l' ensemble de règles $\{S_x \to x \mid x \in X\}$ et en ajoutant, pour chaque règle $S \to u_0 S_1 u_1 S_2 u_2 \ldots S_n u_n$ de P_1 ($S_j \in V_1, u_j \in X^*$) la règle $S \to U_0 S_1 U_1 S_2 U_2 \ldots S_n U_n$ dans P_2, où U_j est la "traduction de u_j sur l' alphabet W_X". On obtient $G_2 =< X, V_1 \cup W_X, P_2, \sigma >$ dont les règles sont de l'une des trois formes suivantes :
(1) $T \to T_1 T_2 \cdots T_p$ où $p \geq 2, T, T_1, T_2, \ldots, T_p \in (V_1 - \{\sigma\}) \cup W_X$
(2) $T \to x$ où $T \in V_1, x \in X$
(3) $\sigma \to \varepsilon$.
Étape 3 : Notons $r_1, \ldots, r_n$ l'ensemble des règles de la forme (1) avec $p \geq 3$

$$r_i : T_i \to T_{i,1} T_{i,2} \cdots T_{i,p_i}$$

On pose $V_3 := V_1 \cup W_X \cup \{U_{i,j} \mid 1 \leq i \leq n, 2 \leq j \leq p_i - 1\}$ et on remplace la règle r_i par l'ensemble des $p_i - 1$ règles

$$T_i \to T_{i,1} U_{i,2}, \quad \ldots \quad , U_{i,2} \to T_{i,2} U_{i,3}, \quad \ldots \quad U_{i,p_i-1} \to T_{i,p_i-1} T_{i,p_i}$$

On obtient ainsi $G_3 =< X, V_3, P_3, S >$ qui est équivalente à G et dont toutes les règles sont de l'une des formes (1),(2) ou (3). □

Exemple 2.4.21 *Soit G d'axiome S, d'alphabet terminal $X = \{a, b\}$, dont l'ensemble des règles est :*

$$S \to SaT; \quad T \to Tbb + bS$$

La première étape produit

$$\sigma \to SaT; \quad S \to SaT; \quad T \to Tbb + bS$$

La deuxième étape produit

$$\sigma \to SAT; \quad S \to SAT; \quad T \to TBB + BS; \quad A \to a; \quad B \to b$$

La troisième étape produit

$$\sigma \to SU_1; \quad S \to SU_1; \quad U_1 \to AT; \quad T \to TU_2 + BS; \quad U_2 \to BB; \quad A \to a; \quad B \to b.$$

2.5 Lemme d'itération

Théorème 2.5.1 *Soit L un langage algébrique. Il existe un entier k tel que, pour tout $w \in L$, si $|w| \geq k$, alors w se factorise en $w = \alpha u \beta v \gamma$ avec*
(1) $\forall n \in \mathbb{N}, \alpha u^n \beta v^n \gamma \in L$
(2) $|uv| \geq 1$
(3) $|u\beta v| \leq k - 1$.

Fixons un langage algébrique L et une grammaire algébrique propre $G = \langle X, V, P, \sigma \rangle$ qui engendre $L - \{\varepsilon\}$. Notons

$$\ell := \max\{|m| \mid (S, m) \in P\}$$

et posons

$$k := \ell^{|V|+1} + \ell.$$

Montrons que le théorème est satisfait par cette valeur de k.

Lemme 2.5.2 *Soit T un arbre de dérivation de G de hauteur inférieure ou égale à h. Alors $|fr(T)| \leq \ell^h$.*

Preuve : Se prouve par récurrence sur h. □
Prouvons maintenant le théorème 2.5.1.
Preuve : Soit $w \in L$ tel que $|w| \geq k$. comme $k \geq \frac{k}{\ell} = \ell^{|V|} + 1$ on a aussi $|w| \geq \ell^{|V|} + 1$. Comme $w \neq \varepsilon$, $w \in L_G$. Soit T un arbre de dérivation (pour G) tel que $\mathrm{r}(T) = \sigma$ et $|\mathrm{fr}(T)| \geq k$. D'après le lemme 2.5.2, il existe une branche b de T de longueur au moins $|V| + 1$:

$$b = x_0, x_1, \ldots, x_p$$

avec $p \geq |V| + 1$. Tous les noeuds $x_0, x_1, \ldots, x_{p-1}$ ont une étiquette dans V. Donc il existe un couple $(i, j) \in [0, p-1] \times [0, p-1]$ tel que

$$i < j \text{ et } T(x_i) = T(x_j).$$

Choisissons une branche b et un couple (i, j), parmi ceux vérifiant la propriété ci-dessus, tels que $|\mathrm{fr}(T/x_i)|$ soit minimale. Notons $S = T(x_i) = T(x_j)$. Une des dérivations associées à l'arbre T (voir la figure 2.4) se décompose sous la forme :

$$\sigma \overset{*}{\rightarrow} \alpha S \gamma; \quad S \overset{*}{\rightarrow} uSv; \quad S \overset{*}{\rightarrow} \beta. \tag{2.1}$$

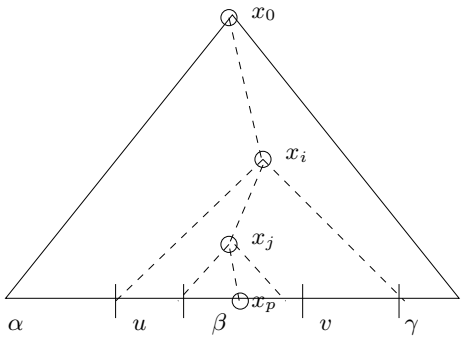


FIG. 2.4 – La décomposition de T.

Comme G est propre on a aussi :

$$|uv| \geq 1 \tag{2.2}$$

La décomposition $w := \alpha \cdot u \cdot \beta \cdot v \cdot \gamma$ fournie par T, b, i, j possède donc les propriétés (1)(2) du théorème. De plus, si $|\mathrm{fr}(T/x_i)| \geq k$, alors, il existe un noeud y fils direct de x_i tel que $|\mathrm{fr}(T/y)| \geq k/\ell$. On pourrait trouver dans le sous-arbre T/y un autre couple de noeuds $z \prec z'$ distincts et de même étiquette dans V, ce qui contredirait la minimalité de $|\mathrm{fr}(T/x_i)|$. Donc on a aussi

$$|u\beta v| \leq k - 1 \tag{2.3}$$

□

Application : $L = \{a^n b^n c^n \mid n \geq 1\}$ n'est pas algébrique

2.6 Propriétés de clôture

Rappelons qu'une notion de *substitution* est donnée par la définition 3.1.7 Une substitution Φ de $\mathcal{P}(X^*)$ dans $\mathcal{P}(X^*)$ est dite *algébrique* ssi,

$$\forall x \in X, \Phi(\{x\}) \in Alg(X^*).$$

Proposition 2.6.1 *La famille des langages algébriques est fermée par substitution algébrique, de façon constructive.*

construction :
Soit $X = \{x_1, \ldots, x_n\}$ un alphabet terminal. Soit L engendré par $G = \langle X, V, P, S_0\rangle$ et soit une substitution $\Phi : \mathcal{P}(X^*) \to \mathcal{P}(X^*)$ telle que, pour tout $i \in [1, n]$,

$$\Phi(\{x_i\}) \text{ est engendré par } G_i = \langle X, V_i, P_i, S_i\rangle.$$

On suppose tous les ensembles V, V_i disjoints 2 à 2. Posons

$$W := V \cup (\bigcup_{i=1}^{n} V_i) \quad Q := \bigcup_{i=1}^{n} P_i.$$

Considérons l'homomorphisme de monoides libres

$$\begin{array}{rcl} \Psi : (X \cup V)^* & \to & (W \cup V)^* \\ v \in V & \mapsto & v \\ x_i \in X & \mapsto & S_i \end{array}$$

On note $\Psi(P) = \{S \to \Psi(m) \mid (S, m) \in P\}$
On peut vérifier que $\Phi(L)$ est engendré par la grammaire

$$G_\Phi =< X, W, \Psi(P) \cup Q, S_0 >$$

Proposition 2.6.2 *La famille des langages algébriques est fermée de façon constructive par*
1- homomorphisme de monoïdes libres
2- union
3- produit
4- étoile

Preuve : On se ramène à la proposition 2.6.1 dans chaque cas. □

Proposition 2.6.3 *La famille des langages algébriques est fermée de façon constructive par homomorphisme alphabétique inverse.*

Preuve : Soit

$$\varphi : X^* \to X^*$$

un homomorphisme *alphabétique* i.e. tel que, pour tout $x \in X, |\varphi(x)| \leq 1$. On pose :

$$X_1 := \{x \in X \mid \varphi(x) \in X\}, \quad X_0 = \{x \in X \mid \varphi(x) = \varepsilon\}.$$

On remarque que, pour tout $x \in X$,

$$\varphi^{-1}(x) = X_0^*(\varphi^{-1}(x) \cap X_1)X_0^*$$

et que, $\varphi^{-1} : \mathcal{P}(X^*) \to \mathcal{P}(X^*)$ vérifie tous les axiomes des substitutions (voir la définition 3.1.7), à l'exception de la partie de l'axiome (S0) concernant ε, i.e. $\varphi^{-1}(\{\varepsilon\}) = X_0^*$ qui, en général, n'est pas égal à $\{\varepsilon\}$. On définit donc une substitution $\Phi : \mathcal{P}(X^*) \to \mathcal{P}(X^*)$ par :

$$\forall x \in X, \Phi(\{x\}) := \varphi^{-1}(x).$$

On a alors, pour tout langage $L \subseteq X^+$:

$$\varphi^{-1}(L) = \Phi(L), \quad \varphi^{-1}(L \cup \{\varepsilon\}) = \Phi(L) \cup X_0^*.$$

Comme Φ est une substitution algébrique, par la proposition 2.6.1, pour tout langage algébrique L, $\Phi(L)$ est algébrique et par les formules ci-dessus, $\varphi^{-1}(L)$ est algébrique. □

Proposition 2.6.4 *La famille des langages algébriques est fermée de façon constructive par intersection avec les langages rationnels.*

Preuve : Soit $G =< X, V, P, S >$ une grammaire algébrique qui engendre L et soit $\mathcal{A} =< X, Q, \{d\}, Q_+, \delta >$ un automate fini déterministe qui reconnaît R.
On construit $G_{\mathcal{A}} =< X, W, P', S' >$ où $W = \{[q, S, q'] \mid q, q' \in Q, \ S \in V\} \cup \{S'\}$ et P est constitué des règles

$$S' \to \sum_{q \in Q_+} [d, S, q] \tag{2.4}$$

$$[q, T, q'] \to u_0[q_1, T_1, q_1']u_1 \cdots [q_i, T_i, q_i']u_i \cdots [q_n, T_n, q_n']u_n \tag{2.5}$$

pour toute règle $T \to u_0T_1u_1 \cdots T_iu_i \cdots T_nu_n \in P$, avec $u_0, u_1 \cdots, u_i, \cdots, u_n \in X^*, T \in V, T_i \in V$ et pour tous états q_i, q_i' vérifiant

$q \odot u_0 = q_1, q_1' \odot u_1 = q_2, \ldots, q_i' \odot u_i = q_{i+1}, \ldots, q_{n-1}' \odot u_{n-1} = q_n$ et $q_n' \odot u_n = q'$.

Les règles de la forme (2.5) assurent que, pour tout $T \in V, q \in Q, q' \in Q$,

$$\mathrm{L}_{G_{\mathcal{A}}}([q, T, q']) = \mathrm{L}_G(T) \cap \{u \in X^* \mid q \odot u = q'\}.$$

Ces égalités, combinées aux règles de la forme (2.4) entraînent que

$$\mathrm{L}_{G_{\mathcal{A}}}(S') = \mathrm{L}_G(S) \cap \mathrm{L}_{\mathcal{A}}.$$

□

Proposition 2.6.5 *La famille des langages algébriques est fermée de (façon constructive) par homomorphisme inverse.*

Preuve : Soit $\varphi : X^* \to Y^*$. On supposera que $X \cap Y = \emptyset$.
On pose

$$K := (\{x\varphi(x)\}_{x \in X})^*$$

$$\begin{array}{rcl} \alpha : (X \cup Y)^* & \to & X^* \\ x & \to & x \\ y & \to & \varepsilon \end{array}$$

$$\begin{array}{rcl} \beta : (X \cup Y)^* & \to & Y^* \\ x & \to & \varepsilon \\ y & \to & y \end{array}$$

On vérifie que

$$\{(f, \varphi(f)) \mid f \in X^*\} = \{(\alpha(k), \beta(k) \mid k \in K\}$$

ce qui entraîne que

$$\forall g \in Y^*, \varphi^{-1}(g) = \alpha(\beta^{-1}(g) \cap K).$$

Schéma :

$$\begin{array}{ccc} & (X \cup Y)^* & \\ \overset{\alpha}{\swarrow} & \overset{\cup}{K} & \overset{\beta}{\searrow} \\ X^* & \to_{\varphi} & Y^* \end{array}$$

Comme α, β sont alphabétiques et K est rationnel,

$$\alpha(\beta^{-1}(L) \cap K) \text{ est algébrique.}$$

□

Exemples :

$$\varphi : \begin{cases} x & \to & ab \\ y & \to & a \\ z & \to & b \end{cases}$$

$$G : [\ S \to aSS + b$$

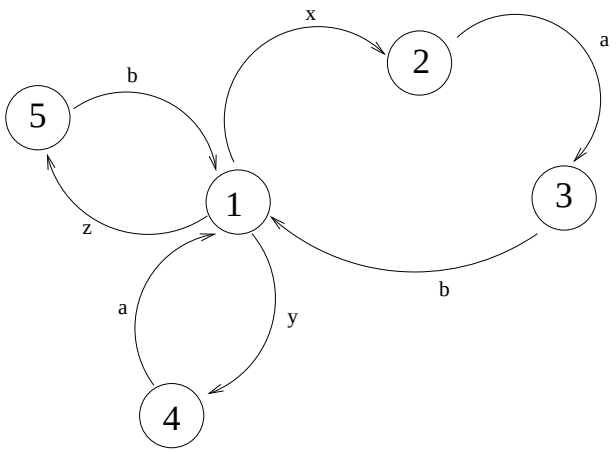
$$K = (xab, ya, zb)^*$$

$$\overset{\alpha}{\swarrow} \qquad\qquad \overset{\beta}{\searrow}$$

$$\{x, y, z\} \qquad\qquad \{a, b\}$$

$$G_{\beta^{-1}} : \begin{cases} S & \to & ASS + B \\ B & \to & EbE \\ A & \to & EaE \\ E & \to & xE + yE + zE + \varepsilon \end{cases}$$

Automate $\mathcal{A}$:



$$
\begin{array}{rcl}
G_{\beta^{-1}\cap K}: \ S' & \to & [1,S,1] \\
[1,S,1] & \to & [1,A,1][1,S,1][1,S,1] + [1,A,3][3,S,1][1,S,1] + [1,B,1] \\
[1,B,1] & \to & [1,E,5]b[1,E,1] \\
[1,A,1] & \to & [1,E,4]a[1,E,1] \\
[1,A,3] & \to & [1,E,2]a[3,E,3] \\
[1,E,5] & \to & z \\
[1,E,4] & \to & y \\
[1,E,1] & \to & \varepsilon \\
[1,E,2] & \to & x \\
[3,E,3] & \to & \varepsilon \\
[3,S,1] & \to & [3,B,1] \\
[3,B,1] & \to & [3,E,3]b[1,E,1]
\end{array}
$$

D'où :

$$
\left\{
\begin{array}{rcl}
[1,S,1] & \to & [1,A,1][1,S,1][1,S,1] + [1,A,3]b[1,S,1] + [1,B,1] \\
[1,B,1] & \to & zb \\
[1,A,1] & \to & ya \\
[1,A,3] & \to & xa
\end{array}
\right.
$$

$$
G_{\beta^{-1}\cap K}: \Big[\quad [1,S,1] \quad \to \quad ya[1,S,1][1,S,1] + xab[1,S,1] + zb
$$

$$
G_{\varphi^{-1}}: \Big[\quad [1,S,1] \quad \to \quad y[1,S,1][1,S,1] + x[1,S,1] + z
$$

2.7 Automates à pile

Nous introduisons ici une notion d'automate plus générale que la notion d'automate fini : ce type d'automate a une mémoire qui est une *pile* (au sens que possède ce terme en algorithmique).

Définition 2.7.1 *Un automate à pile est un 6-uplet* $\mathcal{A} = \langle X, Z, Q, z_0, q_0, \lambda \rangle$ *où*
– *X est un ensemble fini, que l' on appelle "l'alphabet d' entrée"*
– *Z est un ensemble fini, que l' on appelle "l'alphabet de pile"*

– *Q est un ensemble fini, que l' on appelle "l'ensemble des états"*
– *z_0 est un symbole de Z, dit "symbole initial" de pile*
– *q_0 est un élément de Q, dit "état initial" de l' automate*
– *$\lambda \subseteq (X \cup \{\varepsilon\}) \times Q \times Z \times Q \times Z^*$ est "l'ensemble des transitions".*

On note volontiers $(y, q, z) \rightarrow (q', w)$ une transition (i.e. un élément de λ). On appelle *configuration* de l' automate $\mathcal{A}$, tout triplet (f, q, h) où $f \in X^*, q \in Q, h \in Z^*$ (voir la figure 2.5). On définit la relation binaire $\vdash_{\mathcal{A}}$ sur l'ensemble

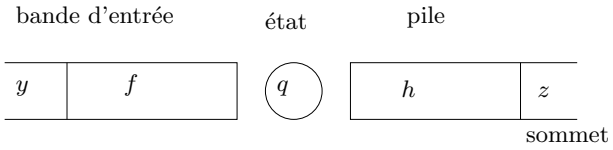


FIG. 2.5 – Une configuration de $\mathcal{A}$.

des configurations de $\mathcal{A}$ par :

$$c \vdash_{\mathcal{A}} c'$$

ssi il existe $y \in X \cup \{\varepsilon\}, f \in X^*, q \in Q, q' \in Q, h \in Z^*, z \in Z, g' \in Z^*$ tels que

$$c = (yf, q, hz), \quad c' = (f, q', hg') \quad \text{et } (y, q, z, q, g') \in \lambda.$$

(Voir la figure 2.6). Un *calcul* de l'automate $\mathcal{A}$ est une suite $(c_0, c_1, \ldots, c_n)$

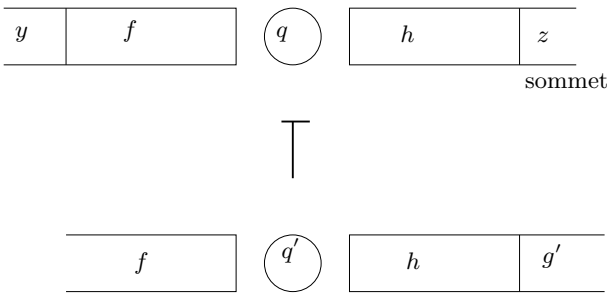


FIG. 2.6 – Un mouvement de $\mathcal{A}$.

de configurations de $\mathcal{A}$ telle que

$$c_0 \vdash_{\mathcal{A}} c_1 \ldots \vdash_{\mathcal{A}} c_i \vdash_{\mathcal{A}} c_{i+1} \ldots \vdash_{\mathcal{A}} c_n.$$

Définition 2.7.2 *Soit $\mathcal{A}$ un automate à pile sur l'alphabet d'entrée X. On appelle* langage reconnu *par $\mathcal{A}$, par pile vide, le langage*

$$\mathrm{L}_N(\mathcal{A}) := \{f \in X^* \mid \exists q' \in Q, (f, q_0, z_0) \vdash_{\mathcal{A}}^* (\varepsilon, q', \varepsilon)\}.$$

Théorème 2.7.3 *Un langage L est algébrique ssi, il existe un automate à pile $\mathcal{A}$ tel que $L = \mathrm{L}_N(\mathcal{A})$.*

Preuve : 1- Soit $L = L_G$ où $G = \langle X, V, P, S\rangle$. On construit $\mathcal{A} = \langle X, X \cup V, Q, S, q, \lambda\rangle$ où $Q := \{q\}$ et λ est l'ensemble des transitions suivantes
$(T1)$ $x, q, x{\rightarrow}q, \varepsilon$ pour toute lettre $x \in X$
$(T2)$ $\varepsilon, q, v{\rightarrow}q, \tilde{m}$ pour toute règle $(v, m) \in P$.
On peut montrer, par induction sur la longueur des dérivations de G, que :

$$L_G \subseteq \mathrm{L}_N(\mathcal{A})$$

et par induction sur la longueur des calculs de $\mathcal{A}$, que

$$\mathrm{L}_N(\mathcal{A}) \subseteq L_G.$$

Donc tout langage algébrique est reconnu par un automate à pile.
2- Soit $\mathcal{A} = \langle X, Z, Q, z_0, q_0, \lambda\rangle$. On pose $G := \langle X, V, P, \sigma\rangle$ où

$$V := \{\sigma\} \cup \{[p, z, q] \mid z \in Z, p, q \in Q\}$$

et P est l'ensemble des règles de l'une des deux formes (2.6)(2.8) suivantes :

$$\sigma{\rightarrow}_G \sum_{q \in Q} [q_0, z_0, q] \tag{2.6}$$

et pour toute transition

$$(y, q, z){\rightarrow}_{\mathcal{A}}(q', z_p z_{p-1} \cdots z_1) \tag{2.7}$$

où $y \in X \cup \{\varepsilon\}$, et tous états $q_1, q_2, \ldots, q_p$,

$$[q, z, q_p]{\rightarrow}_G y[q', z_1, q_1][q_1, z_2, q_2] \cdots [q_{p-1}, z_p, q_p] \tag{2.8}$$

(lorsque $p = 0$, le membre droit de la règle ci-dessus doit être compris comme y) ;
On montre alors que, pour tous $q \in Q, z \in Z, r \in Q$, on a

$$\mathrm{L}_G([q, z, r]) = \{f \in X^* \mid (f, q, z) \vdash_{\mathcal{A}} (\varepsilon, r, \varepsilon).$$

L' idée-clé de cette démonstration consiste à remarquer qu'un calcul de $\mathcal{A}$ débutant sur une configuration (f, q, z) en lui appliquant la transition (2.7), a une décomposition de la forme

$$(yf_1f_2\cdots f_p, q, z) \vdash_{\mathcal{A}} (f_1f_2\cdots f_p, q', z_p\cdots z_2z_1) \vdash^*_{\mathcal{A}} (f_2\cdots f_p, q_1, z_p\cdots z_2) \vdash^*_{\mathcal{A}} (\cdots f_p, q_2, z_p\cdots)$$
$$\cdots \vdash_{\mathcal{A}} (f_p, q_{p-1}, z_p) \vdash_{\mathcal{A}} (\varepsilon, q_p, \varepsilon),$$

(voir la figure 2.7). □

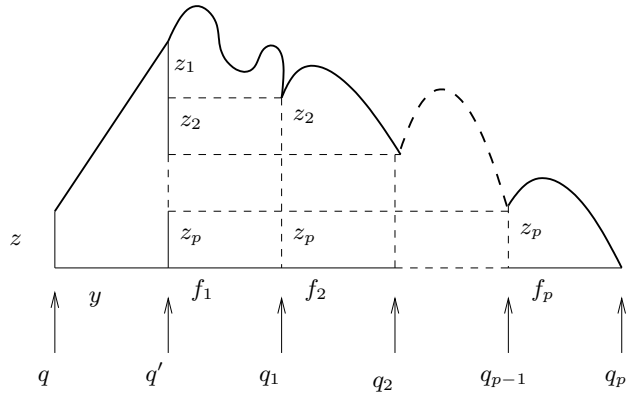


FIG. 2.7 – Un calcul de $\mathcal{A}$.

Exemple 2.7.4 *Considérons la grammaire $G = \langle X, V, P, S\rangle$ où P est constitué des règles :*

$$S \rightarrow aSS + b$$

L'automate à pile $\mathcal{A}$ fourni par la preuve du théorème 2.7.3 a les transitions suivantes :

$$\begin{aligned} (a, q, a) &\rightarrow (q, \varepsilon) \\ (b, q, b) &\rightarrow (q, \varepsilon) \\ (\varepsilon, q, S) &\rightarrow (q, SSa) \\ (\varepsilon, q, S) &\rightarrow (q, b) \end{aligned}$$

Voici un calcul de $\mathcal{A}$ reconnaissant le mot abaabbb :

$$(abaabbb, q, S) \vdash_{\mathcal{A}} (abaabbb, q, SSa) \vdash_{\mathcal{A}} (baabbb, q, SS) \vdash_{\mathcal{A}} (baabbb, q, Sb) \vdash_{\mathcal{A}} (aabbb, q, S)$$

$\vdash_{\mathcal{A}} (aabbb, q, SSa) \vdash_{\mathcal{A}} (abbb, q, SS) \vdash_{\mathcal{A}} (abbb, q, SSSa) \vdash_{\mathcal{A}} (bbb, q, SSS) \vdash^2_{\mathcal{A}} (bb, q, SS)$
$\vdash^2_{\mathcal{A}} (b, q, S) \vdash^2_{\mathcal{A}} (\varepsilon, q, \varepsilon)$.

Notons la configuration (f, q, h) *par le vecteur colonne* $\begin{pmatrix} \tilde{h} \\ f \end{pmatrix}$. *Le calcul précédent se note alors :*

$$\begin{pmatrix} S \\ abaabbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} aSS \\ abaabbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} SS \\ baabbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} bS \\ baabbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} S \\ aabbb \end{pmatrix}$$
$$\vdash_{\mathcal{A}} \begin{pmatrix} aSS \\ aabbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} SS \\ abbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} aSSS \\ abbb \end{pmatrix} \vdash_{\mathcal{A}} \begin{pmatrix} SSS \\ bbb \end{pmatrix} \vdash^2_{\mathcal{A}} \begin{pmatrix} SS \\ bb \end{pmatrix}$$
$$\vdash^2_{\mathcal{A}} \begin{pmatrix} S \\ b \end{pmatrix} \vdash^2_{\mathcal{A}} \begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}$$

On voit que la suite des transitions de type (T2) utilisées dans ce calcul forme une dérivation gauche de S *en* $abaabbb$ *dans la grammaire* G. *Les transitions de type (T1) apparaissent comme des* vérifications *que les choix de règles de dérivations engendrent effectivement le mot* $abaabbb$. *De façon générale, un calcul de* $\mathcal{A}$ *simule une dérivation gauche et toute dérivation gauche de* G *de* S *en un mot terminal est simulée par un calcul de* $\mathcal{A}$.

Exemple 2.7.5 *Considérons le langage* $L := \{u \in \{a, b\}^* \mid |u|_a = |u|_b\}$. *Il est reconnu par l'automate à pile* $\mathcal{A} := \langle \{a, b\}, \{\Omega, F, F'\}, \{q\}, \Omega, q, \lambda \rangle$ *où* λ *est constitué des transitions*

$$\begin{array}{rcl} (q, \Omega) & \xrightarrow{a} & (q, \Omega F) \\ (q, \Omega) & \xrightarrow{b} & (q, \Omega F') \\ (q, F) & \xrightarrow{a} & (q, FF) \\ (q, F) & \xrightarrow{b} & (q, \varepsilon) \\ (q, F') & \xrightarrow{a} & (q, \varepsilon) \\ (q, F') & \xrightarrow{b} & (q, F'F') \\ (q, \Omega) & \xrightarrow{\varepsilon} & (q, \varepsilon) \end{array}$$

Le symbole Ω *est un marqueur de fond de pile. Le contenu de pile après lecture de* u *représente la valeur de* $\pi(u) = |u|_a - |u|_b$, *codée par* $F^{\pi(u)}$ *si* $\pi(u) \geq 0$ *et par* $F'^{-\pi(u)}$ *si* $\pi(u) \leq 0$. *La grammaire algébrique fournie par la preuve du théorème 2.7.3 a les règles suivantes :*

$$[q, \Omega, q] \rightarrow a[q, F, q][q, \Omega, q] + b[q, F', q][q, \Omega, q] + \varepsilon$$

$$[q, F, q] \to a[q, F, q][q, F, q] + b$$
$$[q, F', q] \to b[q, F', q][q, F', q] + a$$

Après renommage des variables on obtient la grammaire $G = \langle \{a, b\}, \{S, T, T'\}, P, S \rangle$ *où* P *consiste en les règles :*

$$S \to aTS + bT'S + \varepsilon, \;\; T \to aTT + b, \;\; T' \to bT'T' + a.$$

Exercice 2.7.6 *Montrer que la grammaire ci-dessus est non-ambiguë*

Chapitre 3

Langages rationnels de termes

3.1 Termes, F-algèbres

Termes Un alphabet *gradué* est un couple $(F, \mathbf{ar})$ où $\mathbf{ar}$ est une application de F dans $\mathbb{N}$. Un *terme* sur l'alphabet F est un arbre planaire t ordonné étiqueté sur F qui respecte les arités, en ce sens que :

$$\forall u \in \mathrm{dom}(t), \forall i \in \mathbb{N}, [(u \cdot i) \in \mathrm{dom}(t)] \Leftrightarrow [i \leq \mathbf{ar}(t(u)) - 1].$$

On note $\mathrm{T}(F)$ l'ensemble des termes sur l'alphabet gradué F.
Etant donné un alphabet gradué F et un alphabet gradué de variables V, toutes d'arité 0, on appelle terme sur F avec des variables dans V un terme de $\mathrm{T}(F \cup V)$. Un terme $t \in \mathrm{T}(F \cup V)$ est dit *linéaire* ssi chaque variable $v \in V$ a au plus une occurrence dans t. On dénote aussi par $\mathrm{T}(F, V)$ l' ensemble des termes sur l'alphabet gradué F et l'ensemble de variables V.

F-algèbres

Définition 3.1.1 *Une F-algèbre $\mathcal{M}$ est un tuple $\langle D, (f^{(\mathcal{M})})_{f \in F} \rangle$ où D est un ensemble et, pour tout $f \in F$, $f^{(\mathcal{M})}$ est une application de $D^{\mathbf{ar}(f)}$ dans*

D.
D est le domaine de $\mathcal{M}$ et les $f^{(\mathcal{M})}$ sont les opérations de $\mathcal{M}$.

$\mathrm{T}(F,V)$ est muni d'une structure de F-algèbre en associant à chaque symbole $f \in F$ d'arité k, l'opération k-aire, $\hat{f} : \mathrm{T}(F,V) \to \mathrm{T}(F,V)$ telle que, pour tous $t_1, \ldots, t_k \in \mathrm{T}(F,V)$, $\hat{f}(t_0, \ldots, t_{k-1})$ est le terme t défini par :

$$\begin{aligned} \mathrm{dom}(t) &:= \{\varepsilon\} \cup [0 \cdot \mathrm{dom}(t_0)] \ldots \cup [j \cdot \mathrm{dom}(t_j)] \ldots \cup [(k-1) \cdot \mathrm{dom}(t_{k-1})], \\ t(\varepsilon) &:= f \\ t(j \cdot u) &=: t_j(u) \text{ pour tous } j \in [0, k-1], u \in \mathrm{dom}(t_j). \end{aligned}$$

(Dans la pratique on omet souvent le "chapeau" sur le symbole $f : f(a, b, g(b, a))$, par exemple, dénote le même terme que $\hat{f}(a, b, \hat{g}(b, a))$).

Exemple 3.1.2 *Soit l'alphabet gradué $F := \{f, m, a\}$ avec $\mathbf{ar} : f \mapsto 2, m \mapsto 1, a \mapsto 0\}$. Nous définissons deux F-algèbres $\mathcal{M}_i = \langle D_i, f_i, m_i, a_i \rangle$ ($i \in \{1, 2\}$) comme suit :*

$$D_1 := \mathbb{Z}, \quad f_1(x, y) := x + y, \quad m_1(x) := -x, \quad a_1 := 1.$$

$$D_2 := \{0, 1, 2\}, \quad f_2(x, y) := x + y (mod 3), \quad m_2(x) := -x (mod 3), \quad a_2 := 1.$$

Soit $h : D_1 \to D_2$ définie par

$$\forall n \in \mathbb{Z}, h(n) := n (mod 3).$$

Cette application h est un homomorphisme de F-algèbre de $\mathcal{M}_1$ dans $\mathcal{M}_2$.

Définition 3.1.3 *Soient $\mathcal{M}_i = \langle D_i, (f^{(\mathcal{M}_i)})_{f \in F} \rangle$ ($i \in \{1, 2\}$), deux F-algèbres. Un homomorphisme de F-algèbres de $\mathcal{M}_1$ dans $\mathcal{M}_2$ est une application $h : D_1 \to D_2$ qui est compatible avec les opérations i.e. telle que, pour tout $f \in F$ et tous $d_1, \ldots d_k \in D_1$ avec $k = \mathbf{ar}(f)$,*

$$h(f^{(\mathcal{M}_1)}(d_1, \ldots, d_k)) = f^{(\mathcal{M}_2)}(h(d_1), \ldots, h(d_k)).$$

La F-algèbre $\mathrm{T}(F,V)$ possède la propriété universelle suivante :

Théorème 3.1.4 *Soit $\mathcal{M} = \langle D, (f^{(\mathcal{M})})_{f\in F}\rangle$ une F-algèbre et soit $\varphi : V \to D$ une application. Alors, il existe un unique prolongement de φ en un homomorphisme de F-algèbres $\Phi : \mathrm{T}(F, V) \to \mathcal{M}$.*

Preuve : L'unicité de $\Phi(t)$ se prouve par récurrence sur $\|t\|$. On constate alors que l'application Φ définie par cette valeur est bien un F-morphisme. □

Exemple 3.1.5 *Soit l'alphabet gradué $F := \{a, b, \#\}$ avec* **ar** $: a \mapsto 1, b \mapsto 1, \# \mapsto 0$. *Nous définissons une F-algèbres $\mathcal{M} = \langle D, (f^{(\mathcal{M})})_{f\in F}\rangle$ comme suit :*
$D := \{0, 1, 2\}$, $\#^{(\mathcal{M})} := 0$ *et les applications $a^{(\mathcal{M})}, b^{(\mathcal{M})}$ sont définies par le tableau :*

x	0	1	2
$a^{(\mathcal{M})}(x)$	1	2	1
$b^{(\mathcal{M})}(x)$	0	0	0

Soit $h : \mathrm{T}(F) \to \mathcal{M}$ l'unique homomorphisme de F-algèbre de $\mathrm{T}(F)$ dans $\mathcal{M}$. Il associe a tout mot $w\#$ (où $w \in \{a, b\}^*$), l'etat final atteint après lecture du mot miroir de w par l'automate $\mathcal{A}$ de la figure 3.1.
Par exemple, les valeurs de h sur les suffixes du mot $w = aaabaababaaaaba\#$ sont :

$\#$	0
$a\#$	1
$ba\#$	0
$aba\#$	1
$aaba\#$	2
$aaaba\#$	1
$aaaaba\#$	2
$baaaaba\#$	0
$abaaaaba\#$	1
$babaaaaba\#$	0
$ababaaaaba\#$	1
$aababaaaaba\#$	2
$baababaaaaba\#$	0
$abaababaaaaba\#$	1
$aabaababaaaaba\#$	2
$aaabaababaaaaba\#$	1

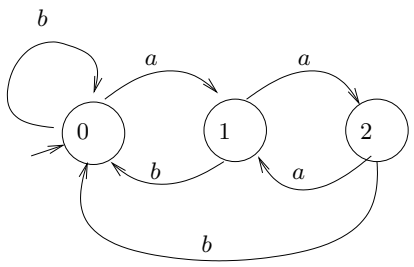


FIG. 3.1 – Automate $\mathcal{A}$.

Substitution de termes Etant donnés, pour chaque variable $v \in V$ un terme $t_v \in \mathrm{T}(F, V)$ on dénote par

$$\{v \leftarrow t_v \mid v \in V\}$$

l'unique morphisme de F-algèbre $\sigma : \mathrm{T}(F, V) \to \mathrm{T}(F, V)$ tel que

$$\forall v \in V, v\sigma = t_v$$

(on dénote ici par $t\sigma$ l'image de t par σ). L'application $\sigma : \mathrm{T}(F, V) \to \mathrm{T}(F, V)$ s'appelle une *substitution*.
Point de vue plus combinatoire : pour tout terme $t \in \mathrm{T}(F, V)$, $t\sigma$ est le terme $t' \in \mathrm{T}(F, V)$ obtenu en remplaçant, dans t, chaque feuille étiquetée par v par le terme t_v.

Exemple 3.1.6 *Soit* $F := \{F, m, a\}$ *avec* $\mathbf{ar}(f) = 2, \mathbf{ar}(m) = 1, \mathbf{ar}(a) = 0$ *et* $V := \{v_1, v_2\}$ *(les variables ont une arité nulle).*
Soit $t := f(m(m(v_1)), f(v_2, f(a, v_1)))$ *et*

$$\sigma := \{v_1 \leftarrow f(v_2, m(a)), \quad v_2 \leftarrow m(f(a, a))\}$$

On a alors

$$t\sigma = f(m(m(f(v_2, m(a)))), f(m(f(a, a)), f(a, f(v_2, m(a))))).$$

Substitution de langages $\mathcal{P}(\mathrm{T}(F, V))$ est muni d'une structure de F-algèbre en associant à chaque symbole $f \in F$ d'arité k, l'opération k-aire, $\hat{f} : \mathcal{P}(\mathrm{T}(F, V)) \to \mathcal{P}(\mathrm{T}(F, V))$ définie par : pour tous $T_1, \ldots, T_k \in \mathcal{P}(\mathrm{T}(F, V))$,

$$\hat{f}(T_1, \ldots, T_k) := \{\hat{f}(t_1, \ldots, t_k \mid t_1 \in T_1, \ldots, t_k \in T_k\}.$$

Définition 3.1.7 *Etant donné un alphabet gradué F, on appelle* substitution *de langages toute application*

$$\Phi : \mathcal{P}(\mathrm{T}(F,V)) \to \mathcal{P}(\mathrm{T}(F,V))$$

vérifiant les 3 propriétés suivantes :
(S0) $\Phi(\emptyset) = \emptyset$
(S1) $\forall k \in \mathbb{N}, f \in F$ *avec* $\mathbf{ar}(f) = k, \forall A_1, \ldots, A_k \in \mathcal{P}(\mathrm{T}(F,V)),\ \Phi(\hat{f}(A_1, \ldots, A_k)) = \hat{f}(\Phi(A_1), \ldots, \Phi(A_k))$
(S2) *pour toute suite A_n d'éléments de $\mathcal{P}(\mathrm{T}(F,V))$, $\Phi(\bigcup_{n\in\mathbb{N}} A_n) = \bigcup_{n\in\mathbb{N}} \Phi(A_n)$.*

Remarquons que Φ est entièrement déterminée par $(\Phi(\{v\}))_{v\in V}$.
Etant donnés, pour chaque variable $v \in V$ un ensemble de termes $T_v \in \mathcal{P}(\mathrm{T}(F,V))$, on dénote par

$$\{v \leftarrow T_v \mid v \in V\}$$

l'unique substitution de langages Φ telle que $\forall v \in V, v\Phi = T_v$.
Point de vue plus combinatoire : pour tout ensemble de termes $T \in \mathrm{T}(F,V)$, $T\Phi$ est l'ensemble des termes $t' \in \mathrm{T}(F,V)$ obtenus en remplaçant, dans un terme $t \in T$, chaque feuille étiquetée par v par un terme $t_v \in T_v$ (mais le choix de t_v peut être différent d'une occurrence de v à une autre occurrence de v dans le même terme t).

Exemple 3.1.8 *On reprend les alphabets F, V de l'exemple 3.1.6. Soit $t := f(m(m(v_1)), f(v_2, f(a, v_1)))$, $T_1 := \{f(v_2, m(a)), a\}, T_2 := \{m(f(a,a))\}$ et*

$$\Phi := \{v_1 \leftarrow T_1,\ \ v_2 \leftarrow T_2\}$$

On a alors

$$\begin{aligned} t\Phi \ = \ & \{f(m(m(f(v_2, m(a)))), f(m(f(a,a)), f(a, f(v_2, m(a))))), \\ & f(m(m(f(v_2, m(a)))), f(m(f(a,a)), f(a,a))) \\ & f(m(m(a)), f(m(f(a,a)), f(a, f(v_2, m(a))))),\ \ f(m(m(a)), f(m(f(a,a)), f(a,a)))\}. \end{aligned}$$

Contextes On appelle *contexte* un terme de $\mathrm{T}(F, \{\bullet\})$ contenant exactement une occurrence de $\bullet$ (où $\bullet$ est une variable distinguée). Etant donné un contexte C et un terme $t \in \mathrm{T}(F')$, on dénote par $C[t]$ le terme $C\{\bullet \leftarrow t\}$.
On dénote par $\mathcal{C}_1(F)$ l'ensemble des contextes sur F.

3.2 Automates finis

Définition 3.2.1 (Automate) *Un* automate *de termes est un 4-uplet,* $\mathcal{A} = \langle F, Q, Q_+, \delta \rangle$ *où*
– F *est un alphabet gradué*
– Q *est un ensemble fini, l'ensemble des états*
– $Q_+ \subseteq Q$ *est l'ensemble des états d'arrivée*
– $\delta \subseteq \mathrm{T}(F \cup Q) \times Q$ *est l'ensemble des transitions ; chaque transition est de la forme :*

$$f(q_0, q_1, \ldots, q_{k-1}) \rightarrow q$$

où $\mathbf{ar}(f) = k$.
Il est dit fini *lorsque l'ensemble* Q *est fini.*

On appelle *calcul* (ou *course*) de l'automate $\mathcal{A}$ sur le terme t, toute application $\rho : \mathrm{dom}(t) \rightarrow Q$ telle que, $\forall u \in \mathrm{dom}(t)$:

$$\text{si } \mathbf{ar}(t(u)) = k \geq 1 \quad \text{alors } t(u)(\rho(u0), \ldots, \rho(u \cdot (k-1))) \rightarrow \rho(u) \in \delta \quad (3.1)$$

et

$$\text{si } \mathbf{ar}(t(u)) = 0 \quad \text{alors } t(u) \rightarrow \rho(u) \in \delta \quad (3.2)$$

Définition 3.2.2 *Un terme* $t \in \mathrm{T}(F)$ *est accepté (ou reconnu) par* $\mathcal{A}$*, ssi il existe un calcul* ρ *de* $\mathcal{A}$ *sur* t*, tel que* $t(\varepsilon) \in Q_+$*. On appelle langage accepté (ou reconnu) par* $\mathcal{A}$*, noté* $\mathrm{L}_{\mathcal{A}}$*, l'ensemble des termes acceptés (ou reconnnus) par* $\mathcal{A}$*.*
Un langage $L \subseteq \mathrm{T}(F)$ *est dit* reconnaissable *ssi il existe un automate fini (de termes)* $\mathcal{A}$ *tel que* $L = \mathrm{L}_{\mathcal{A}}$*.*

Exemple 3.2.3 *Considérons l'automate fini de termes* $\mathcal{A} = \langle F, Q, Q_+, \delta \rangle$ *où* F *est l'alphabet gradué de l'exemple 3.1.6,* $Q := \{\bot, 0, 1, 2\}, Q_+ := \{2\}$ *et* δ *est formé des transitions suivantes :*

$$a \rightarrow \bot, \;\; m(\bot) \rightarrow \bot, \;\; m(\bot) \rightarrow 0, \;\; m(0) \rightarrow 1, \;\; m(1) \rightarrow 2$$

$$f(q, r) \rightarrow \bot, \;\; f(0, q) \rightarrow 1, \;\; f(q, 0) \rightarrow 1 \;\; f(q, 1) \rightarrow 2 \;\; f(1, q) \rightarrow 2$$

pour tous $q, r \in Q$. On vérifie que $L_{\mathcal{A}} = \{t \in \mathrm{T}(F) \mid \exists u \in \mathrm{dom}(t), |u| = 2$ et $t(u) = m\}$. Autrement dit, $\mathcal{A}$ reconnaît l'ensemble des termes qui possèdent un symbole m à distance 2 de la racine. La figure 3.2 montre deux courses de $\mathcal{A}$ sur un même terme.

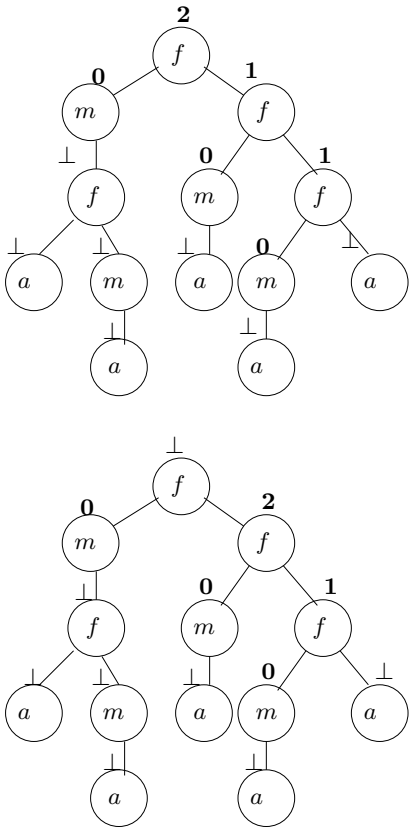


FIG. 3.2 – Des courses de $\mathcal{A}$.

On appelle *relation de réduction immédiate* de l'automate, notée $\rightarrow_{\mathcal{A}}$, la plus petite relation binaire sur $\mathrm{T}(F \cup Q)$ (où les éléments de Q ont une arité 0), qui contient δ et qui est compatible avec les contextes c'est à dire :
R1- si $(f(q_1, \ldots, q_k), q) \in \delta$ alors $f(q_1, \ldots, q_k) \rightarrow_{\mathcal{A}} q$
R2- si $t \rightarrow_{\mathcal{A}} t'$ et si C est un contexte, alors $C[t] \rightarrow_{\mathcal{A}} C[t']$.
On note $\stackrel{*}{\rightarrow}_{\mathcal{A}}$ la clôture réflexive et transitive de $\rightarrow_{\mathcal{A}}$. On appelle cette relation la réduction associée à l'automate $\mathcal{A}$. On remarque que, pour tout terme $t \in \mathrm{T}(F)$ et tout état $q \in Q$, $t \stackrel{*}{\rightarrow}_{\mathcal{A}} q$ ssi, il existe une course ρ de $\mathcal{A}$

sur t telle que $\rho(\varepsilon) = q$.

L'automate $\mathcal{A}$ est dit *déterministe* ssi pour tout $f \in F$ et tout $\overrightarrow{q} \in Q^k$, il existe au plus un état $r \in Q$ tel que $f(\overrightarrow{q}) \to r \in \delta$.
L'automate $\mathcal{A}$ est dit *complet* ssi pour tout $f \in F$ et tout $\overrightarrow{q} \in Q^k$, il existe au moins un état $q \in Q$ tel que $f(\overrightarrow{q}) \to r \in \delta$.

Exemple 3.2.4 *Considérons l'automate fini de termes $\mathcal{A}' = \langle F, Q', Q'_+, \delta' \rangle$ où F est l'alphabet gradué de l'exemple 3.1.6, $Q' := \{0,1\}, Q'_+ := \{0\}$ et δ est formé des transitions suivantes :*

$$a \to 1, \quad m(0) \to 0, \quad m(1) \to 1,$$

$$f(0,0) \to 0, \quad f(0,1) \to 1, \quad f(1,0) \to 1 \quad f(1,1) \to 0.$$

Cet aft $\mathcal{A}'$ est déterministe et complet. On vérifie que $L_{\mathcal{A}} = \{t \in \mathrm{T}(F) \mid t$ possède un nombre pair de symboles $a\}$. La figure 3.3 montre la course de $\mathcal{A}$ sur le terme de la figure 3.2.

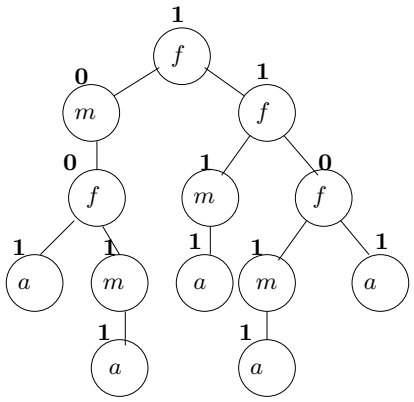


FIG. 3.3 – La course de $\mathcal{A}'$.

Théorème 3.2.5 *Soit $\mathcal{A}$ un automate fini de termes. On peut construire un automate fini de termes $\mathcal{B}$, tel que $\mathcal{B}$ est déterministe, complet et $\mathrm{L}_{\mathcal{A}} = \mathrm{L}_{\mathcal{B}}$.*

Preuve : Soit $\mathcal{A} = \langle F, Q, Q_+, \delta\rangle$ un automate fini (non-déterministe) de termes. Construisons un automate fini déterministe complet $\mathcal{B}$ tel que $L_\mathcal{A} = L_\mathcal{B}$. Posons

$$\mathcal{B} = \langle F, \mathcal{R}, \mathcal{R}_+, \Delta\rangle$$

où $\mathcal{R} = \mathcal{P}(Q)$, $\mathcal{R}_+ = \{P \in \mathcal{P}(Q), P \cap Q_+ \neq \emptyset\}$ $\Delta \subseteq \mathrm{T}(F \cup \mathcal{P}(Q)) \times \mathcal{P}(Q)$ est l'ensemble de toutes les transitions de la forme :

$$f(P_1, \ldots, P_k) \longrightarrow \{q \in Q, \exists p_1 \in P_1, \ldots, \exists p_k \in P_k, (f(p_1, \ldots, p_k), q) \in \delta\}$$

pour $f \in F, \mathbf{ar}(f) = k, P_1, P_2, \ldots, P_k \in \mathcal{P}(Q)$. On démontre par récurrence sur $\|t\|$ que :

$$t \xrightarrow{*}_\mathcal{B} \{q \in Q \mid t \xrightarrow{*}_\mathcal{A} q\}. \tag{3.3}$$

Soit $t \in \mathrm{T}(F)$. Comme $\mathcal{B}$ est déterministe et complet, il existe un unique état $R \in \mathcal{R}$ tel que $t \xrightarrow{*}_\mathcal{B} R$. D'après (3.3) on a alors :

$$\begin{aligned} t \in L_\mathcal{A} &\Leftrightarrow \exists q \in Q_+, t \xrightarrow{*}_\mathcal{A} q \\ &\Leftrightarrow \exists q \in Q_+, q \in R \\ &\Leftrightarrow R \in \mathcal{R}_+. \end{aligned}$$

Donc $\mathrm{L}_\mathcal{A} = \mathrm{L}_\mathcal{B}$. □

3.3 Lemme d'itération

Etant donnés deux contextes C, C' on note $C \circ C'$ la *composition* de ces contextes, définie par

$$C \circ C' := C\{\bullet \leftarrow C'\}.$$

l'expression C^n dénote alors la puissance n-ième de C, pour cette opération de composition.

Théorème 3.3.1 *Soit $L \subseteq \mathrm{T}(F)$ un langage reconnaissable. Il existe un entier k tel que, pour tout $t \in L$, si $\|t\| \geq k$, alors t se factorise en $t = B[C[d]]$ avec B, C contextes, $d \in \mathrm{T}(F)$ et*
(1) $\forall n \in \mathbb{N}, B[C^n[d]] \in L$
(2) $\|C\| \geq 2$
(3) $\|C[d]\| \leq k - 1$.

Soit $\mathcal{A}$ un af reconnaissant L. Notons

$$\ell := \max(\{\mathbf{ar}(f), f \in F\} \cup \{2\}), c := \mathrm{Card}(Q_{\mathcal{A}})$$

et posons

$$k := \ell^{c+1} + \ell.$$

Lemme 3.3.2 *Soit t un terme de hauteur inférieure ou égale à h. Alors $\|t\| \leq \ell^{h+1}$.*

Preuve : On prouve par récurrence sur h que le nombre de noeuds de t de longueur h est majoré par ℓ^h. Donc

$$\|t\| \leq \ell^0 + \ell^1 + \ldots + \ell^k + \ldots + \ell^h$$

Or $\sum_{k=0}^{h} \ell^k = \frac{\ell^{h+1}-1}{\ell-1} \leq \ell^{h+1}$. □
Montrons le théorème 3.3.1.
Soit $t \in \mathrm{L}_{\mathcal{A}}$ tel que $\|t\| \geq k/\ell$ et soit ρ une course de $\mathcal{A}$ sur t telle que $\rho(\varepsilon) \in Q_{\mathcal{A},+}$. D'après le lemme 3.3.2, il existe une branche b de T de longueur au moins $c = \mathrm{Card}(Q_{\mathcal{A}})$:

$$b = x_0, x_1, \ldots, x_p$$

avec $p \geq \mathrm{Card}(Q_{\mathcal{A}})$. Toutes les étiquettes $\rho(x_0), \rho(x_1), \ldots, \rho(x_p)$ appartiennent à $Q_{\mathcal{A}}$, donc il existe un couple $(i, j) \in [0, p] \times [0, p]$ tel que

$$i < j \text{ et } \rho(x_i) = \rho(x_j).$$

Choisissons une branche b et un couple (i, j), parmi ceux vérifiant la propriété ci-dessus, tels que $\|(t/x_i)\|$ soit minimale. Notons $q = \rho(x_i) = \rho(x_j)$. Définissons alors des contextes B, C et un terme d par :

$$\mathrm{dom}(B) := \{u \in \mathrm{dom}(t) \mid x_i \not< u\}, \quad \forall u \in \mathrm{dom}(B) - \{x_i\}, B(u) := t(u), \quad B(x_i) := \bullet,$$

$$\mathrm{dom}(C) := \{u \in \mathbb{N}^* \mid x_i \cdot u \in \mathrm{dom}(t) \text{ et } x_j \not< x_i \cdot u\},$$
$$\forall u \in \mathrm{dom}(C) - \{x_i^{-1} x_j\}, C(u) := t(x_i \cdot u), \quad C(x_i^{-1} x_j) := \bullet,$$
$$d := t/x_j.$$

On a alors $t = B[C[d]]$ (voir la figure 3.4). Comme $q = \rho(x_i) = \rho(x_j)$ on a :

$$d \xrightarrow{*}_{\mathcal{A}} q, \quad C[q] \xrightarrow{*}_{\mathcal{A}} q, \quad B[q] \xrightarrow{*}_{\mathcal{A}} \rho(\varepsilon).$$

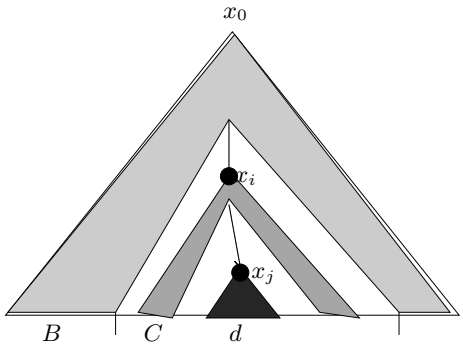


FIG. 3.4 – La décomposition de t.

On en déduit que, pour tout $n \geq 0$, $B[C^n[d]] \xrightarrow{*}_{\mathcal{A}} \rho(\varepsilon)$, donc $B[C^n[d]] \in L$, ce qui est la propriété (1) demandée.
Le fait que $x_i \neq x_j$ nous garantit que $\mathrm{dom}(C)$ a au moins deux éléments : ε et $x_i^{-1}x_j$, ce qui prouve la propriété (2) demandée.
Enfin, si $\|C[d]\| \geq k$ alors, l'un des sous-termes t' de $C[d]$, dont la racine est un fils de ε, a une norme $\geq k/\ell$ et par le raisonnement ci-dessus, ce sous-terme t' possède une branche b' sur laquelle la course ρ a une répétition, ce qui contredirait la minimalité de $\|t/x_i\|$. Donc $\|C[d]\| \leq k-1$ i.e. l'inégalité (3) est vérifiée.□

Corollaire 3.3.3 *Etant donné un aft $\mathcal{A}$, on peut décider si* $\mathrm{L}(\mathcal{A}) = \emptyset$.

En effet, d'après le théorème 3.3.1, $\mathrm{L}(\mathcal{A})$ reconnaît au moins un terme ssi il reconnaît au moins un terme de norme $\leq k-1$.

Corollaire 3.3.4 *Etant donné un aft $\mathcal{A}$, on peut décider si* $\mathrm{L}(\mathcal{A})$ *est infini.*

En effet, d'après le théorème 3.3.1, $\mathrm{L}(\mathcal{A})$ reconnaît une infinité de termes ssi il reconnaît au moins un terme t tel que $k \leq \|t\| \leq 2k$.

3.4 Propriétés de clôture

Nous souhaitons définir, pour les termes, une notion d'*homomorphisme* qui n'exprime pas une compatibilité avec les opérations de F-algèbre, mais plutôt une compatibilité avec la substitution. Une autre façon de voir les termes de $\mathrm{T}(F,V)$ est de les considérer comme des applications de $\mathrm{T}(F,V)$ dans lui-même. De ce point de vue un homomorphisme de termes est une application qui est compatible avec l'opération de *composition* des applications.

Définition 3.4.1 *Soient F, F' des alphabets gradués et V un ensemble de variables. On appelle homomorphisme de termes, de $\mathrm{T}(F,V)$ dans $\mathrm{T}(F',V)$ toute application $\varphi : \mathrm{T}(F,V) \to \mathrm{T}(F',V)$ vérifiant*
(HT1) $\forall v \in V, \varphi(v) = v$
(HT2) $\forall t \in \mathrm{T}(F,V), var(\varphi(t)) \subseteq var(t)$
(HT3) $\forall t, t' \in \mathrm{T}(F,V), \forall v \in V, \varphi(t\{v \leftarrow t'\}) = \varphi(t)\{v \leftarrow \varphi(t')\}$

On remarquera qu'étant donné, pour tout symbole $f \in F$ d'arité k, un terme $t_f \in \mathrm{T}(F', \{v_1, \ldots, v_k\})$, il existe un unique homomorphisme de termes $\varphi : \mathrm{T}(F,V) \to \mathrm{T}(F',V)$ tel que,

$$\forall k \in \mathbb{N}, \forall f \in F_k, \varphi(f(v_1, \ldots, v_k)) = t_f.$$

Exemple 3.4.2 *Soit $F := \{f, m, a\}$, d'arités respectives $2, 1, 0$ et $F' := \{f, m, a, b\}$, d'arités respectives $2, 1, 0, 0$. Soit H l' unique homomorphiseme de termes de $\mathrm{T}(F,V)$ dans $\mathrm{T}(F',V)$ tel que*

$$H(f(v_1, v_2)) = f(f(m(a), v_2)v_1), \quad H(m(v_1)) = f(v_1, v_1), \quad H(a) = b.$$

On obtient

$$H(m(f(f(a,a), m(a)))) =$$

$$f(f(f(m(a), f(b,b)), f(f(m(a), b), b)), f(f(m(a), f(b,b)), f(f(m(a), b), b))).$$

Le calcul de H sur $m(f(f(a,a), m(a)))$ est représenté sur les figures (3.5-3.6).

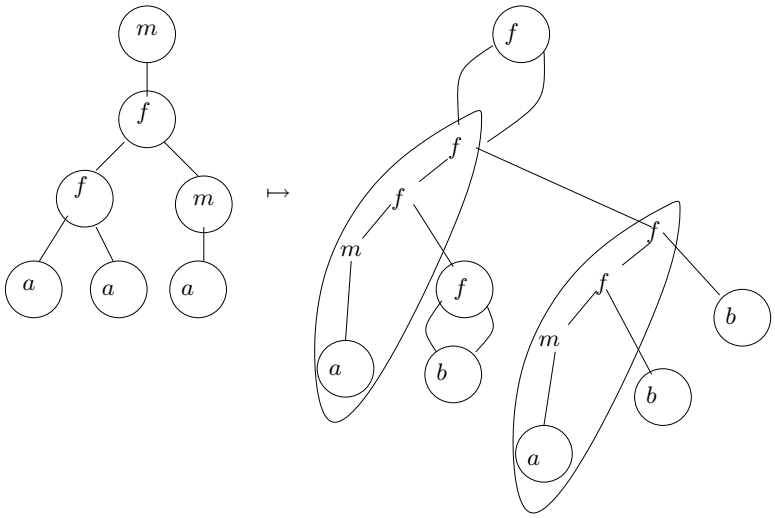


FIG. 3.5 – Image par H : le graphe.

Un homomorphisme de termes φ est dit *linéaire* ssi il envoie tout terme linéaire sur un terme linéaire. Cette condition est équivalente au fait que l'image de chaque $f(v_1, \ldots, v_k)$, pour chaque symbole f d'arité k, est un terme linéaire.

Théorème 3.4.3 *La famille des langages reconnaissables de termes est close par*
1- union
2- complémentation
3- intersection
4- homomorphisme inverse
5- homomorphisme linéaire.

Nous utiliserons dans la preuve de ce théorème une notion plus large d'automate fini de termes.

Définition 3.4.4 *Un* automate *généralisé de termes est un 5-uplet,* $\mathcal{A} = \langle F, Q, Q_+, \delta \rangle$ *où*
– *F est un alphabet gradué*
– *Q est un ensemble fini, l'ensemble des états*

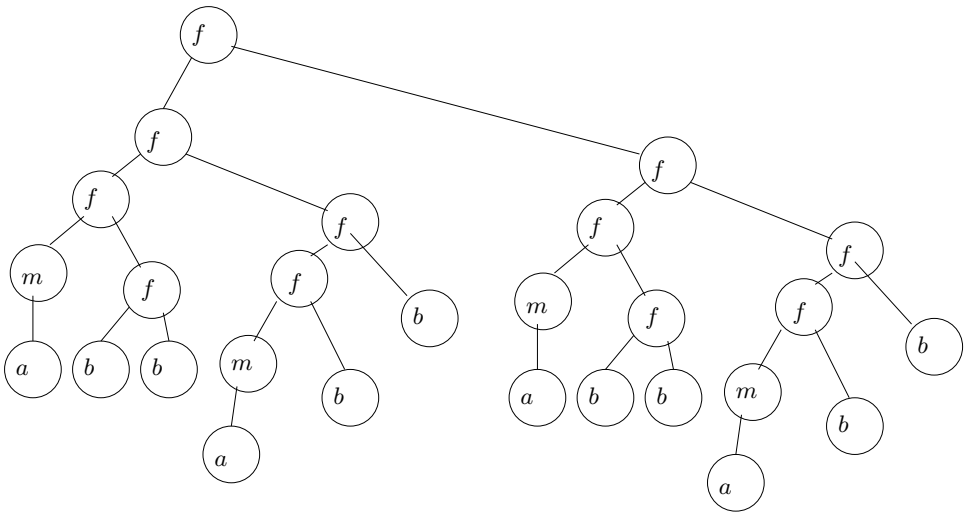


FIG. 3.6 – Image par H : le terme.

– $Q_+ \subseteq Q$ *est l'ensemble des états d'arrivée*
– $\delta \subseteq \mathrm{T}(F \cup Q) \times Q$ *est l'ensemble des transitions ; chaque transition est de l'une des deux formes :*

$$f(q_0, q_1, \ldots, q_{k-1}) \rightarrow r \quad \text{où } q_i, r \in Q, \mathbf{ar}(f) = k \tag{3.4}$$
$$q \rightarrow r \quad \text{où } q, r \in Q. \tag{3.5}$$

La notion de course serait plus délicate à définir pour ces automates généralisés que pour les automates ordinaires ; en revanche, la notion de *réduction* associée à l'automate est définie exactement comme pour un automate ordinaire.

Lemme 3.4.5 *Etant donné un aft généralisé $\mathcal{A}$ on peut construire un aft déterministe complet $\mathcal{B}$ tel que $\mathrm{L}_\mathcal{A} = \mathrm{L}_\mathcal{B}$.*

La preuve de ce lemme est une adaptation de la preuve du théorème 3.2.5. La seule modification consiste à définir l'ensemble des transitions Δ comme l'ensemble de toutes les transitions de la forme :

$$f(P_1, \ldots, P_k) \longrightarrow \{q \in Q, \exists p_1 \in P_1, \ldots, \exists p_k \in P_k, f(p_1, \ldots, p_k) \xrightarrow{*}_\mathcal{A} q\}$$

pour $f \in F, \mathbf{ar}(f) = k, P_1, P_2, \ldots, P_k \in \mathcal{P}(Q)$. Nous prouvons le théorème 3.4.3 en fournissant, pour chaque opération sur les langages, une opération correspondante sur les automates.

Union :
Soient $\mathcal{A} = \langle F, Q, Q_+, \delta\rangle, \mathcal{A}' = \langle F, Q', Q'_+, \delta'\rangle$ deux aft. On peut supposer, sans perte de généralité, que $Q \cap Q' = \emptyset$. Posons $\mathcal{B} = \langle F, Q \cup Q', Q_+ \cup Q'_+, \delta \cup \delta'\rangle$. On vérifie que

$$\mathrm{L}_{\mathcal{B}} = \mathrm{L}_{\mathcal{A}} \cup \mathrm{L}_{\mathcal{A}'}.$$

Complémentation :
Soit $\mathcal{A} = \langle F, Q, Q_+, \delta\rangle$ un aft. On peut supposer, sans perte de généralité (selon le théorème 3.2.5) que $\mathcal{A}$ est déterministe complet . Posons $\mathcal{B} = \langle F, Q, Q - Q_+, \delta\rangle$. On vérifie que

$$\mathrm{L}_{\mathcal{B}} = \mathrm{T}(F) - \mathrm{L}_{\mathcal{A}}.$$

Intersection :
Comme $L \cap L' = \mathrm{T}(F) - ((\mathrm{T}(F) - L) \cup (\mathrm{T}(F) - L'))$, la propriété de clôture par intersection résulte des propriétés de clôture par union et complémentation.

Homomorphisme inverse :
Soit $\mathcal{A} = \langle F', Q, Q_+, \delta\rangle$ un aft déterministe et soit $\varphi : \mathrm{T}(F, V) \to \mathrm{T}(F', V)$ un homomorphisme de termes. On étend φ en un homomorphisme de termes $\tilde{\varphi} : \mathrm{T}(F \cup Q, V) \to \mathrm{T}(F' \cup Q, V)$ en posant : pour tout $q \in Q$,

$$\tilde{\varphi}(q) = q$$

et pour tous $f \in F, v_i \in V$, tels que $\mathbf{ar}(f) = k$

$$\tilde{\varphi}(f(v_0, \ldots, v_{k-1})) = \varphi(f(v_0, \ldots, v_{k-1})).$$

Posons $\mathcal{B} = \langle F, Q, Q_+, \lambda\rangle$ où λ est l'ensemble des transitions suivantes :

$$f(q_0, \ldots, q_{k-1}) \to q$$

ssi, $\tilde{\varphi}(f(q_0, \ldots, q_{k-1})) \xrightarrow{*}_{\mathcal{A}} q$. On vérifie que

$$\mathrm{L}_{\mathcal{B}} = \varphi^{-1}(\mathrm{L}_{\mathcal{A}}).$$

N.B. L'hypothèse que $\mathcal{A}$ est déterministe est essentielle pour montrer que les transitions de $\mathcal{B}$ sur t simulent un calul de $\mathcal{A}$ sur $\varphi(t)$: en effet les calculs de $\mathcal{A}$ sur différentes copies de l'image d'un sous-terme de t doivent être identiques.

Homomorphisme linéaire :

Soit $\mathcal{A} = \langle F, Q, Q_+, \delta\rangle$ un aft et soit $\varphi : \mathrm{T}(F,V) \to \mathrm{T}(F',V)$ un homomorphisme linéaire d'arbres.
Cas 1 Homomorphisme alphabétique.
On suppose ici que, pour tout symbole $f \in F$, $\varphi(f(v_1,\ldots,v_k)) = g(v_1,\ldots,v_k)$ où $g \in F'$. Posons $\mathcal{B} = \langle F', Q, Q_+, \lambda\rangle$, où

$$\lambda := \{\varphi(f(q_0,\ldots,q_{k-1}))\to q \mid f(q_0,\ldots,q_{k-1})\to_{\mathcal{A}} q\}.$$

On vérifie que

$$\mathrm{L}_{\mathcal{B}} = \varphi(\mathrm{L}_{\mathcal{A}}).$$

Cas 2 Homomorphisme linéaire général.
Soit $V := \{v_1,\ldots,v_n,\ldots\}$ un ensemble de variables. Posons $\mathcal{B} = \langle F', Q_{\mathcal{B}}, Q_{\mathcal{B},+}, \lambda\rangle$, où

$$\begin{aligned} Q_{\mathcal{B}} \;:=\; & \{[t\sigma] \mid \exists f \in F \mid \mathbf{ar}(f) = k, t \text{ est sous-terme de } \varphi(f(v_1,\ldots,v_k)),\\ & \sigma \text{ substitution de } V\to Q\} \cup \{[q] \mid q \in Q\}, \end{aligned}$$

$Q_{\mathcal{B},+} := \{[q] \mid q \in Q_+\}$ et λ est l'ensemble des transitions de l'une des deux formes :

$$f([t_1\sigma],\ldots,[t_k\sigma])\to[f(t_1\sigma,\ldots,t_k\sigma)] \tag{3.6}$$

où les $[t_i\sigma]$ et $[f(t_1\sigma,\ldots,t_k\sigma)]$ appartiennent à $Q_{\mathcal{B}}$,

$$[t\sigma]\to[q] \tag{3.7}$$

où $[t\sigma],[q] \in Q_{\mathcal{B}}$, $t(v_1,\ldots,v_k) = \varphi(f(v_1,\ldots,v_k))$ et $f(v_1\sigma,\ldots,v_k\sigma) \xrightarrow{*}_{\mathcal{A}} q$. Les transitions de la forme (3.6) permettent à $\mathcal{B}$ de simuler la reconnaissance d'une image par φ d'un symbole $f \in F$; les transitions de la forme (3.7) permettent à $\mathcal{B}$ de simuler une transition de l'automate $\mathcal{A}$. On vérifie que

$$\mathrm{L}_{\mathcal{B}} = \varphi(\mathrm{L}_{\mathcal{A}}).$$

□.
Nous introduisons ci-dessous deux opérations sur les ensembles de termes qui généralisent les opérations de produit et d'étoile sur les ensembles de mots.

Définition 3.4.6 *Soit F un alphabet gradué et V un ensemble de variables. On définit, pour toute variable $v \in V$ et tout entier $n \in \mathbb{N}$, les opérations $\circ_v$, (n,v) et $*_v$ par : pour tous $T, T' \in \mathcal{P}(\mathrm{T}(F,V))$,*
1- $T \circ_v T' := T\{v \leftarrow T'\}$
2- $T^{(0,v)} := \{v\}, T^{(n+1,v)} := T \circ_v T^{(n,v)}$
3- $T^{*_v} := \bigcup_{n\geq 0}(T \cup \{v\})^{(n,v)}$

Autrement dit l'élévation à la puissance (n, v) consiste à itérer, n fois, le produit $\circ_v$. On remarquera que T^{*_v} n'est *pas* l'union des puissances (n, v)-ièmes de T, mais l'union des puissances (n, v)-ièmes de $T \cup \{v\}$.

Exemple 3.4.7 *Soient* $F = \{f, g, a\}$ *avec* $\mathbf{ar}(f) = \mathbf{ar}(g) = 2$,$\mathbf{ar}(a) = 0, V = \{v, w, \ldots\}$ *et* $T = \{f(a, v)\}$. *Alors*

$$\begin{aligned} T^{(2,v)} &= T \circ_v T = \{f(a, f(a, v))\}, \\ (T \cup \{v\})^{(2,v)} &= (T \cup \{v\}) \circ_v (T \cup \{v\}) = \{v, f(a, v), f(a, f(a, v))\}, \\ T^{(5,v)} &= \{f(a, f(a, f(a, f(a, f(a, v)))))\}, \end{aligned}$$

$$\begin{aligned} (T \cup \{v\})^{(5,v)} = \{&v, f(a, v), f(a, f(a, v)), f(a, f(a, f(a, v))), f(a, f(a, f(a, f(a, v)v))), \\ &f(a, f(a, f(a, f(a, f(a, v)))))\}. \end{aligned}$$

T^{*_v} est l'ensemble des termes de la forme représentée sur la figure 3.7, que l'on nomme parfois "peignes droits". Considérons maintenant $T_1 :=$

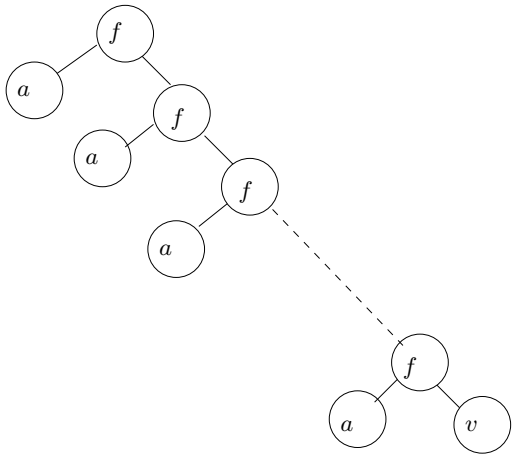


FIG. 3.7 – Un élément de T^{*_v}.

$\{f(v, v)\}$. *Alors*

$$\begin{aligned} T_1^{(2,v)} &= \{f(f(v, v), f(v, v))\}, \\ (T \cup \{v\})^{(2,v)} &= \{v, f(v, v), f(f(v, v), v), f(v, f(v, v)), f(f(v, v), f(v, v))\}, \\ T^{(*_v)} &= T(\{f\}, \{v\})\}. \end{aligned}$$

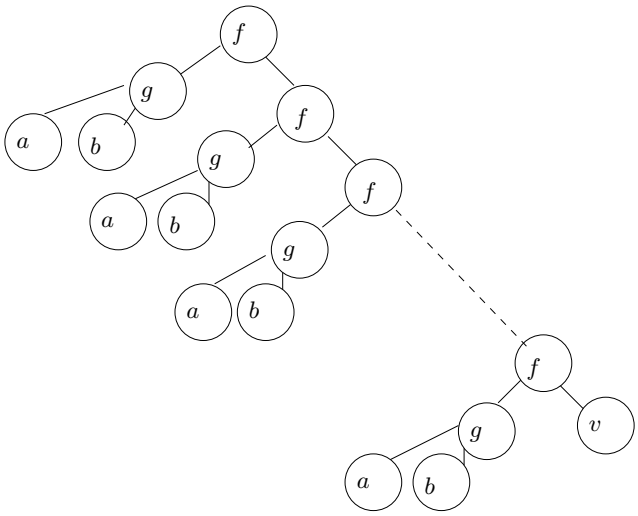


FIG. 3.8 – Un élément de R.

Le langage $R := \{f(w,v)\}^{*_v} \circ_w \{g(a,b)\}$ *est l'ensemble des termes de la forme représentée sur la figure 3.8.*

Théorème 3.4.8 *Soit* V *un ensemble de variables. La famille des langages reconnaissables de* $\mathrm{T}(F,V)$ *est close par les opérations* $\circ_v$ *et* $*_v$ *(pour tout* $v \in V$*).*

Preuve : Comme précédemment, nous prouvons le théorème 3.4.8 en fournissant, pour chaque opération sur les langages, une opération correspondante sur les automates.
Soient $\mathcal{A}_1 = \langle F \cup V, Q_1, Q_{1,+}, \delta_1 \rangle$, $\mathcal{A}_2 = \langle F \cup V, Q_2, Q_{2,+}, \delta_2 \rangle$ deux aft. On suppose que $Q_1 \cap Q_2 = \emptyset$.
Produit $\circ_v$:
Posons $\mathcal{B} = \langle F \cup V, Q_1 \cup Q_2, Q_{1,+}, \delta \rangle$ où

$$\delta := \delta_1 \cup \delta_2 \cup \{(q_2, q_1) \mid q_2 \in Q_{2,+} \text{ et } v \rightarrow_{\mathcal{A}_1} q_1\} - \{(v, q_1) \mid v \rightarrow_{\mathcal{A}_1} q_1\}.$$

Toute course de l'automate $\mathcal{B}$ est obtenue en substituant des courses de l'automate $\mathcal{A}_2$ dans une course de l'automate $\mathcal{A}_1$, puis en changeant chaque étiquette $q_2 \in Q_{2,+}$ de la racine de chaque course de $\mathcal{A}_2$ en une étiquette q_1, telle que $v \rightarrow_{\mathcal{A}_1} q_1$ (voir figure 3.9). On en déduit que $\mathrm{L}_{\mathcal{B}} = \mathrm{L}_{\mathcal{A}_1}\{v \leftarrow \mathrm{L}_{\mathcal{A}_2}\}$

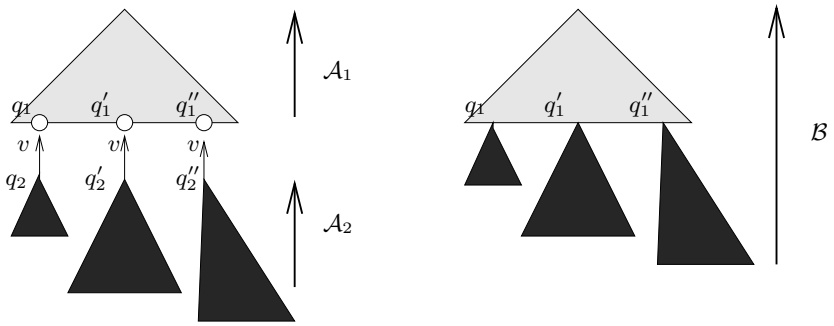


FIG. 3.9 – Une course de $\mathcal{B}$.

i.e. $\mathrm{L}_{\mathcal{B}} = \mathrm{L}_{\mathcal{A}_1} \circ_v \mathrm{L}_{\mathcal{A}_2}$.

Etoile $*_v$:

Soit $\mathcal{A} = \langle F \cup V, Q, Q_+, \delta \rangle$. Comme $\mathrm{L}_{\mathcal{A}}^{*_v} = (\mathrm{L}_{\mathcal{A}} \cup \{v\})^{*_v}$, on peut, sans perte de généralité, supposer que v est reconnu par $\mathcal{A}$. Posons $\mathcal{C} = \langle F \cup V, Q, Q_+, \lambda \rangle$ où

$$\lambda := \delta \cup \delta_* \text{ avec } \delta_* := \{q \to q' \mid q \in Q_+, q' \in Q, v \to_{\mathcal{A}} q'\}.$$

Toute course ρ de l'automate $\mathcal{C}$ sur un terme t est, soit une course de l'automate $\mathcal{A}$, soit est obtenue en substituant une course ρ_2 de l'automate $\mathcal{A}$ (sur un terme t_2 non-réduit à v) dans une course ρ_1 de l'automate $\mathcal{C}$ sur un terme t_1, en une feuille x telle que $t_1(x) = v$, puis en affectant à $\rho(x)$ une valeur q', telle que $v \to_{\mathcal{A}} q'$ (voir figure 3.10). Par récurrence sur la taille des

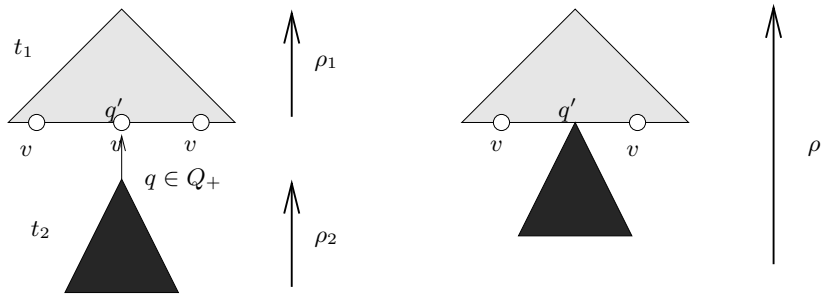


FIG. 3.10 – Une course de $\mathcal{C}$.

termes, il en résulte que

$$L_{\mathcal{C}} \subseteq L_{\mathcal{A}}^{*_v}.$$

Réciproquement, on peut montrer par récurrence sur la taille des termes que

$$L_{\mathcal{C}} \supseteq L^{*_v}_{\mathcal{A}}.$$

□

3.5 Automate minimal

Nous montrons ici que tout langage reconnaissable de termes admet un unique automate déterministe complet dont le nombre d'états est minimum et nous fournissons une méthode permettant de le calculer.

Définition 3.5.1 *Morphisme d'automates de termes. Soient* $\mathcal{A}_i = \langle F, Q_i, Q_{i,+}, \delta_i \rangle$ *(*$i \in \{1,2\}$*) des automates de termes. Un homomorphisme de* $\mathcal{A}_1$ *dans* $\mathcal{A}_2$ *est une application* $\varphi : Q_1 \to Q_2$ *vérifiant :*
(HA1) $\forall k \in \mathbb{N}, \forall f \in F_k, \forall q_0, \ldots, q_{k-1}, q \in Q_1,$

$$f(q_0, \ldots, q_{k-1}) \to_{\mathcal{A}_1} q \Rightarrow f(\varphi(q_0), \ldots, \varphi(q_{k-1})) \to_{\mathcal{A}_2} \varphi(q)$$

(HA2) $\forall q \in Q, \;\; q \in Q_{1,+} \Leftrightarrow \varphi(q) \in Q_{2,+}.$

Nous définissons une équivalence $\equiv$ sur $\mathrm{T}(F)$ par :

$$t \equiv t' \text{ ssi } (\forall C \in \mathcal{C}(F), C[t] \in L \Leftrightarrow C[t'] \in L).$$

Lemme 3.5.2 *L'équivalence* $\equiv$ *est une congruence de* F*-algèbre.*

La relation $\equiv$ s'appelle la *congruence syntaxique* du langage L. Soit $L \subseteq \mathrm{T}(F)$. Introduisons deux automates de termes qui reconnaissent L.

$$\mathcal{L} := \langle F, Q_{\mathcal{L}}, Q_{\mathcal{L},+}, \delta_{\mathcal{L}} \rangle$$

où
– $Q_{\mathcal{L}} := \mathrm{T}(F)$
– $Q_{\mathcal{L},+} := L$ est l'ensemble des états d'arrivée

– $\delta_{\mathcal{L}} := \{f(t_0, \ldots, t_k) \to \hat{f}(t_0, \ldots, t_k) \mid f \in F, \mathbf{ar}(f) = k, t_i \in \mathrm{T}(F)\}$

$$\mathcal{M} := \langle F, \mathrm{T}(F), \mathbb{Q}_+(L), \delta_{\mathcal{M}} \rangle$$

où
– $Q_{\mathcal{M}} := \{[t]_{\equiv} \mid t \in \mathrm{T}(F)\}$
– $Q_{\mathcal{M},+} := \{[t]_{\equiv} \mid t \in L\}$
– $\delta_{\mathcal{M}} := \{f([t_0]_{\equiv}, \ldots, [t_k]_{\equiv}) \to [f(t_0, \ldots, t_k)]_{\equiv} \mid f \in F, \mathbf{ar}(f) = k, t_i \in \mathrm{T}(F)\}$
Dans le cas de $\mathcal{L}$: pour tout terme $t \in \mathrm{T}(F), q \in Q_{\mathcal{L}}$,

$$t \xrightarrow{*} q \Leftrightarrow q = t$$

donc

$$(\exists q \in Q_{\mathcal{L},+} \mid t \xrightarrow{*} q) \Leftrightarrow (t \in L)$$

i.e.

$$\mathrm{L}_{\mathcal{L}} = L.$$

Dans le cas de $\mathcal{M}$: soit $\pi : \mathrm{T}(F) \to \mathrm{T}(F)/\equiv$ la projection i.e. pour tout terme $t \in \mathrm{T}(F)$,$\pi(t) := [t]_{\equiv}$. Comme $\equiv$ est une congruence de F-algèbre, π est un homomorphisme de F-algèbre. D'autre part,

$$\begin{aligned} \pi(t) \in Q_{\mathcal{M},+} &\Leftrightarrow \exists t' \in L, \pi(t) = \pi(t') \\ &\Leftrightarrow t \in L \\ &\Leftrightarrow t \in Q_{\mathcal{L},+} \end{aligned}$$

Donc π est un homomorphisme d'automates, ce qui entraine que $\mathrm{L}_{\mathcal{M}} = \mathrm{L}_{\mathcal{L}} = L$.

Théorème 3.5.3 *Soit $L \subseteq \mathrm{T}(F)$ et soit $\mathcal{A}$ un automate déterministe complet. Les propriétés suivantes sont équivalentes :*
(1) $L = \mathrm{L}_{\mathcal{A}}$
(2) Il existe un homomorphisme d'automates $\varphi : \mathcal{L} \to \mathcal{A}$
(3) Il existe un homomorphisme d'automates $\psi : \mathcal{A} \to \mathcal{M}$

Preuve : L'implication (2) ⇒ (1) résulte du fait que, s'il existe un homomorphisme d'automates de $\mathcal{L}$ vers $\mathcal{A}$ alors $\mathrm{L}_{\mathcal{L}} = \mathrm{L}_{\mathcal{A}}$ et du fait que $\mathrm{L}_{\mathcal{L}} = L$. De même, (3) ⇒ (1) résulte du fait que $\mathrm{L}_{\mathcal{M}} = L$.
(1) ⇒ (2)
Supposons (1). Pour tout $t \in \mathrm{T}(F)$, définissons $\varphi(t)$ comme l'unique état

$q \in Q_{\mathcal{A}}$ tel que $t \xrightarrow{*}_{\mathcal{A}} q$ et vérifions que c'est un homomorphisme d'automates.
Soit $f \in F, \mathbf{ar}(f) = k, t_i \in \mathrm{T}(F)(i \in [0, k-1])$. Par définition de φ,

$$t_i \xrightarrow{*}_{\mathcal{A}} \varphi(t_i)$$

et si

$$f(\varphi(t_0), \ldots, \varphi(t_{k-1}) \rightarrow_{\mathcal{A}} q, \tag{3.8}$$

par définition de la réduction de l'automate $\mathcal{A}$ on a :

$$f(t_0, \ldots, t_{k-1}) \xrightarrow{*}_{\mathcal{A}} q$$

donc

$$\varphi(f(t_0, \ldots, t_{k-1}) = q.$$

La transition (3.8) s'écrit donc

$$f(\varphi(t_0), \ldots, \varphi(t_{k-1}) \rightarrow_{\mathcal{A}} \varphi(f(t_0, \ldots, t_{k-1})$$

ce qui prouve la condition $(HA1)$. D'autre part $t \in L$ ssi t est reconnu par $\mathcal{A}$ ssi $\varphi(t) \in Q_{\mathcal{A},+}$, ce qui prouve la condition $(HA2)$.
(1) $\Rightarrow$ (3)
Supposons (1). Considérons les homorphismes $\pi : \mathcal{L} \rightarrow \mathcal{M}$ et $\varphi : \mathcal{L} \rightarrow \mathcal{A}$. Remarquons que, pour tous $t, t' \in \mathrm{T}(F)$, si $\varphi(t) = \varphi(t')$, alors , pour tout contexte C, $\forall q \in Q_{\mathcal{A}}, C[t] \xrightarrow{*}_{\mathcal{A}} q \Leftrightarrow C[t'] \xrightarrow{*}_{\mathcal{A}} q$, donc $C[t] \in L \Leftrightarrow C[t'] \in L$. Donc

$$(\mathrm{Ker}\varphi) \subseteq (\mathrm{Ker}\pi). \tag{3.9}$$

On définit alors une application $\bar{\pi} : Q_{\mathcal{A}} \rightarrow \mathrm{T}(F)/\equiv$ par

$$\bar{\pi}(\varphi(t)) = \pi(t).$$

Cette égalité définit bien une application $\bar{\pi}$ car tout état $q \in Q_{\mathcal{A}}$ est de la forme $\varphi(t)$ et, d'autre part, d'après (3.9), le membre droit $\pi(t)$ ne dépend pas du terme t choisi parmi tous les termes dont l'image par φ est égale à q. En utilisant le fait que φ et π sont des homomorphismes, on prouve aisément que $\bar{\pi}$ est, lui-aussi, un homomorphisme. □

Corollaire 3.5.4 *Soit $L \subseteq \mathrm{T}(F)$. Le langage L est reconnaissable ssi sa congruence syntaxique $\equiv_L$ a un index fini.*

Corollaire 3.5.5 *Soit $L \subseteq \mathrm{T}(F)$ un langage reconnaissable. Alors*
1- $\mathcal{M}$ est l'automate fini déterministe complet qui a le nombre minimum d'états, parmi les aftd complets qui reconnaissent L.
2- Si $\mathcal{M}'$ est un aftd complet qui reconnait L et qui a le même nombre d'états que $\mathcal{M}$, alors $\mathcal{M} \sim \mathcal{M}'$.

Corollaire 3.5.6 *Soit $\mathcal{A}$ un automate (de termes) fini déterministe complet. On peut calculer l'automate minimal du langage $\mathrm{L}_{\mathcal{A}}$.*

3.6 Expressions rationnelles

Définition 3.6.1 *Soit F un alphabet gradué et V un ensemble dénombrable de variables (disjoint de F). L'ensemble des parties rationnelles de $\mathrm{T}(F,V)$ est le plus petit élément $\mathcal{R}$ de $\mathcal{P}(\mathcal{P}(\mathrm{T}(F,V)))$, vérifiant :*
(i) pour tout terme $t \in \mathrm{T}(F,V)$, $\{t\} \in \mathcal{R}$
(ii) $\emptyset \in \mathcal{R}$ et $\mathcal{R}$ est clos par union finie,
(iii) $\mathcal{R}$ est clos par toutes les opérations $\circ_v$ ($v \in V$),
(iv) $\mathcal{R}$ est clos par toutes les opérations $*_v$ ($v \in V$).

On note $RAT(\mathrm{T}(F,V))$ l'ensemble des parties rationnelles de $\mathrm{T}(F,V)$. Un langage $L \subseteq \mathrm{T}(F,V)$ est dit rationnel ssi il appartient à $RAT(\mathrm{T}(F,V))$. Plus concrètement : L est rationnel ssi il existe une expression arithmétique correctement formée, dont les opérandes sont des singletons et dont les opérations sont $\cup, \circ_v, *_v$. On appelle *expression rationnelle* une telle expression. Bien sûr, dans une expression rationnelle, seul un nombre fini d'opérations peut apparaître.

Théorème 3.6.2 *Soit $L \subseteq \mathrm{T}(F)$. Le langage L est reconnaissable ssi il est rationnel.*

Preuve : 1- Tout singleton est reconnaissable, d'après le théorème 3.4.3, $REC(\mathrm{T}(F,V)))$ possède le langage vide, est clos par union et d'après le théorème 3.4.8, $REC(\mathrm{T}(F,V)))$ est clos par les opérations $\circ_v, *_v$. On en

conclut que

$$RAT(\mathrm{T}(F,V)) \subseteq REC(\mathrm{T}(F,V)).$$

2- Montrons que tout langage reconnaissable de termes est rationnel. Soit $\mathcal{A} = \langle F, Q, Q_+, \delta\rangle$ un aft. Supposons que $F \cap Q = \emptyset$. Nous avons défini une notion de course de $\mathcal{A}$ sur un terme de $\mathrm{T}(F)$. Nous étendons cette notion aux termes de $\mathrm{T}(F,Q)$ en appelant E-*course* de $\mathcal{A}$ sur un terme $t \in \mathrm{T}(F,Q)$, toute application $\rho : \mathrm{dom}(t) \to Q$ telle que, $\forall u \in \mathrm{dom}(t)$:

$$\text{si } \mathbf{ar}(t(u)) = k \geq 1 \quad \text{alors } t(u)(\rho(u0, \ldots, \rho(u\cdot(k-1))) \to \rho(u) \in \delta \quad (3.10)$$

et

$$\text{si } \mathbf{ar}(t(u)) = 0 \quad \text{alors } t(u) \to \rho(u) \in \delta \text{ ou } t(u) = \rho(u). \quad (3.11)$$

On remarquera que, aucune règle de δ n'a un membre gauche appartenant à Q ; donc, si ρ est une E-course sur t et u est une feuille de t étiquetée sur Q, alors $\rho(u) = t(u)$. Un terme $t \in \mathrm{T}(F,Q)$ est dit E-accepté (ou reconnu) par $\mathcal{A}$, ssi il existe un E-calcul ρ de $\mathcal{A}$ sur t, tel que $t(\varepsilon) \in Q_+$.

Soit ρ un E-calcul de $\mathcal{A}$. On appelle *intérieur* de ρ, et l'on note $I(\rho)$, l'ensemble d'états :

$$I(\rho) := \{q \in Q \mid \exists u \in \mathrm{dom}(\rho), u \neq \varepsilon, u0 \in \mathrm{dom}(\rho), \rho(u) = q\}.$$

Pour toute partie $R \subseteq Q$ et tout $q \in Q$, on note

$$\mathrm{L}_q^R := \{t \in \mathrm{T}(F,Q) \mid \exists \rho : \mathrm{dom}(t) \to Q, \rho \text{ est une } E-\text{course de } \mathcal{A}, \rho(\varepsilon) = q \text{ et } I(\rho) \subseteq R\}.$$

Nous prouvons, par induction sur $|R|$, que, pour tout $R \subseteq Q$ et tout $q \in Q$, le langage L_q^R est une partie rationnelle de $\mathrm{T}(F,Q)$.
Base : $R = \emptyset$.
Alors $\mathrm{L}_q^\emptyset = \{t \in \mathrm{T}(F,Q) \mid \mathrm{h(t)} \leq 1, \mathrm{t} \xrightarrow{*}_{\mathcal{A}} \mathrm{q}\}$, qui est fini, donc rationnel (clauses (i-ii)).
Étape d'induction : Soit $R \subseteq Q, r \in Q - R, q \in Q$.

$$\mathrm{L}_q^{R\cup\{r\}} = \mathrm{L}_q^R \cup (\mathrm{L}_q^R \circ_r (\mathrm{L}_r^R)^{*_r}).$$

Revenons maintenant au langage $\mathrm{L}_\mathcal{A}$. Pour tout $q \in Q$ posons

$$T_q := \{f \in F \mid \mathbf{ar}(f) = 0, \quad f \to_\mathcal{A} q\}.$$

Numŕotons les états : $Q = \{q_1, \ldots, q_n\}$. On a alors

$$\mathrm{L}_{\mathcal{A}} = \bigcup_{q \in Q_+} \mathrm{L}_q^Q \circ_{q_1} T_{q_1} \circ_{q_2} T_{q_2} \ldots \circ_{q_n} T_{q_n}. \tag{3.12}$$

Comme les langages L_q^Q sont rationnels et les langages T_{q_j} sont finis, la formule (3.12) montre que $\mathrm{L}_{\mathcal{A}}$ est rationnel. □

Chapitre 4

Logique Monadique du Second Ordre

4.1 Syntaxe et sémantique

Les formules Soit $\mathcal{S} = \{r_1, \ldots, r_n\}$ une signature consistant en n symboles de relation munis chacun d'une arité : le symbole r_i a une arité $(\rho_i, \tau_i) \in \mathbb{N}^2$.

Soient $\mathcal{V}_1 = \{x, y, z, \ldots, \}, \mathcal{V}_2 = \{X, Y, Z \ldots\}$ deux ensembles de variables : les variables d'ordre 1 et d'ordre 2, respectivement. L'ensemble des formules de logique Monadique du Second-Ordre sur $\mathcal{S}$ est défini inductivement par :

- pour tout $x \in \mathcal{V}_1, Y, X \in \mathcal{V}_2$,

$$x \in X, \quad Y \subseteq X$$

  sont des formules MSO
- pour tout $r \in \mathcal{S}$ d'arité (ρ, τ) et tous $x_1, \ldots x_\rho \in \mathcal{V}_1$, $X_1 \ldots X_\tau \in \mathcal{V}_2$

$$r(x_1, \ldots x_\rho, X_1 \ldots X_\tau)$$

  est une formule MSO
- si Φ, Ψ sont des formules MSO alors

$$\neg\Phi, \ \ \Phi \vee \Psi, \ \ \Phi \wedge \Psi, \ \ \Phi \rightarrow \Psi, \ \ \Phi \leftrightarrow \Psi$$

sont des formules MSO.

- si Φ est une formule MSO et $x \in \mathcal{V}_1, X \in \mathcal{V}_2$ alors

$$\exists x.\Phi, \ \ \forall x.\Phi, \ \ \exists X.\Phi, \ \ \forall X.\Phi$$

sont des formules MSO.

Les interprétations Soit $\mathcal{S}$ une signature relationnelle. On appelle structure sur la signature $\mathcal{S}$ un t-uple

$$\mathcal{M} = \langle D^{(\mathcal{M})}, r_1^{(\mathcal{M})} \ldots, r_n^{(\mathcal{M})} \rangle$$

où $D^{(\mathcal{M})}$ est un ensemble (le domaine de la structure), et pour tout $i \in [1, n]$, $r_i^{(\mathcal{M})} \subseteq (D^{(\mathcal{M})})^\rho \times (\mathcal{P}(D^{(\mathcal{M})}))^\tau$.

Exemple 4.1.1 *Considérons la signature $\mathcal{S} := \{P, E\}$ avec P d' arité $(3, 0)$ et E d' arité $(2, 0)$. Nous donnons ci-dessous deux structures sur cette signature.*

$$\mathcal{N} = \langle D^{(\mathcal{N})}, P^{(\mathcal{N})}, E^{(\mathcal{N})} \rangle$$

est définie par

$$D^{(\mathcal{N})} := \mathbb{N}, \ \ P^{(\mathcal{N})} := \{(x, y, z) \in \mathbb{N}^3 \mid x{+}y = z\}, \ \ E^{(\mathcal{N})} := \{(x, y) \in \mathbb{N}^2 \mid x = y\}$$

$$\mathcal{M} = \langle D^{(\mathcal{M})}, P^{(\mathcal{M})}, E^{(\mathcal{M})} \rangle$$

est définie par

$$\begin{array}{rcl} D^{(\mathcal{M})} & := & \{0, 1, 2, 3, 4\}; \ P^{(\mathcal{M})} := \{(x, y, z) \in \{0, 1, 2, 3, 4\}^3 \mid x \oplus y = z\}, \\ E^{(\mathcal{N})} & := & \{(x, y) \in \{0, 1, 2, 3, 4\}^2 \mid x = y\} \end{array}$$

où la loi de composition $\oplus$ est définie par la table

$\oplus$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	2
2	2	3	4	2	3
3	3	4	2	3	4
4	4	2	3	4	2

Une *valuation* sur la structure $\mathcal{M}$ est une application $\nu : (\mathcal{V}_1 \cup \mathcal{V}_2) \to D_{\mathcal{M}} \cup \mathcal{P}(D_{\mathcal{M}})$ telle que, pour tout $x \in \mathcal{V}_1$, $\nu(x) \in D_{\mathcal{M}}$ et pour tout $X \in \mathcal{V}_2, \nu(X) \in \mathcal{P}(D_{\mathcal{M}})$.

La valeur de vérité d'une formule MSO dans une structure $\mathcal{M}$ pour une valuation ν est obtenue, intuitivement, en attribuant aux formules atomiques la valeur 1 (resp. 0), si la formule exprime un fait vrai (resp. faux) dans $\mathcal{M}$, puis en interprétant les connecteurs et les quantificateurs selon l'usage courant en mathématique. Plus formellement, on introduit le symbole $\models$ et on définit, pour tout couple $(\mathcal{M}, \nu)$ formé d'une structure sur la signature $\mathcal{S}$ et d'une valuation sur cette structure et toute formule MSO Φ sur la signature $\mathcal{S}$, la validité de

$$(\mathcal{M}, \nu) \models \Phi.$$

La définition procède par induction structurelle sur Φ.
Formules atomiques :

$$(\mathcal{M}, \nu) \models x \in X$$

ssi $\nu(x) \in \nu(X)$.

$$(\mathcal{M}, \nu) \models Y \subseteq X$$

ssi $\nu(Y) \subseteq \nu(X)$.

$$(\mathcal{M}, \nu) \models r(x_1, \dots x_\rho, X_1 \dots X_\tau)$$

ssi $(\nu(x_1), \dots \nu(x_\rho), \nu(X_1) \dots \nu(X_\tau)) \in r^{(\mathcal{M})}$.
Connecteurs :

$$(\mathcal{M}, \nu) \models \neg\Phi$$

ssi, la relation $(\mathcal{M}, \nu) \models \Phi$ n'est pas vérifiée.

$$(\mathcal{M}, \nu) \models \Phi \vee \Psi$$

ssi, $((\mathcal{M}, \nu) \models \Phi)$ ou $((\mathcal{M}, \nu) \models \Psi)$.
etc...
Quantificateurs :

$$(\mathcal{M}, \nu) \models \exists x.\Phi$$

ssi, il existe un élément $d \in D^{(\mathcal{M})}$, tel que la relation $(\mathcal{M}, \nu') \models \Phi$ est vérifiée par la valuation ν' définie par $\forall y \in \mathcal{V}_1 - \{x\}, \nu'(y) := \nu(y),\ \ \nu'(x) := d,\ \ \forall X \in \mathcal{V}_2, \nu'(X) := \nu(X)$.

$$(\mathcal{M}, \nu) \models \forall x.\Phi$$

ssi, pour tout élément $d \in D^{(\mathcal{M})}$, la relation $(\mathcal{M}, \nu') \models \Phi$ est vérifiée par la valuation ν' définie par $\forall y \in \mathcal{V}_1 - \{x\}, \nu'(y) := \nu(y),\ \ \nu'(x) := d,\ \ \forall X \in \mathcal{V}_2, \nu'(X) := \nu(X)$.
Des clauses similaires définissent "$(\mathcal{M}, \nu) \models \exists X.\Phi$" et "$(\mathcal{M}, \nu) \models \forall X.\Phi$".

Exemple 4.1.2 *(on continue l'exemple 4.1.1). Examinons la formule*

$$\Phi := \forall x.\forall y.\forall y.\forall t.(P(x, y, t) \wedge P(x, z, t)) \rightarrow E(y, z).$$

Cette formule est vraie dans $\mathcal{N}$ *car la loi d'addition est simplifiable à gauche. (i.e.* $x+y = x+z \Rightarrow y = z$*). Par contre,* Φ *est fausse dans* $\mathcal{M}$ *car* $2 \oplus 0 = 2 \oplus 3$ *mais* $0 \neq 3$*. Autrement écrit :*

$$\mathcal{N} \models \Phi$$

mais par contre,

$$\mathcal{M} \not\models \Phi.$$

Examinons maintenant la formule

$$\Psi := \forall y \cdot P(x, y, y),$$

et les valuations suivantes dans $\mathcal{N}$ *:*

$$\nu(x) := 0,\ \ \forall v \in \mathcal{V}_1 - \{x\}, \nu(v) := 3;\ \ \forall X \in \mathcal{V}_2, \nu(X) := \{2, 5\}$$

$$\mu(x) := 1,\ \ \forall v \in \mathcal{V}_1 - \{x\}, \mu(v) := 7;\ \ \forall X \in \mathcal{V}_2, \mu(X) := \{1, 2, 39\}$$

On vérifie que :

$$(\mathcal{N}, \nu) \models \Psi,\ \ (\mathcal{N}, \mu) \not\models \Psi.$$

Remarquons que seule la valeur $\nu(x)$ *(resp.* $\mu(x)$*) influe sur la validité de* Ψ*, car c'est la seule variable* libre *(i.e. qui apparaît sans être l'objet d'une quantification) de* Ψ*.*

4.2 Ensembles de mots définissables en MSO

Étant donné un alphabet A, nous lui associons une signature MSO $\mathcal{S}_A$ comme suit :

$$\mathcal{S}_A := \{P_a \mid a \in A\} \cup \{S\}$$

où chaque P_a est d'arité $(1,0)$ et S est d'arité $(2,0)$. Un mot $w \in A^*$ peut "naturellement" être vu comme une structure sur la signature $\mathcal{S}_A$ en posant

$$D^{(w)} := \{0, 1, \ldots |w| - 1\}$$

et pour tout $i, j \in D^{(w)}$,

$$P_a^{(w)} := \{i \in D^{(w)} \mid w[i] = a\}$$

$$S^{(w)} := \{(i,j) \in D^{(w)} \times D^{(w)} \mid i + 1 = j\}.$$

Chaque formule MSO, close, Φ sur la signature $\mathcal{S}_A$ définit alors le langage des mots qui sont des modèles de Φ :

$$\mathrm{L}(\Phi) := \{w \in A^* \mid w \models \Phi\}.$$

Un langage $L \subseteq A^*$ est dit MSO-définissable ssi, il existe une formule MSO Φ sur la signature $\mathcal{S}_A$ telle que $L = \mathrm{L}(\Phi)$.

Théorème 4.2.1 (Büchi,1960) *Soit A un alphabet fini et $\mathcal{S}_A$ la signature MSO associée. Un langage $L \subseteq A^*$ est rationnel ssi il est MSO-définissable.*

Partie 1 : Montrons que tout rationnel est définissable par une formule MSO, close.
Soit L un langage rationnel de A^* (nous supposerons, dans un premier temps que $\varepsilon \notin L$). Soit $\mathcal{A} = \langle A, Q, I, F, \delta \rangle$ un automate fini reconnaissant L (ici $I \subseteq Q$ est l'ensemble des états initiaux et F est l'ensemble des états finaux). Construisons une formule MSO Φ qui exprime l'existence d'un calcul réussi de $\mathcal{A}$.
Notons $Q = \{q_1, q_2, \ldots, q_n\}$. L'idée est de coder un calcul

$$p_0 \xrightarrow{a_0} p_1 \cdots p_k \xrightarrow{a_k} p_{k+1} \cdots p_\ell$$

de $\mathcal{A}$ sur le mot $w = a_0 \cdots a_k \cdots a_{\ell-1}$ (avec $\ell \geq 1$) par le nouveau mot

$$(p_0, a_0) \cdots (p_k, a_k) \cdots (p_{\ell-1}, a_{\ell-1}) \tag{4.1}$$

et d'exprimer l'existence de cet étiquetage par une formule MSO. Introduisons une famille de variables d'ordre 2, $X_1, \ldots, X_n$ (X_i représente l'ensemble des positions k étiquetées par (q_i, a_k), où a_k est une lettre quelconque de A, dans le "mot-calcul" (4.1)). Considérons les prédicats auxiliaires :

- $Prem(x)$: x est l'entier 0.

- $Der(x)$: x est l'entier maximum de dom(w).

- $Emptydom$: le domaine est l'ensemble vide.

- $Empty(X)$: X est l'ensemble vide.

- $Emptyinter(X, Y)$: $X \cap Y$ est vide.

- $Part(X_1, \ldots, X_n)$: les ensembles $X_1, \ldots, X_n$ sont disjoints et ont une union égale à dom(w).

- $Initial(x)$: x appartient à un ensemble X_i avec $q_i \in I$.
- $Final(x)$: x appartient à un ensemble X_i et il existe une transition (q_i, a, q_j) avec $P_a(x)$ et $q_j \in F$.
- $Trans(x, y)$: il existe $(q_i, a, q_j) \in \delta$ tel que $x \in X_i, P_a(x), y \in X_j$.

Ces prédicats sont exprimables par les formules MSO suivantes :
- $Prem(x)$ est exprimé par : $\forall y, \neg S(y, x)$

- $Der(x) : \forall y \cdot \neg S(x, y)$

- $Emptydom : \forall X \cdot \forall Y \cdot X \subseteq Y$

- $Empty(X) : \forall Y \cdot X \subseteq Y$.

- $Emptyinter(X, Y) : \forall Z \cdot \forall x \cdot (x \in X \wedge x \in Y) \rightarrow x \in Z$.

- $Part(X_1, \ldots, X_n) : \bigwedge_{1 \leq i < j \leq n} Emptyinter(X_i, X_j) \wedge (\forall x, \bigvee_{1 \leq i \leq n} x \in X_i)$
- $Init(x) : \bigvee_{q_i \in I} x \in X_i$.
- $Final(x) : \bigvee_{\substack{(q_i, a, q_j) \in \delta \\ q_j \in F}} x \in X_i \wedge P_a(x)$.

– $Trans(x,y) : \bigvee_{(q_i,a,q_j)\in\delta}(x \in X_i \wedge P_a(x) \wedge y \in X_j)$.
Si $\varepsilon \notin L$, on pose alors :

$$\begin{array}{lll} \Phi & := & \exists X_1 \cdot \exists X_2 \cdot \ldots \cdot \exists X_n \cdot \forall x \cdot \forall y \cdot \\ & & Part(X_1,\ldots,X_n) \wedge (Prem(x) \rightarrow Init(x)) \wedge (Der(x) \rightarrow Final(x)) \\ & \wedge & (S(x,y) \rightarrow Trans(x,y)) \\ \wedge \neg Emptydom & & \end{array} \tag{4.2}$$

Si $\varepsilon \in L$, on omet la dernière ligne dans la définition (4.2) de Φ.
Partie 2 :
Montrons que tout langage défini par une formule MSO2 Φ, close, est rationnel.
Étape 1 : Afin de rendre cette preuve plus aisée, on se ramène à des formules sur une signature modifiée, où tous les symboles de relations ont des arguments d'ordre 2 (i.e. des variables qui représentent des ensembles).
Définissons la signature MSO $\mathcal{S}_{A,2}$ par :

$$\mathcal{S}_{A,2} := \{P_a \mid a \in A\} \cup \{S\}$$

où chaque P_a est d'arité $(0,1)$ et S est d'arité $(0,2)$. La signification de P_a, S sur un mot $w \in A^*$ et une valuation ν est donnée par :

$$P_a^{(w)} := \{P \in \mathcal{P}(D^{(w)}) \mid \forall i \in P, w[i] = a\}$$

$$S^{(w)} := \{(P,Q) \mid P,Q \in \mathcal{P}(D^{(w)}), \forall i \in P, \forall j \in Q, i+1 = j\}$$

Appelons logique MSO2, sur la signature $\mathcal{S}_{A,2}$, la logique MSO sur la signature $\mathcal{S}_{A,2}$, que l'on *restreint* aux formules qui ne comportent que des variables de $\mathcal{V}_2$.

Lemme 4.2.2 *Toute formule MSO sur $\mathcal{S}_A$ est équivalente à une formule MSO2 sur $\mathcal{S}_{A,2}$.*

(une formule est dite équivalente à une autre si elle a exactement les mêmes modèles).
Preuve : On peut tout d'abord restreindre les connecteurs à $\neg, \vee$ et les quantificateurs à $\exists$. On associe ensuite à chaque variable $x \in \mathcal{V}_1$ une nouvelle variable $Z_x \in \mathcal{V}_2$. Etant donnée Φ, formule MSO sur $\mathcal{S}_A$, on définit une formule MSO2 $\hat{\Phi}$ sur $\mathcal{S}_{A,2}$, par induction structurelle :

- si $\Phi = X \subseteq Y$ alors $\hat{\Phi} = \Phi$
- si $\Phi = S(x, y)$ alors $\hat{\Phi} = Sing(Z_x) \wedge Sing(Z_y) \wedge S(Z_x, Z_y)$
- si Φ, Ψ sont des formules MSO alors $\hat{\neg\Phi} = \neg\hat{\Phi}$, $\hat{\Phi \vee \Psi} = \hat{\Phi} \vee \hat{\Psi}$.
- si Φ est une formule MSO et $x \in \mathcal{V}_1$ alors $\hat{\exists x.\Phi} = \exists Z_x \cdot Sing(Z_x) \wedge \hat{\Phi}$
- si Φ est une formule MSO et $X \in \mathcal{V}_2$ alors $\hat{\exists X.\Phi} = \exists X \cdot \hat{\Phi}$

où $Sing(X)$ est la formule suivante, qui exprime que X est un singleton :

$$\begin{array}{rcl} Sing(X) & := & \exists Y \cdot \neg(X \subseteq Y) \\ & \wedge & (\forall Y' \cdot (Y' \subseteq X) \rightarrow [(Y' \subseteq Y) \vee (X \subseteq Y')]) \end{array}$$

□

Étape 2 : On souhaite démontrer que tout langage défini par une formule MSO2 Φ, close, est rationnel. On envisage de procéder par induction structurelle sur Φ, ce qui nous amène à associer un langage à *toute* formule Φ, même lorsqu'elle comporte des variables libres.
Soit $n \in \mathbb{N}, w \in (\{0,1\}^n \times A)^*$ et $\overrightarrow{X} = (X_1, \ldots, X_n) \in \mathcal{V}_2^n$. On associe au mot w la valuation :

$$\nu_w : X_i \mapsto \{k \in dom(w) \mid \pi_i(w[k]) = 1\}, \quad X \in \mathcal{V}_2 - \{X_1, \ldots, X_n\} \mapsto \emptyset.$$

On peut alors associer à chaque formule Φ et vecteur $\overrightarrow{X}$ de variables, le langage de ses modèles :

$$\mathrm{L}(\Phi, \overrightarrow{X}) := \{w \in (\{0,1\}^n \times A)^* \mid (\pi_{n+1}(w), \nu_w) \models \Phi\}$$

Exemple 4.2.3 *Prenons $\Phi := X \subseteq Y$, $n := 2$, $\overrightarrow{X} := (X, Y)$, $A := \{a, b, c\}$ et considérons les deux mots de A^* :*

$w := (0,0,a)(1,1,b)(0,1,a)(0,1,b)(1,1,c)$ $w' := (0,0,a)(1,0,b)(0,1,a)(0,1,b)(1,1,c)(0,0,c)$.

On vérifie que $w \models \Phi$ tandis que $w \not\models \Phi$. Par conséquent $w \in \mathrm{L}(\Phi, \overrightarrow{X})$ et $w' \notin \mathrm{L}(\Phi, \overrightarrow{X})$.

Lemme 4.2.4 *Pour toute formule MSO2 Φ (sur $\mathcal{S}_{A,2}$) et tout vecteur $\overrightarrow{X}$ contenant les variables de Φ, le langage $\mathrm{L}(\Phi, \overrightarrow{X})$ est rationnel.*

Preuve : On prouve ce lemme par induction structurelle sur Φ.

- $\Phi = X \subseteq Y$ et $\overrightarrow{X} = (X, Y, X_3, \ldots, X_n)$.
  Considérons l'homomorphisme $h : (\{0,1\}^n \times A)^* \to (\{0,1\}^2)^*$ défini par $h(b_1, b_2, \ldots, b_n, a) = (b_1, b_2)$. On a

$$\mathrm{L}(\Phi, \overrightarrow{X}) = h^{-1}(\{(0,0),(0,1),(1,1)\}^*)$$

  qui est bien rationnel (car les langages rationnels sont clos par homomorphisme inverse).
- $\Phi = S(X, Y)$ et $\overrightarrow{X} = (X, Y, X_3, \ldots, X_n)$.
  Posons $B_1 := \{(0,0),(0,1)\}, B_2 := \{(0,0),(1,0)\}$.

$$\mathrm{L}(\Phi, \overrightarrow{X}) = h^{-1}(B_1^* \cup B_2^* \cup \{(0,0)\}^* \cdot \{(1,0)(0,1) \cdot \{(0,0)\}^*)$$

  qui est bien rationnel (car les langages rationnels sont clos par homomorphisme inverse).
- $\Phi = \neg\Psi$

$$\mathrm{L}(\Phi, \overrightarrow{X}) = (\{0,1\}^n \times A)^* - \mathrm{L}(\Psi, \overrightarrow{X}),$$

  qui est bien rationnel car, par hypothèse de récurrence, $\mathrm{L}(\Psi, \overrightarrow{X})$ est rationnel et les langages rationnels sont clos par complémentaire).
- $\Phi = \Psi_1 \vee \Psi_2$

$$\mathrm{L}(\Phi, \overrightarrow{X}) = \mathrm{L}(\Psi_1, \overrightarrow{X}) \cup \mathrm{L}(\Psi_2, \overrightarrow{X}).$$

  Comme Ψ_1, Ψ_2 sont des sous-formules de Φ, $\overrightarrow{X}$ contient bien toutes les variables de Ψ_1 (resp. Ψ_2) et, par hypothèse de récurrence, les langages $\mathrm{L}(\Psi_i, \overrightarrow{X})$ sont rationnels. La formule ci-dessus entraîne alors que $\mathrm{L}(\Phi, \overrightarrow{X})$ est rationnel (car les langages rationnels sont clos par union).
- $\Phi = \exists X \cdot \Psi$ et $\overrightarrow{X} = (X, X_2, \ldots, X_n)$
  Considérons l'homomorphisme $\pi : (\{0,1\}^n \times A)^* \to (\{0,1\}^{n-1} \times A)^*$ défini par $\pi(b_1, b_2, \ldots, b_n, a) = (b_2, \ldots, b_n, a)$.

$$\mathrm{L}(\Phi, \overrightarrow{X}) = \pi^{-1}(\pi(\mathrm{L}(\Psi, \overrightarrow{X}))),$$

  donc $\mathrm{L}(\Phi, \overrightarrow{X})$ est rationnel (car les langages rationnels sont clos par homomorphismes directs et inverses).

□

Achevons la partie 2 : Soit Φ une formule close. Soit $\overrightarrow{X} = (X_1, \ldots, X_n)$ un vecteur de variables contenant l'ensemble des variables de Φ. On a établi plus haut que $\mathrm{L}(\Phi, \overrightarrow{X})$ est rationnel. Comme Φ est close, pour tout mot $w \in A^*$,

$w \models \Phi$ si et seulement si il existe une valuation ν telle que $(w, \nu) \models \Phi$ si et seulement si $\exists w' \in (\{0,1\}^n \times A)^* \mid (\pi_{n+1}(w'), \nu_{w'}) \models \Phi$ et $\pi_{n+1}(w') = w$. Donc

$$\mathrm{L}(\Phi) = \pi_{n+1}(\mathrm{L}(\Phi, \overrightarrow{X})),$$

et comme les langages rationnels sont clos par homomorphisme, $\mathrm{L}(\Phi)$ est rationnel.
Preuve du théorème 4.2.1 :
la partie 1 montre que tout rationnel de mots est définissable en logique MSO et la partie 2 montre que tout ensemble de mots définissable en logique MSO est rationnel.□

4.3 Ensembles de termes definissables en MSO

Étant donné un alphabet gradué F, nous lui associons une signature MSO $\mathcal{S}_F$ comme suit :

$$\mathcal{S}_F := \{P_f \mid f \in F\} \cup \{S_k \mid 0 \leq k \leq K-1\}$$

où $K := \max\{\mathrm{ar}(f) \mid f \in F\}$. Un terme $t \in T(F)$ peut "naturellement" être vu comme une structure sur la signature $\mathcal{S}_F$ en posant

$$D^{(t)} := \mathrm{dom}(t)$$

$$P_f^{(t)} := \{u \in D^{(w)} \mid t[u] = f\}, \;\; S_k^{(t)} := \{(u, v) \in D^{(w)} \times D^{(w)} \mid u \cdot k = v\}.$$

Chaque formule MSO, close, Φ sur la signature $\mathcal{S}_A$ définit alors le langage des termes qui sont des modèles de Φ :

$$\mathrm{L}(\Phi) := \{t \in T(F) \mid t \models \Phi\}.$$

Un langage $L \subseteq T(F)$ est dit MSO-définissable ssi, il existe une formule MSO Φ sur la signature $\mathcal{S}_F$ telle que $L = \mathrm{L}(\Phi)$.

Théorème 4.3.1 (Thatcher-Wright, 1968) *Soit F un alphabet gradué fini et $\mathcal{S}_F$ la signature MSO associée. Un langage de termes $L \subseteq T(F)$ est rationnel ssi il est MSO-définissable.*

On adapte la preuve du théorème sur les mots.
Partie 1 : Tout langage rationnel est définissable
Cette fois-ci, on construit une formule Φ qui exprime l'existence d'une "course" de l'automate fini sur le terme t.
Partie 2 : Tout langage définissable est rationnel.
Étape 1 : Toute formule MSO est equivalente à une formule de $MSO2$
Étape 2 :
- Toute formule atomique de $MSO2$ a un ensemble de modèles rationnel.
- Les connecteurs et les quantificateurs conservent la rationnalité : en effet les langages rationnels de termes sont clos par complémentation, union, homomorphismes alphabétiques directs et homomorphismes inverses.

Chapitre 5

Langages récursivement énumérables de mots

5.1 Grammaires contextuelles

Définition 5.1.1 *Une grammaire* contextuelle *est un quadruple* $< X, V, P, S >$ *où* X *et* V *sont deux alphabets disjoints,* S *est un élément de* V *et* P *est un ensemble fini de couples de la forme*

$$\alpha T \beta \to \alpha m \beta$$

où $\alpha, \beta \in (X \cup V)^*, T \in V$ *et* $m \in (X \cup V)^+$.

Les éléments de P sont appelés les *règles* de la grammaire. On remarquera que le mot m ne peut pas pas être *vide* dans la définition précédente. La définition 2.2.2 de la notion de dérivation directe reste valable pour les grammaires contextuelles. La relation de dérivation, notée $\overset{*}{\to}_G$, est la fermeture réflexive et transitive de la relation $\to_G$. Le langage *engendré* par G, noté L_G est défini par

$$\mathrm{L}_G := \{w \in X^* \mid S \overset{*}{\to}_G w\}$$

Un langage $L \subseteq X^*$ est dit contextuel ssi il existe une grammaire contextuelle G telle que $L = \mathrm{L}_G$. On note $CS(X^*)$ l'ensemble des langages contex-

tuels sur l'alphabet X (les initiales CS évoquent le mot anglais "context-sensitive" qui signifie "sensible au contexte").

Proposition 5.1.2 *L'ensemble $CS(X^*)$ est fermé par*
1- substitution contextuelle de $\mathcal{P}(X^*)$ dans $\mathcal{P}(X^*)$.
2- homomorphisme de X^* dans X^*.
3- inverse d'un homomorphisme de X^* dans X^*.
4- union.
5- produit.
6- étoile.
7- intersection avec les rationnels.

Preuve : La propriété 1 se prouve comme pour l'ensemble des langages algébriques. Les propriétés 2,4,5,6 s'en déduisent immédiatement. La propriété 7 se prouve de façon analogue à la proposition 2.6.4. On en déduit la propriété 3 en décomposant un homomorphisme en 3 opérations, comme on l'a vu dans la preuve de la proposition 2.6.5. □

Proposition 5.1.3 *L'ensemble $CS(X^*)$ est fermé par intersection.*

Esquisse de preuve : La preuve la plus naturelle de cette propriété consiste à définir une classe d'automates qui reconnaissent exactement les langages contextuels. Il s'agit des "machines de Turing linéairement bornées". L' idée intuitive est que, un langage L est CS ssi, il est possible de tester qu' un mot $w \in L$ au moyen d'un programme, qui n'utiliserait qu'un espace mémoire comportant au plus $k \cdot |w|$ bits (où k est un entier, qui dépend de L mais ne dépend pas de w). Un tel programme est dit fonctionner en *espace linéaire*. On montre alors aisément que si L_1, L_2 sont reconnus en espace linéaire (par des programmes P_1, P_2), alors il existe un programme P_3, qui reconnaît $\mathrm{L}_{M_1} \cap \mathrm{L}_{M_2}$ en espace linéaire. □

Théorème 5.1.4 (Szelepcsényi-Immerman 88) *L'ensemble $CS(X^*)$ est fermé par complémentation.*

Ce théorème a une preuve très délicate, qui repose, elle aussi, sur les MT linéairement bornées.

5.2 Grammaires à structure de phrase

Définition 5.2.1 *Une grammaire à structures de phrases est un quadruple* $< X, V, P, S >$ *où* X *et* V *sont deux alphabets disjoints,* S *est un élément de* V *et* P *est une partie finie de* $(X \cup V)^* V (X \cup V)^* \times (X \cup V)^*$.

Les éléments de P sont les règles de la grammaire. La seule restriction imposée par cette définition aux règles de grammaire est que, $(u, v) \in P \Rightarrow u$ contient au moins une lettre de V. Remarquons que v peut être le mot vide.

Exemple 5.2.2 *Considérons la grammaire* $G := \langle \{a, b, c\}, \{S, A, B, C, \bar{C}\}, P, S \rangle$ *dont l'ensemble de règles est constitué de :*

$$S \to AB\bar{C} \tag{5.1}$$

$$ABC \to AABBCC \quad AB\bar{C} \to AABBC\bar{C} \tag{5.2}$$

$$BC \to CB \tag{5.3}$$

$$CB \to BC \tag{5.4}$$

$$A \to a \qquad B \to b \tag{5.5}$$

$$\bar{C} \to c \qquad C\bar{C} \to \bar{C}c \tag{5.6}$$

Vérifions que $\mathrm{L}_G = \{a^n b^n c^n \mid n \geq 1\}$.
On vérifie le mot particulier :

$$S \to_G abc.$$

On a

$$ABC \to_G A \cdot ABC \cdot BC$$

dont on déduit, par récurrence sur n *que, pour tout* $n \geq 0$,

$$ABC \to_G A^n \cdot ABC \cdot (BC)^n \tag{5.7}$$

En utilisant la règle de départ (5.1), puis (5.2) et une commutation (5.3), on obtient

$$S \to_G AB\bar{C} \to_G AABBC\bar{C} \to_G A \cdot ABC \cdot B\bar{C} \tag{5.8}$$

En utilisant une suite de règles (5.3-5.4), on obtient

$$AA^n \cdot ABC \cdot (BC)^n B\bar{C} \stackrel{*}{\to}_G A^{n+2}B^{n+2}C^{n+1}\bar{C} \tag{5.9}$$

En enchaînant (5.8)(5.7)(5.9) on obtient

$$S \to_G A^{n+2}B^{n+2}C^{n+1}\bar{C}$$

et on appliquant ensuite des règles (5.6), puis (5.5)

$$S \to_G a^{n+2}b^{n+2}c^{n+2}.$$

Nous avons ainsi prouvé que

$$\mathrm{L}_G \supseteq \{a^n b^n c^n \mid n \geq 1\}.$$

Montrons l'inclusion réciproque. Le premier fait est que le rationnel

$$R := \{S\} \cup (\{A, a\}^* \cdot \{B, b, C\}^* \cdot \{\bar{C}, c\} \cdot c^*$$

est saturé par la relation $\to_G$ i.e., si $w \in R$ et $w \to_G w'$ alors $w' \in R$. Comme $S \in R$ on en conclut que $\mathrm{L}_G \subseteq R$.
Le second fait est que les trois nombres

$$|w|_{\{A,a\}} - |w|_{\{B,b\}}, \quad |w|_{\{A,a\}} - |w|_{\{C,c,\bar{C}\}}, \quad |w|_{\{B,b\}} - |w|_{\{C,c,\bar{C}\}}$$

sont invariants par la relation $\to_G$. Comme ces trois nombres sont nuls lorsque $w = S$, on en conclut que $\mathrm{L}_G \subseteq \{w \in \{A, a, B, b, C, c, \bar{C}, S\}^* \mid |w|_{\{A,a\}} = |w|_{\{B,b\}} = |w|_{\{C,c,\bar{C}\}}\}$.
Des deux inclusions ci-dessus on peut conclure que

$$\mathrm{L}_G \subseteq \{a^n b^n c^n \mid n \geq 1\},$$

ce qui achève notre vérification.

Une grammaire SP $< X, V, P, S >$ est dite *croissante* ssi, pour toute règle $(u, v) \in P$, $|u| \leq |v|$.

Proposition 5.2.3 *Un langage L est engendré par une grammaire contextuelle ssi il est engendré par une grammaire SP croissante.*

Esquisse de preuve : 1- Toute grammaire contextuelle est croissante.
2- Soit $G =< X, V, P, \sigma >$ une grammaire SP croissante. Nous décrivons une méthode de transformation de G en une grammaire CS G', équivalente à G. On appelle règle c.f. ("context-free" i.e. sans-contexte, en Anglais), une règle de la forme $S \rightarrow m$.
Etape 1 : Par la transformation déjà utilisée pour la forme normale de Chomsky, on transforme G en G_1, croissante, dont toute règle est, soit c.f., soit dans $V^* \times V^*$.
Etape 2 : On se ramène à des règles soit c.f., soit conservant la longueur, par la transformation suivante. Toute règle r :

$$S_1 \dots S_p \rightarrow T_1 \dots T_p T_{p+1} \dots T_q$$

avec $2 \leq p < q, S_i, T_j \in V$, est remplacée par l'ensemble de deux règles

$$S_1 \dots S_p \rightarrow T_1 \dots T_{p-1} T_{p+1} \dots T_q U_r, \;\; U_r \rightarrow T_{p+1} \dots T_q$$

où U_r est une nouvelle variable. On obtient ainsi G_2.
Etape 3 : On se ramène à des règles soit c.f., soit de longueur 2, par la transformation suivante. Toute règle r :

$$S_1 \dots S_p \rightarrow T_1 \dots T_p$$

avec $3 \leq p$, est remplacée par l'ensemble de $p-1$ règles

$$\begin{array}{rcl} S_1 S_2 & \rightarrow & T_1 U_1 \\ U_1 S_3 & \rightarrow & T_2 U_2 \\ \vdots & & \vdots \\ U_{p-2} S_p & \rightarrow & T_{p-1} T_p \end{array}$$

où $U_1, \dots, U_{p-2}$ sont de nouvelles variables (chaque traitement d'une règle r entraîne la création de nouvelles variables U_j).On obtient ainsi G_3.

Etape 4 : On se ramène à des règles soit c.f., soit contextuelles, par la transformation suivante. Toute règle r :

$$S_1 S_2 \rightarrow T_1 T_2$$

avec $S_1 \neq T_1$ et $S_2 \neq T_2$ est remplacée par l'ensemble de 4 règles

$$\begin{array}{rcl} S_1S_2 & \rightarrow & S_1\bar{S}_2 \\ S_1\bar{S}_2 & \rightarrow & \bar{S}_1\bar{S}_2 \\ \bar{S}_1\bar{S}_2 & \rightarrow & \bar{S}_1T_2 \\ \bar{S}_1T_2 & \rightarrow & T_1T_2 \end{array}$$

On obtient ainsi G_4 et l'on pose $G' := G_4$.
On démontre aisément, pour chaque étape de transformation, que $u \stackrel{*}{\rightarrow}_{G_i} v \Rightarrow u \stackrel{*}{\rightarrow}_{G_{i+1}} v$. La réciproque n'est vraie que pour des mots u, v écrits sur l'alphabet des variables de la grammaire G_i union l'alphabet X. Cette réciproque est plus délicate à prouver formellement. On trouvera à la section 5.4 une démarche possible pour construire une telle *preuve*. □

Exemple 5.2.4 *En appliquant la transformation de l'étape 4 à la grammaire croissante de l'exemple 5.2.2 on obtient la grammaire contextuelle équivalente* $G' := \langle\{a,b,c\}, \{S, A, B, B_1, B_2, C, C_1, C_2, \bar{C}, \bar{C}_3\}, P', S\rangle$

$$\begin{array}{rrrr} & & S\rightarrow AB\bar{C} & \\ & & ABC\rightarrow AABBCC & ABC\bar{C}\rightarrow AABBCC\bar{C} \\ BC\rightarrow BC_1, & BC_1\rightarrow B_1C_1, & B_1C_1\rightarrow B_1B, & B_1B\rightarrow CB \\ CB\rightarrow CB_2, & CB_2\rightarrow C_2B_2, & C_2B_2\rightarrow C_2C, & C_2C\rightarrow BC \\ & & A\rightarrow a, & B\rightarrow b \\ & & \bar{C}\rightarrow c & \\ C\bar{C}\rightarrow C\bar{C}_3, & C\bar{C}_3\rightarrow C_3\bar{C}_3, & C_3\bar{C}_3\rightarrow C_3c, & C_3c\rightarrow\bar{C}c \end{array}$$

Un langage est dit *récursivement énumérable* ssi il est engendré par une grammaire à structure de phrase (pas nécessairement croissante). On note $RE(X^*)$ l'ensemble des langages récursivement énumérables sur l'alphabet X.
Intuitivement : Un langage L est RE ssi, il existe un programme P, qui prenant en entrée n'importe quel entier n, s'arrête toujours après un temps de calcul fini, en imprimant un mot $w(n)$ sur l'alphabet X et qui énumère les mots de L i.e.

$$L = \{w(n) \mid n \geq 0\}.$$

On remarquera que, cette fois-ci, *aucune restriction* sur les ressources en temps et espace n'est imposée au programme P. Le programme P ne permet pas, a priori, de tester qu'un mot w appartient à L : si on constate que $w = w(n)$, alors on est sûr que $w \in L$; mais si l'on constate que $w \notin \{w(n) \mid 0 \leq n \leq M\}$, on ne sait pas s'il existe un entier $n' > M$ tel que $w = w(n')$ ou bien si $w \notin L$.

Proposition 5.2.5 *L'ensemble $RE(X^*)$ est fermé par*
1- substitution récursivement énumérable de $\mathcal{P}(X^*)$ dans $\mathcal{P}(X^*)$.
2- homomorphisme de X^* dans X^*.
3- inverse d'un homomorphisme de X^* dans X^*.
4- union.
5- produit.
6- étoile.
7- intersection.
L'ensemble $RE(X^*)$ n'est pas fermé par complémentation.

Esquisse de preuve : La propriété 1 se prouve comme pour l'ensemble des langages algébriques. Les propriétés 2,4,5,6 s'en déduisent immédiatement. La propriété 7 se prouve en définissant une classes d'automates, les machines de Turing, qui reconnaissent exactement les langages récursivement énumérables. On en déduit la propriété 3 en décomposant un homomorphisme en 3 opérations, comme on l'a vu dans la preuve de la proposition 2.6.5.
La non-clôture par complémentation se prouve "par diagonalisation". Soit $G_0, G_1, \ldots, G_n, \ldots$, une énumération effective des grammaires SP sur l'alphabet $\{a\}$. Posons

$$D := \{a^n \mid a^n \in L_{G_n}\}, \quad E := a^* - D$$

Montrons que E n'est pas RE. Si E était RE, alors il existerait $n \in \mathbb{N}$ tel que

$$E = L_{G_n}.$$

On aurait alors la suite d'équivalences :

$$a^n \in E \Leftrightarrow a^n \notin D \Leftrightarrow a^n \notin L_{G_n} \Leftrightarrow a^n \notin E$$

ce qui est contradictoire. On en conclut que E n'est pas RE.
En revanche, on peut énumérer, au moyen d'un programme, les éléments de

D : on classe, selon un ordre lexicographique, les couples (G, d) formés d'une grammaire SP G et d'une dérivation d dans G. Le programme énumère tous les couples

$$(G_0, d_0), \ldots, (G_k, d_k), \ldots,$$

-à la fin de la génération de (G_k, d_k), il imprime le mot a^m dans le cas où la dérivation d_k est terminale et se termine par a^m,
- sinon il n'imprime rien et passe au calcul de (G_{k+1}, d_{k+1}).
Ce programme énumère les éléments de D, donc D est RE. □

5.3 Hiérarchie de Chomsky

Nous avons étudié plusieurs ensembles de langages de mots :
3- l'ensemble des langages rationnels $RAT(X^*)$
2- l'ensemble des langages algébriques $ALG(X^*)$
1- l'ensemble des langages contextuels $CS(X^*)$
0- l'ensemble des langages récursivement énumérables $RE(X^*)$
On appelle "hiérarchie de Chomsky" cette classification des langages formels.

Théorème 5.3.1 *1- Pour tout alphabet X de cardinal supérieur ou égal à 2, la hiérarchie de Chomsky est stricte. i.e.*

$$RAT(X^*) \subset ALG(X^*) \subset CS(X^*) \subset RE(X^*).$$

2- Pour un alphabet X de cardinal 1,

$$RAT(X^*) = ALG(X^*) \subset CS(X^*) \subset RE(X^*).$$

Esquisse de preuve : Les inclusions au sens large sont claires.
Le langage $\{a^n b^n \mid n \geq 0\}$ appartient à $ALG(\{a, b\}^*) - RAT(\{a, b\}^*)$ (on le démontre à partir du "lemme de l'étoile" des rationnels, par exemple). Le langage $\{a^{n^2} \mid n \geq 0\}$ appartient à $CS(\{a\}^*) - ALG(\{a\}^*)$ (on le démontre à partir du "lemme d'itération" des algébriques).

On peut construire un ensemble d'entiers $E' \subseteq \mathbb{N}$ tel que : le problème de savoir si $n \in E'$ soit résoluble au moyen d'un algorithme, mais ce problème

ne puisse pas être résolu en utilisant un espace mémoire linéaire par rapport à n. Le langage $\{a^n \mid n \in E'\}$ appartient alors à $RE(\{a\}^*) - CS(\{a\}^*)$.
Un tel ensemble E' peut être construit par diagonalisation.
Soit $C_0, C_1, \ldots, C_n, \ldots$, une énumération effective des grammaires CS sur l'alphabet $\{a\}$. Posons

$$D' := \{a^n \mid a^n \in L_{C_n}\}, \;\; E' := a^* - D'.$$

Par le même argument que pour E (voir ci-dessus), E' n'est pas contextuel.
Il existe un programme qui teste si un mot $a^n \in E'$:
- on calcule la grammaire C_n
- on teste si a^n est engendré par C_n (ce qui peut être fait car C_n est croissante)
- on accepte a^n ssi a^n n'est pas engendré par C_n. Il existe donc, a fortiori, un programme qui énumère les éléments de E'. Donc $E' \in RE(a^*) - CS(a^*)$.
□

5.4 Un problème

(extrait de l'épreuve de Septembre 2004 de l'UE "Théorie des langages").
On se propose d'écrire la *preuve complète* du théorème : "toute grammaire croissante est équivalente à une grammaire contextuelle".
Partie 1
Soit $\mathcal{X}$ un alphabet fini et $\mathcal{R}$ un système semi-thuéien sur $\mathcal{X}^*$:

$$\mathcal{R} = \{ABC \longrightarrow DEF\} \cup \mathcal{R}'.$$

On suppose que A, B, C, D, E, F sont des lettres de $\mathcal{X}$, non nécessairement distinctes deux à deux. On construit un nouveau système semi-thuéien $\mathcal{S}$

$$\begin{array}{rcl} \mathcal{S} & = & \{A \longrightarrow L_1, B \longrightarrow L_2, C \longrightarrow L_3\} \cup \{R_1 \longrightarrow D, R_2 \longrightarrow E, R_3 \longrightarrow F\} \\ & \cup & \{L_1L_2L_3 \longrightarrow R_1R_2R_3\} \cup \mathcal{R}', \end{array}$$

où $\{L_i, R_i \mid 1 \leq i \leq 3\}$ est un ensemble de 6 lettres distinctes n'appartenant pas à $\mathcal{X}$.

1-Montrer que, pour tous $u, v \in \mathcal{X}^*$, si $u \xrightarrow[\mathcal{R}]{*} v$ alors $u \xrightarrow[\mathcal{S}]{*} v$.

On note $\mathcal{Y} = \mathcal{X} \cup \{L_i, R_i \mid 1 \leq i \leq 3\}$ et on définit un morphisme $\Phi : \mathcal{Y}^* \to \mathcal{X}^*$ par :

$$\Phi(L_1) = A, \Phi(L_2) = B, \Phi(L_3) = C, \Phi(R_1) = D, \Phi(R_2) = E, \Phi(R_3) = F$$

et $\forall x \in \mathcal{X}, \Phi(x) = x$.

2-Montrer que, pour tous $u, v \in \mathcal{Y}^*$, si $u \xrightarrow[\mathcal{S}]{*} v$ alors $\Phi(u) \xrightarrow[\mathcal{R}]{*} \Phi(v)$.

3-Montrer que, pour tous $u, v \in \mathcal{X}^*$, $u \xrightarrow[\mathcal{R}]{*} v$ si et seulement si $u \xrightarrow[\mathcal{S}]{*} v$.

On décompose le système $\mathcal{S}$ en

$$\mathcal{S} = \{L_1L_2L_3 \longrightarrow R_1R_2R_3\} \cup \mathcal{S}'$$

On construit un nouveau système semi-thuéien $\mathcal{T}$

$$\begin{array}{rcl} \mathcal{T} & = & \{L_1L_2 \longrightarrow \bar{L}_1\bar{L}_2, \bar{L}_2L_3 \longrightarrow \bar{L}_2\bar{L}_3, \bar{L}_2\bar{L}_3 \longrightarrow \hat{L}_2R_3, \bar{L}_1\hat{L}_2 \longrightarrow R_1R_2\} \\ & \cup & \mathcal{S}', \end{array}$$

où $\{\bar{L}_1, \bar{L}_2, \bar{L}_3, \hat{L}_2\}$ est un ensemble de 4 lettres distinctes n'appartenant pas à $\mathcal{Y}$.
On note $\mathcal{Z} = \mathcal{Y} \cup \{\bar{L}_1, \bar{L}_2, \bar{L}_3, \hat{L}_2\}$.

4- Montrer que, pour tous $u, v \in \mathcal{Y}^*$, si $u \xrightarrow[\mathcal{S}]{*} v$ alors $u \xrightarrow[\mathcal{T}]{*} v$.

5- Montrer que, pour tous $u \in \mathcal{Y}^*, v \in \mathcal{Z}^*$, si $u \xrightarrow[\mathcal{T}]{*} v$ alors toute occurrence de $\bar{L}_2$ dans v est immédiatement précédée d'une occurrence de $\bar{L}_1$.

6- On définit des applications $\Psi_1 : \{\bar{L}_1\bar{L}_2\bar{L}_3, \bar{L}_1\hat{L}_2\} \to \mathcal{Y}^*$ et $\Psi_2, \Psi : \mathcal{Z}^* \to \mathcal{Y}^*$ de la façon suivante :
on pose

$$\Psi_1(\bar{L}_1\bar{L}_2\bar{L}_3) = R_1R_2R_3, \Psi_1(\bar{L}_1\hat{L}_2) = R_1R_2$$

l'application $\Psi_2 : \mathcal{Z}^* \to \mathcal{Y}^*$ est le morphisme de monoïdes tel que :

$$\Psi_2(\bar{L}_i) = L_i, \Psi_2(\hat{L}_2) = L_2 \text{ et } , \forall y \in \mathcal{Y}, \Psi_2(y) = y,$$

puis, pour tout

$$w = \alpha_0 w_1 \alpha_1 \cdots \alpha_{j-1} w_j \alpha_j \cdots w_n \alpha_n \tag{5.10}$$

où chaque w_j est l' un des deux mots ($\bar{L}_1\bar{L}_2\bar{L}_3$ ou $\bar{L}_1\hat{L}_2$) et, pour tout $k \in [0, n]$, le mot α_k ne contient aucune occurrence ni de $\bar{L}_1\bar{L}_2\bar{L}_3$, ni de $\bar{L}_1\hat{L}_2$,

$$\Psi(w) = \Psi_2(\alpha_0)\Psi_1(w_1)\Psi_2(\alpha_1)\cdots\Psi_2(\alpha_{j-1})\Psi_1(w_j)\Psi_2(\alpha_j)\cdots\Psi_1(w_n)\Psi_2(\alpha_n).$$

Exemple : Le mot

$$w = \bar{L}_3\hat{L}_2A\bar{L}_1BA\bar{L}_1\hat{L}_2A\bar{L}_1\bar{L}_2\bar{L}_3AB\bar{L}_1\bar{L}_2\bar{L}_3BA\bar{L}_1\bar{L}_2BB$$

admet la décomposition

$$w = \bar{L}_3\hat{L}_2A\bar{L}_1BA\cdot\bar{L}_1\hat{L}_2\cdot A\cdot\bar{L}_1\bar{L}_2\bar{L}_3\cdot AB\cdot\bar{L}_1\bar{L}_2\bar{L}_3\cdot BA\bar{L}_1\bar{L}_2BB$$

et a pour image

$$\Psi(w) = L_3L_2AL_1BA\cdot R_1R_2\cdot A\cdot R_1R_2R_3\cdot AB\cdot R_1R_2R_3\cdot BAL_1L_2BB.$$

6.1 Tout mot $w \in \mathcal{Z}^*$ possède-t-il une décomposition de la forme (5.10) ? Cette décomposition est-elle unique ? L'application Ψ est-elle bien définie ? s'agit-t-il d'un homomorphisme ?

6.2 Supposons que $u \in \mathcal{Y}^*, v \in \mathcal{Z}^*$ et $u \xrightarrow[\mathcal{T}]{*} v$. Montrer que $\Psi(u) \xrightarrow[\mathcal{S}]{*} \Psi(v)$.

7- Montrer que, pour tous $u, v \in \mathcal{Y}^*$, $u \xrightarrow[\mathcal{S}]{*} v$ si et seulement si $u \xrightarrow[\mathcal{T}]{*} v$.

Partie 2

Soit $\mathcal{Y}_1$ un alphabet fini et $\mathcal{S}_1$ un système semi-thuéien sur $\mathcal{Y}_1^*$:

$$\mathcal{S}_1 = \{L_1L_2 \longrightarrow R_1R_2\} \cup \mathcal{S}'.$$

On suppose que L_1, L_2, R_1, R_2 sont 4 lettres distinctes de $\mathcal{Y}_1$ et $\mathcal{S}'$ ne contient pas la règle $L_1L_2 \longrightarrow R_1R_2$. On construit un nouveau système semi-thuéien

$$\begin{array}{rcl} \mathcal{T}_1 & = & \{L_1L_2 \longrightarrow L_1\bar{L}_2, L_1\bar{L}_2 \longrightarrow \bar{L}_1\bar{L}_2, \bar{L}_1\bar{L}_2 \longrightarrow \bar{L}_1R_2, \bar{L}_1R_2 \longrightarrow R_1R_2\} \\ & \cup & \mathcal{S}_1' \end{array}$$

où $\bar{L}_1, \bar{L}_2$ sont 2 lettres distinctes n'appartenant pas à $\mathcal{Y}_1$.

8- Montrer que, pour tous $u, v \in \mathcal{Y}_1^*$, si $u \xrightarrow[\mathcal{S}_1]{*} v$ alors $u \xrightarrow[\mathcal{T}_1]{*} v$.

9- On note $\mathcal{Z}_1 = \mathcal{Y}_1 \cup \{\bar{L}_1, \bar{L}_2\}$ et on définit des applications $\theta_2, \theta : \mathcal{Z}_1^* \to \mathcal{Y}_1^*$

de la façon suivante :
l'application $\theta_2 : \mathcal{Z}_1^* \to \mathcal{Y}_1^*$ est le morphisme de monoïdes tel que :

$$\theta_2(\bar{L}_1) = R_1, \theta_2(\bar{L}_2) = L_2 \text{ et } , \forall y \in \mathcal{Y}_1, \theta_2(y) = y,$$

puis, pour tout

$$w = \alpha_0 \bar{L}_1 \bar{L}_2 \alpha_1 \cdots \alpha_{j-1} \bar{L}_1 \bar{L}_2 \alpha_j \cdots \bar{L}_1 \bar{L}_2 \alpha_n \tag{5.11}$$

où aucun mot α_k ne contient d'occurrence du mot $\bar{L}_1 \bar{L}_2$,

$$\theta(w) = \theta_2(\alpha_0) L_1 L_2 \theta_2(\alpha_1) \cdots \theta_2(\alpha_{j-1}) L_1 L_2 \theta_2(\alpha_j) \cdots L_1 L_2 \theta_2(\alpha_n).$$

9.1 Tout mot $w \in \mathcal{Z}_1^*$ admet-il une décomposition de la forme (5.11) ? Certains mots $w \in \mathcal{Z}_1^*$ admettent-ils plusieurs décompositions de la forme (5.11) ?
Que peut-on en déduire sur la définition de θ ?

9.2 Supposons que $u, v \in \mathcal{Z}_1^*$ et $u \xrightarrow[\mathcal{T}_1]{*} v$. Montrer que $\theta(u) \xrightarrow[\mathcal{S}_1]{*} \theta(v)$.

10- Montrer que, pour tous $u, v \in \mathcal{Y}_1^*$, $u \xrightarrow[\mathcal{S}_1]{*} v$ si et seulement si $u \xrightarrow[\mathcal{T}_1]{*} v$.

Partie 3
On cherche à adapter les transformations de la partie 1 à des systèmes semi-thuéiens plus généraux.
Soit $\mathcal{X}$ un alphabet fini et $\mathcal{R}$ un système semi-thuéien sur $\mathcal{X}^*$:

$$\mathcal{R} = \{AB \longrightarrow DEF\} \cup \mathcal{R}'.$$

On suppose que A, B, D, E, F sont des lettres de $\mathcal{X}$, non nécessairement distinctes deux à deux.
11- On propose le nouveau système $\mathcal{R}_1$

$$\mathcal{R}_1 \quad = \quad \{B \longrightarrow BC, ABC \longrightarrow DEF\} \cup \mathcal{R}'$$

11.1 Est-il vrai que, pour tous $u, v \in \mathcal{X}^*$, si $u \xrightarrow[\mathcal{R}]{*} v$ alors $u \xrightarrow[\mathcal{R}_1]{*} v$?

11.2 Est-il vrai que, pour tous $u, v \in \mathcal{X}^*$, si $u \xrightarrow[\mathcal{R}_1]{*} v$ alors $u \xrightarrow[\mathcal{R}]{*} v$?

12- On propose le nouveau système $\mathcal{S}$:

$$\begin{aligned} \mathcal{S} \quad &= \quad \{A \longrightarrow L_1, B \longrightarrow L_2 L_3\} \cup \{R_1 \longrightarrow D, R_2 \longrightarrow E, R_3 \longrightarrow F\} \\ &\cup \quad \{L_1 L_2 L_3 \longrightarrow R_1 R_2 R_3\} \cup \mathcal{R}', \end{aligned}$$

où $\{L_i, R_i \mid 1 \leq i \leq 3\}$ est un ensemble de 6 lettres distictes n'appartenant pas à $\mathcal{X}$.

13-Montrer que, pour tous $u, v \in \mathcal{X}^*$, si $u \xrightarrow[\mathcal{R}]{*} v$ alors $u \xrightarrow[\mathcal{S}]{*} v$.

On note $\mathcal{Y} = \mathcal{X} \cup \{L_i, R_i \mid 1 \leq i \leq 3\}$

14- Est-il vrai que, pour tous $u, v \in \mathcal{X}^*$, $u \xrightarrow[\mathcal{R}]{*} v$ si et seulement si $u \xrightarrow[\mathcal{S}]{*} v$?

Partie 4

Une grammaire $G = (X, V, P, S)$ est dite *croissante* ssi ses règles sont de la forme

$$(\alpha, \beta) \in (X \cup V)^* V (X \cup V)^* \times (X \cup V)^* \text{ avec } \mid \alpha \mid \leq \mid \beta \mid .$$

15- Montrer que, pour toute grammaire croissante G, il existe une grammaire croissante $G' = (X, V', P', S')$, telle que $L(G) = L(G')$ et telle que, pour toute règle $(\alpha, \beta) \in P'$, $\alpha \in V'^+$.

On appelle *poids* de la grammaire $G = (X, V, P, S)$, l'entier $\nu(G) = max\{|\beta|, (\alpha, \beta) \in P\}$.

16- En utilisant les parties 1 et 3, montrer que toute grammaire croissante de poids 3 est équivalente à une grammaire croissante de poids 2.

17- En utilisant la partie 2, montrer que toute grammaire croissante de poids 2 est équivalente à une grammaire contextuelle de poids 2.

18- Pouvez-vous généraliser les arguments précédents pour prouver que : toute grammaire croissante est équivalente à une grammaire contextuelle de poids 2 ?

Bibliographie

[Aut94] J.M. Autebert. *Théorie des langages et des automates*. MASSON, 1994.

[BBC94] D. Beauquier, J. Berstel, and P. Chrétienne. *Eléments d'algorithmique*. MASSON, 1994.

[CDG+02] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 2002. Draft, available from `www.grappa.univ-lille3.fr/tata`.

[Sén02] G. Sénizergues. Outils mathématiques pour l'informatique, 2002. Notes de cours, disponibles à l'adresse `http://dept-info.labri.u-bordeaux.fr/~ges/ENSEIGNEMENT/polycop_omi.ps`.

[Tho97] W. Thomas. Languages, automata and logic. In *Handbook of language theory, Vol. 3, chap.7*, pages 389–455. Springer Verlag, 1997.