UNIVERSITÉ PARIS-EST

Thèse de doctorat

pour l'obtention du grade de

Docteur de l'Université Paris-Est

Spécialité Informatique

au titre de l'École Doctorale MSTIC

Présentée et soutenue publiquement par

DINH Trong Hieu

le juin 2011

Grammaires de graphes et langages formels

Devant le jury composé par :

Table des matières

1

Introduction

Cette thèse s'inscrit dans l'étude de modèles mathématiques permettant de décrire et manipuler des langages formels. Un premier modèle formel de calcul est celui des systèmes de récriture de mots, introduit par Axel Thue en 1914 [Thu 14]. Le problème du mot, formulé trois ans auparavant par Max Dehn [De 11] pour les groupes, est alors posé. Il consiste à trouver une méthode constructive pour déterminer l'existence d'une suite de récritures pour transformer un mot donné en un autre. Ce n'est que vingt ans plus tard que Alan Turing publie son article sur un modèle universel de machines [Tu 36]. Comme toute machine de Turing peut être simulée par un système de récriture de mots, il en résulte que le problème du mot est indécidable pour les systèmes de récriture. Ce résultat d'indécidabilité pour les systèmes de récriture de mots n'est pas un cas particulier : au contraire, d'après le théorème de Rice [Ri 53], l'indécidabilité reste vraie pour toute propriété non triviale. Pour obtenir des résultats de décidabilité, il y a donc lieu de particulariser les systèmes de récriture.

En 1956, Noam Chomsky [Ch 56] a présenté quatre familles décroissantes de grammaires. Les grammaires les plus générales sont les systèmes de récriture de mots sur deux alphabets de lettres terminales et non-terminales. Le langage engendré par une grammaire est l'ensemble des mots de lettres terminales obtenus par récriture (itérée) à partir d'une lettre non-terminale donnée. Les langages engendrés coincident avec les langages reconnus par les machines de Turing, appelés langages récursivement énumérables. Ces langages forment une famille non fermée par complémentaire. La deuxième famille de grammaires est celle des grammaires contextuelles qui engendrent les langages dits contextuels. Le problème du mot devient alors décidable, mais la vacuité d'un langage contextuel reste un problème indécidable. Cependant, la famille des langages contextuels est une algèbre de Boole effective, fermée notamment par complémentaire [Im 88], [Sz 88]. La troisième famille de grammaires

est constituée des grammaires algébriques, dont les membres gauches des règles sont réduites à une lettre non-terminale. Elles engendrent les langages algébriques. La vacuité d'un langage algébrique est un problème décidable, mais moult problèmes restent indécidables, comme par exemple celui de déterminer si le complémentaire d'un langage algébrique est algébrique. Enfin, la plus petite famille de grammaires dans cette hiérarchie de Chomsky est celle des grammaires linéaires droites (ou gauches) qui engendrent les langages réguliers. L'ensemble des langages réguliers est une algèbre de Boole effective. Elle est la plus petite famille de langages contenant les langages finis et qui soit fermée par union, concaténation et concaténation itérée, dite étoile.

C'est aussi en 1956 que Stephen Kleene a montré que les langages réguliers sont les langages reconnus par les automates finis [Kl 56]. Un automate fini peut être vu comme un graphe fini, simple, orienté, dont les arcs sont étiquetés par des lettres, et dont certains sommets sont désignés comme sommets de départ et sommets d'arrivée. Le langage reconnu par un automate fini est alors l'ensemble des mots étiquetant les chemins acceptants, c'est-à-dire allant d'un sommet de départ à un sommet d'arrivée.

Tous ces résultats en théorie des langages formels sont bien connus, et sont présentés dans de nombreux ouvrages (voir entre autres [Ha 78], [HU 79] et [Au 94]). Néanmoins, on n'y trouvera pas un résultat majeur et pourtant datant de plus de vingt cinq ans. Précisément, c'est en 1985 que David Muller et Paul Schupp [MS 85] ont apporté une vision géométrique des langages algébriques. Pour cela, ils ont étendu les automates finis en les automates, dits réguliers, dont les graphes sont de degré fini et finiment décomposables par distance : ils n'ont qu'un nombre fini (à isomorphisme près) de composantes connexes en supprimant pour tout entier d, les sommets dont la distance aux sommets de départ est inférieure à d. Les automates réguliers sont aussi les graphes des transitions des automates à pile, restreints aux ensembles réguliers de configurations. En particulier, les automates réguliers reconnaissent exactement les langages algébriques. De façon duale à la décomposition, on peut définir le graphe d'un automate régulier en l'engendrant à l'aide d'une grammaire algébrique de graphes : tout membre gauche est un arc étiqueté par une lettre non-terminale, et tout membre droit est un graphe fini dont les arcs sont étiquetés par des lettres terminales et non-terminales. De plus, on demande à ce qu'une grammaire de graphes soit fonctionnelle : deux membres gauches ne peuvent avoir la même étiquette.

Le fait de pouvoir décrire, à l'aide d'une grammaire de graphes, la structure géométrique d'un ensemble algébrique de mots, n'est pas anodin. En effet, bien des résultats classiques sur les langages algébriques deviennent naturels si on les aborde avec ce point de vue géométrique. Par exemple, la détermination des configurations accessibles à partir d'une configuration donnée se traduit par le calcul d'un plus petit point fixe sur la grammaire de graphes [Ca 08]. Au chapitre 3, on illustre

la pertinence de ce formalisme pour donner une nouvelle démonstration de deux résultats fondamentaux sur les langages algébriques : le lemme des paires itérantes et le lemme de Parikh. La démonstration du lemme de Parikh a été présentée dans [Di 06]. On pourrait poursuivre ce travail pour retrouver à l'aide des grammaires de graphes les autres résultats principaux sur les langages algébriques.

Les automates réguliers forment une extension simple et naturelle des automates finis. De nombreuses publications concernent les graphes finis et leurs applications. Le chapitre 4 est une première étape vers le développement d'une théorie algorithmique des graphes réguliers. On y décrit une extension aux graphes réguliers d'algorithmes de base pour déterminer un plus court chemin. Une façon générale d'aborder les problèmes de plus courts chemins ainsi que d'autres problèmes d'algorithmique sur les graphes est de prendre pour étiquettes des arcs les éléments d'un demi-anneau idempotent et continu. Pour tout graphe, la valeur d'un chemin est le produit de ses étiquettes successives. La valeur d'un graphe régulier engendré à partir d'un arc membre gauche est la somme des valeurs de ses chemins allant de la source au but de l'arc initial. La valeur d'une grammaire de graphes est alors le vecteur des valeurs des graphes engendrés à partir de chaque membre gauche.
On établit l'équivalence entre la sémantique algébrique et la sémantique opérationnelle : la valeur d'une grammaire est la borne supérieure de la suite obtenue en appliquant itérativement l'interprétation de la grammaire à partir de son plus petit élément. On présente ensuite deux méthodes pour calculer la valeur d'une grammaire pour un demi-anneau commutatif (idempotent et continu). La première méthode s'applique pour des demi-anneaux dont l'ordre naturel est total, et pour lesquels la valeur de la grammaire est déterminée en appliquant l'interprétation de la grammaire un nombre fini de fois. Le deuxième algorithme travaille avec n'importe quel demi-anneau commutatif et est une simple généralisation de la méthode de Hopkins-Kozen [HK 99] développée pour les grammaires algébriques (de mots). Ces deux algorithmes sont de complexité polynomiale et résolvent entre autres le problème du plus court chemin sur les graphes réguliers en utilisant les grammaires de graphes. Ces résultats ont été présentés au 16^{ieme} symposium international FCT [CD 07].

Le chapitre 5 porte sur les systèmes de récriture de mots. Tout comme une grammaire, un système de récriture de mots permet d'engendrer un langage, à savoir le langage des mots obtenus par récriture itérée à partir d'un mot donné, voire d'un langage donné. On dit qu'un système préserve la régularité si pour tout langage initial régulier, le langage engendré reste régulier. De même, un système préserve l'algébricité si pour tout langage initial algébrique, le langage engendré reste algébrique. D'après le théorème de Rice, on ne peut pas décider si un système quelconque préserve la régularité ou l'algébricité. Par contre, on connaît des sous-familles générales de systèmes préservant la régularité ou l'algébricité. Une famille bien connue est celle des systèmes monadiques dont les membres droits sont réduits à une

lettre ou le mot vide. Les systèmes monadiques préservent la régularité mais pas l'algébricité, et les systèmes monadiques inverses préservent l'algébricité mais pas la régularité [BO 93]. Une autre famille de systèmes est celle des systèmes effaçants pour lesquels toute lettre d'un membre droit est inférieure à une lettre du membre gauche pour un certain pré-ordre sur les lettres. Les systèmes effaçants préservent la régularité [HW 04], et les systèmes effaçants inverses préservent l'algébricité [Hi 74]. Enfin, tous les systèmes dont la relation de récriture itérée est une relation rationnelle, c'est-à-dire reconnue par un transducteur fini, préservent la régularité et l'algébricité. Ceci est le cas notamment pour les systèmes préfixes [Ca 90], les systèmes bifixes [KKO 06], et les systèmes bifixes marqués [Al 08]. On présente dans ce dernier chapitre une méthode de décomposition de la récriture itérée permettant d'obtenir des familles générales de systèmes préservant la régularité ou l'algébricité. En particulier, on a étendu les sytèmes bifixes marqués en autorisant l'ajout de marques dans les membres droits des règles. Ces systèmes n'ont plus une récriture itérée rationnelle mais ils préservent l'algébricité, et leurs inverses préservent la régularité ; ce qui répond positivement à la conjecture émise en [Al 09]. Ce travail vient d'être présenté à la conférence FoSSaCS [CD 11].

Avant de présenter ces divers travaux, le chapitre 2 rassemble les principales définitions et notations qui nous seront utiles dans ce document.

2

Notions préliminaires

Dans ce chapitre, nous introduisons les notations de base et les concepts fondamentaux que nous utilisons dans ce mémoire. Après avoir rappelé les notions de mot et de langage formel, nous présentons les systèmes de récriture de mots, ainsi qu'un bref survol de la hiérarchie bien connue des familles de langages, attribuée à Chomsky. Nous donnons ensuite les notions de graphe et de grammaire de graphes. Enfin, nous montrons comment un graphe reconnaît un langage, et comment on engendre un graphe à l'aide d'une grammaire de graphes.

2.1 Grammaires de mots et langages

2.1.1 Mots et langages

Nous considérons des ensembles finis de symboles, ou *lettres*, appelés *alphabets*.

Les suites finies de lettres sont appelées *mots*. Autrement dit, un mot u de longeur $n \geq 0$ sur un alphabet Σ peut être vu comme un n-uplet $(a_1, \cdots, a_n)$ d'éléments de Σ, ou de façon équivalente comme une application de l'intervalle des entiers [1,n] dans Σ. Tout mot $u = (a_1, \cdots, a_n)$ est habituellement noté $a_1 \cdots a_n$; sa i-ème lettre est $u(i) = a_i$. L'ensemble de tous les mots construits sur Σ se note Σ^*.

Un *langage* sur l'alphabet Σ est un sous-ensemble de Σ^*.

Le nombre d'occurrences d'une lettre a dans un mot u est noté $|u|_a$. Ainsi, le nombre total d'occurrences de toutes les lettres dans le mot u est sa *longueur* et se note $|u|$. L'unique mot de longueur 0, qui ne contient aucune lettre, est appelé le *mot vide* et est noté ϵ.

La *concaténation* de deux mots $u = a_1 \cdots a_n$ et $v = b_1 \cdots b_m$, qui se note $u.v$ ou plus simplement uv, est le mot $uv = a_1 \cdots a_n b_1 \cdots b_m$.

Ainsi, la concaténation est une opération associative ayant ϵ pour élément neutre.

Pour tout mot $u = xvy$, v est un *facteur* de u, x est un *préfixe* de u et y est un *suffixe* de u. En particulier, tout préfixe ou suffixe d'un mot est un *facteur* du mot.

Pour tout $n \in \mathbb{N}$, *le mot* u^n est la concaténation n fois du mot u et est défini par récurrence de la façon suivante :

$$\begin{array}{lcl} u^0 & = & \epsilon \\ u^n & = & u^{n-1}.u \quad \textit{pour tout } n > 0. \end{array}$$

L'opération de *concaténation* s'étend aux langages : pour tout $A, B \subseteq \Sigma^*$,

$$AB = \{\, uv \mid u \in A \wedge v \in B \,\}.$$

L'itération de la concaténation à tout langage L est : $L^* = \bigcup_{n \geq 0} L^n$, avec $L^0 = \{\epsilon\}$ et $L^{n+1} = L^n.L$ pour tout $n \geq 0$.

De façon abusive, on pourra écrire u à la place de $\{\mathrm{u}\}$. Ainsi Σ^n est l'ensemble des mots de longueur n.

Exemple. Avec un alphabet $\Sigma = \{0, 1\}$, nous avons :

$$\begin{array}{rcl} \Sigma^* & = & \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \cdots\} \\ (11000)^0 & = & \epsilon \\ (110)^2 & = & 110110 \\ 1^3 & = & 111 \end{array}$$

Monoïdes. La structure algébrique sous-jacente aux mots est le *monoïde*. Un monoïde M est simplement un ensemble, potentiellement infini, muni d'une opération interne associative appelée produit, et d'un élément neutre 1_M. Une partie S d'un monoïde M est un ensemble de *générateurs* de M si chaque élément de M admet une décomposition comme produit d'éléments de S : $M = S^*$; on dit que S est *libre* si toute décomposition en éléments de S est unique :

Si $\quad u_1 \cdots u_m = v_1 \cdots v_n$ avec $u_1, \cdots, u_m, v_1, \cdots, v_n \in S$
alors $\quad m = n$ et $u_1 = v_1, \cdots, u_n = v_n$.

Si de plus S est fini, M est dit de *type fini*.

Exemple. L'ensemble $\mathbb{N}$ des entiers positifs est un monoïde pour l'addition. Son élément neutre est 0. Il est librement engendré par le singleton $\{1\}$ et est donc *de type fini*.

Exemple. L'ensemble Σ^* de tous les mots sur un alphabet Σ est un monoïde pour la concaténation. Il est de type fini et librement engendré par Σ. Son élément neutre est le mot vide ϵ.

2.1.2 Systèmes de récriture

Les systèmes de récriture figurent parmi les formalismes les plus simples et les plus généraux pour la modélisation de transformations sur les mots (voir par exemple [DJ 90] et [Ca 90]). Ils généralisent les grammaires, peuvent représenter des calculs d'automates finis, de transducteurs, d'automates à pile ou même de machines de Turing. Autant que faire se peut, nous donnerons une caractérisation de chacun de ces formalismes en termes de systèmes de récriture.

Un système de récriture de mots sur l'alphabet Σ est un ensemble potentiellement infini de règles de récriture, c'est-à-dire de couples de mots $(l, r) \in \Sigma^* \times \Sigma^*$.

Exemple. Un système de récriture sur l'alphabet $\Sigma = \{0, 1, S\}$ est le suivant :

$$R \;=\; \{(S, 0S), (S, 1S), (S, 0), (S, 1)\}$$

Ce qui s'écrit de la façon plus lisible suivante :

$$\begin{array}{lcl} S & \rightarrow & 0S \\ S & \rightarrow & 1S \\ S & \rightarrow & 0 \\ S & \rightarrow & 1 \end{array}$$

On note $Dom(R)$ (resp. $Im(R)$) l'ensemble des parties gauches (récipoquement droites) des règles de R. Une règle (l, r) récrit un mot w en w' en remplaçant une instance de l dans w par r. Plus formellement, la relation de *récriture* d'un système R est :

$$\underset{R}{\longrightarrow} \;:=\; \{\; (ulv, urv) \mid (l, r) \in R \wedge u, v \in \Sigma^* \;\}.$$

La clôture réflexive et transitive de cette relation pour la composition est la *dérivation* de R et est notée habituellement $\overset{*}{\underset{R}{\longrightarrow}}$. Pour tout u, v tels que $u \overset{*}{\underset{R}{\longrightarrow}} v$, il existe une suite de mots $u_0, \cdots, u_n$ telle que :

$$u \;=\; u_0 \;\underset{R}{\longrightarrow}\; u_1 \cdots u_{n-1} \;\underset{R}{\longrightarrow}\; u_n \;=\; v.$$

Nous disons que v est *dérivable* de u ou bien que u se dérive en v.

Un mot w ne contenant aucune instance de partie gauche de règle d'un système R est appelé une *forme normale* (ou *forme irréductible*) de R. Un système de récriture pour lequel tout mot w peut être dérivé en une forme normale est appelé *normalisant*, et *fortement normalisant* si cette forme normale est unique pour tout mot donné.

2.1.3 Grammaires et hiérarchie de langages

Dans cette partie, nous donnons quelques rappels sur la hiérarchie la plus connue de la théorie des langages formels, à savoir la hiérarchie de familles de langages dite de Chomsky [Ch 59]. Définie il y a plus de cinquante ans, elle est formée des quatre familles décroissantes des langages suivants : les langages récursivement énumérables, les langages contextuels, les langages algébriques et les langages réguliers. Cette partie rappelle les quatre classes de grammaires utilisées pour définir les familles de la hiérarchie de Chomsky.

La hiérarchie de Chomsky fut initialement définie en termes de grammaires qui sont des cas particuliers de systèmes de récriture de mots, et ce dans le but de fournir des outils pour l'étude de langues naturelles.

Une *grammaire* G est caractérisée par un quadruplet $G = (T, N, S, P)$ où T et N sont deux alphabets finis disjoints et

(i) T est l'alphabet de symboles terminaux de la grammaire ; T est appelé l'*alphabet terminal*,

(ii) N est l'alphabet de symboles non-terminaux ou symboles variables de la grammaire ; N est appelé l'*alphabet non-terminal*,

(iii) $S \in N$ est le symbole initial appelé *axiome* ou *racine* de la grammaire,

(iv) P est un système de récriture de mots sur l'alphabet $T \cup N$, dont les éléments sont en général appelés les règles de la grammaire. Pour plus de simplicité, on requiert en général que les parties gauches des règles de P contiennent au moins un symbole non-terminal.

On dit qu'un mot $u \in T^*$ est *engendré* par G s'il peut être dérivé depuis l'axiome S selon le système P. Le *langage* de la grammaire G est défini comme l'ensemble de tous les mots terminaux (dans T^*) qui sont engendrés par G :

$$L(G) \;=\; \{\, u \in T^* \mid S \xrightarrow[P]{*} u \,\}.$$

Remarque. Par définition des règles de la grammaire, un mot ne contenant que des symboles terminaux ne peut plus se récrire ; il s'agit d'une forme normale pour le système de récriture P. Le fait de distinguer explicitement les symboles terminaux des non-terminaux n'est pas une caractéristique essentielle des grammaires, qui peuvent en fait être vues simplement comme des systèmes de récriture.

Les quatre classes de langages de la hiérarchie de Chomsky sont définies en posant des contraintes supplémentaires sur la forme des règles de production des grammaires. Rappelons qu'une première contrainte est que toute partie gauche contienne au moins un non-terminal. Les familles de grammaires considérées sont les suivantes (voir [Ha 78] ou [HU 79]) :

(i) Une grammaire est *non restreinte* ou *de type 0* si aucune contrainte supplémentaire n'est mise sur la forme de ses règles. Ces grammaires engendrent les langages *récursivement énumérables*, aussi appelés *langages de type 0*.

(ii) Une grammaire est *sensible au contexte* ou *contextuelle* ou *de type 1* si toutes ses productions sont de la forme :
- $yAz \to ywz$ où $A \in N, w \in (T \cup N)^+$, et $y, z \in (T \cup N)^*$, c'est-à-dire que A peut être remplacé par w non vide, avec le contexte gauche y et le contexte droit z.
- $S \to \epsilon$, à condition que S ne se retrouve pas dans la partie droite d'une règle.

Son langage engendré est appelé *contextuel*, ou *de type 1*.

(iii) Une grammaire est *hors-contexte* ou *algébrique* ou *de type 2* si toutes ses productions sont de la forme $(A,\ w)$, avec $A \in N$ et $w \in (N \cup T)^*$. Le langage d'une telle grammaire est *hors-contexte* ou *algébrique* ou *de type 2*. La famille des langages algébriques sur l'alphabet T est notée $Alg(T^*)$. Ce sont aussi les langages reconnus par les *automates à pile*.

(iv) Une grammaire est *linéaire à gauche* (ou à droite) si toutes ses règles sont de la forme (A,u) ou (A,Bu) (resp. (A,uB)), où A, B sont des non-terminaux et u est un mot terminal. Une grammaire est dite *régulière* ou *de type 3* si elle est linéaire à gauche ou à droite. Le langage engendré est appelé *régulier* ou *rationnel* ou *de type 3*. La famille des langages rationnels sur l'alphabet T est notée $Rat(T^*)$. Ce sont aussi les langages reconnus par les automates finis.

Ces quatre familles de grammaires engendrent une hiérarchie stricte de familles de langages, dite de Chomsky. Les langages de type 0 sont les plus généraux :

$$\mathcal{L}_3 \ \subset \ \mathcal{L}_2 \ \subset \ \mathcal{L}_1 \ \subset \ \mathcal{L}_0.$$

Remarque. L'inclusion est *stricte*, c'est-à-dire qu'il y a au moins un langage dans $\mathcal{L}_i$ qui n'est pas dans $\mathcal{L}_{i+1}$ pour chaque $0 \leq i < 3$.

2.2 Grammaires de graphes et langages

2.2.1 Graphes

Un *graphe* G simple, orienté et étiqueté sur un alphabet Σ, ou Σ-graphe, est une partie de $V \times \Sigma \times V$, où V est un ensemble dénombrable quelconque. Un élément (s, a, t) de G est un *arc* de *source* s, de *but* t et d'*étiquette* a, et se note $s \xrightarrow[G]{a} t$ ou simplement $s \xrightarrow{a} t$ si G peut clairement se déduire du contexte.

Un arc dont la source et le but sont identiques est une *boucle*. L'ensemble

$$V_G \;=\; \{\, s \mid \exists\, a,\ \exists\, t,\ (s \xrightarrow{a} t \;\vee\; t \xrightarrow{a} s)\,\}$$

des sources et buts des arcs de G est l'ensemble des *sommets* du graphe G.

Cette vision des graphes comme des ensembles d'arcs nous permet de leur appliquer les opérations ensemblistes habituelles, notamment l'union, l'intersection et la complémentation.

Un graphe inclus dans un autre graphe G est appelé un *sous-graphe* de G. Le sous-graphe $G_{|P}$ *induit* par un sous-ensemble $P \subseteq V_G$ de sommets de G est l'ensemble des arcs de G dont la source et le but sont tous les deux dans P :

$$G_{|P} \;=\; G \cap (P \times \Sigma \times P);$$

on parle aussi de la *restriction* de G à P.

2.2.2 Chemins

Un suite d'arcs $(s_1 \xrightarrow{a_1} t_1, \cdots, s_k \xrightarrow{a_k} t_k)$ telle que $t_1 = s_2, \cdots, t_{k-1} = s_k$ est appelée un *chemin*, et on note $s_1 \xrightarrow{a_1 \cdots a_k} t_k$; les sommets s_1 et t_k sont respectivement la *source* et le *but* du chemin.

Un chemin est un *circuit* si sa source et son but sont égaux.

Exemple. Dans la figure ci-dessous, on a le chemin : $1 \xrightarrow{a} 2 \xrightarrow{d} 4 \xrightarrow{f} 5 \xrightarrow{g} 6$, et le circuit : $4 \xrightarrow{f} 5 \xrightarrow{b} 3 \xrightarrow{c} 2 \xrightarrow{d} 4$.

Un graphe G est *déterministe* (respectivement *co-déterministe*) s'il ne contient aucune paire d'arcs de même source (respectivement but) et de même étiquette :
si $r \xrightarrow{a} s$ et $r \xrightarrow{a} t$ alors $s \;=\; t$ (respectivement si $s \xrightarrow{a} r$ et $t \xrightarrow{a} r$ alors $s \;=\; t$).

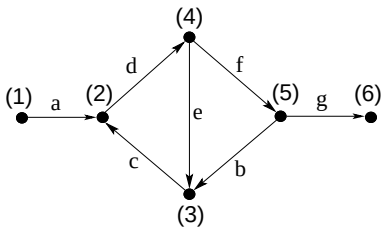


FIGURE 2.1 – Un graphe.

Étant donné un sommet s, le nombre d'arcs de source (resp. de but) s est son *degré sortant* $d^+(s)$ (resp. *degré entrant* $d^-(s)$) : $d^+(s) = |\{ (a,t) \mid s \xrightarrow{a} t\}|$ et $d^-(s) = |\{ (t,a) \mid t \xrightarrow{a} s \}|$.

La somme de ces deux valeurs est le *degré* $d(s)$ de s : $d(s) = d^+(s) + d^-(s)$.

Ces trois nombres $d^+(s)$, $d^-(s)$, $d(s)$ sont étendus des sommets aux graphes en considérant le maximum sur tous les sommets : $d^+(G) = max\{ d^+(s) \mid s \in V_G \}$; $d^-(G) = max\{ d^-(s) \mid s \in V_G \}$ et $d(G) = d^+(G) + d^-(G)$.

D'une façon équivalente, un graphe G est un *système de transition étiquetée*, c'est-à-dire un ensemble fini de relations binaires d'arcs $\xrightarrow{a}$ sur V, une pour chaque étiquette $a \in \Sigma$. Le graphe G^{-1}, défini par $(\xrightarrow{a})^{-1}_{a\in\Sigma}$, est le *graphe inverse* de G. Nous notons $\longrightarrow$ la relation d'adjacence (définie comme $\bigcup_{a\in\Sigma} \xrightarrow{a}$) et $\longleftrightarrow$ sa fermeture par inverse :

$$\longrightarrow \;=\; \{ (s,t) \mid \exists\, a \; (s \xrightarrow[G]{a} t) \} \quad \text{et} \quad \longleftrightarrow \;=\; \longrightarrow \cup \longrightarrow^{-1}.$$

Un sous-graphe de G dont le support est fermé par la relation $\longleftrightarrow$ est appelé une *composante connexe* de G. Intuitivement, les composantes connexes d'un graphe sont les parties maximales du graphe dans lesquelles tout sommet est accessible depuis tout autre par un chemin non dirigé (c'est-à-dire en utilisant les arcs dans les deux sens). Un sous-graphe H de G est *fortement connexe* si pour tout $s,t \in V_H$, on a $s \xrightarrow[H]{*} t$. Une composante fortement connexe de G est une partie fortement connexe et maximale (pour l'inclusion). Un sommet d'un graphe est appelé *racine* s'il existe un chemin depuis ce sommet vers tout autre sommet du graphe. Un graphe qui possède une racine est dit *enraciné*.

2.2.3 Graphes et langages

On peut associer un langage à un graphe en considérant l'ensemble des mots qui étiquettent ses chemins entre deux ensembles déterminés de sommets. Formellement,

on appelle *ensemble de traces* (ou *traces*) d'un graphe G à étiquettes dans Σ entre les ensembles de sommets I et F, ou simplement *langage* de G, le langage

$$L(G, I, F) \;=\; \{\, u \in \Sigma^* \mid \exists\, s \in I,\ \exists\, t \in F,\ (s \xrightarrow[G]{u} t) \,\}.$$

La spécification des ensembles des sommets initiaux et finaux autorisés se révèle importante pour les traces d'un graphe ou d'une famille de graphes. Par exemple, cela induit des différences importantes de ne considérer que des ensembles finis plutôt que des ensembles réguliers ou algébriques. Nous dirons que deux familles de graphes F_1 et F_2 sont *trace-équivalentes* par rapport à deux familles K_I et K_F de sous-ensembles de V si, pour tous $G \in F_i, I \in K_I, F \in K_F$, il existe un graphe $H \in F_{3-i}$ et deux ensembles $I' \in K_I$ et $F' \in K_F$ tels que $L(G, I, F) = L(H, I', F')$. L'étude des traces de familles de graphes infinis est une motivation principale de bien des travaux dans le domaine des graphes infinis.

2.2.4 Isomorphisme de graphes

Il existe plusieurs façons de composer des graphes. Deux graphes sont évidemment identiques s'ils consistent en un même ensemble d'arcs (et donc de sommets). Mais l'on s'intéresse en général à des notions d'équivalence moins restrictives. Nous avons déjà cité la notion de trace-équivalence, nous allons à présent rappeler celle d'isomorphisme de graphes.

L'image d'un graphe G de support V_G par une application ϕ de V_G dans un ensemble V est le graphe

$$\phi(G) \;=\; \{\, (\phi(s), a, \phi(t)) \mid (s, a, t) \in G \,\}.$$

On dit que ϕ est un *morphisme* du graphe G dans un graphe H si pour tout arc $(s, a, t) \in G,\ (\phi(s), a, \phi(t)) \in H$ c'est-à-dire $\phi(G) \subseteq H$. Si de plus ϕ est bijectif et ϕ^{-1} est un morphisme de H à G, alors ϕ est un *isomorphisme* et on dit que G et H sont *isomorphes*. La classe d'équivalence $[G]$ d'un graphe G par isomorphisme est l'ensemble de tous les graphes isomorphes à G. On dit alors que G est un *représentant* de sa classe d'équivalence. On confond très souvent G et sa classe d'équivalence, et on les note ainsi tous les deux G.

Intuitivement, deux graphes sont isomorphes si ils sont identiques à un renommage des sommets près. En pratique, on compare deux familles de graphes en oubliant le nom des sommets pour ne considérer que la structure des graphes.

2.2.5 Grammaires de graphes

Nous nous limitons à des grammaires pour lesquelles chaque membre gauche est un arc étiqueté du sommet 1 au sommet 2, et il n'y a pas dans les membres droits

d'arc sortant de 2, ni d'arc entrant en 1.
Une *grammaire* R est un ensemble fini de règles de la forme :

$$(1, A, 2) \to H$$

où $(1, A, 2)$ est un arc étiqueté par A de source 1, de but 2, et H est un graphe fini.

Les étiquettes des membres gauches forment l'ensemble N_R des *non-terminaux* de R :

$$N_R \ := \ \{\, A \mid (1, A, 2) \in Dom(R) \,\}$$

et les étiquettes restantes dans R forment l'ensemble T_R des *terminaux* :

$$T_R \ := \ \{\, A \notin N_R \mid \exists H \in Im(R),\ \exists s, t,\ (s, A, t) \in H \,\}.$$

En outre, nous avons besoin que chaque membre droit n'ait pas d'arc de but 1 ou de source 2 :

$$\forall H \in Im(R),\ \forall (s, a, t) \in H,\ s \neq 2 \ \wedge \ t \neq 1.$$

Nous disons que R est une *grammaire sans circuit* si ses membres droits sont des graphes sans circuits.

A partir de n'importe quel graphe axiome, nous voulons une grammaire de graphes pour engendrer un graphe unique à isomorphisme près. Nous devons donc nous limiter aux grammaires *déterministes* pour lesquelles deux règles ont des membres gauches distincts :

$$((1, A, 2), H)\,,\ ((1, A, 2), K) \in R \ \implies \ H = K.$$

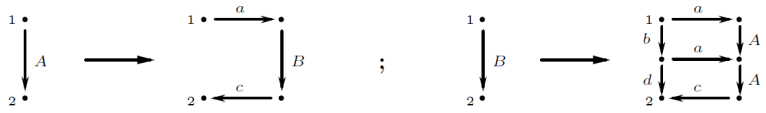


FIGURE 2.2 – Une grammaire déterministe de graphes

A partir d'un graphe, cette grammaire engendre un graphe unique (à isomorphisme près) obtenu en appliquant indéfiniment des récritures parallèles. Pour toute grammaire R, la *récriture* $\underset{R}{\longrightarrow}$ est la relation binaire entre graphes définie par $M \underset{R}{\longrightarrow} N$ si nous pouvons choisir un arc non-terminal $X = (s, A, t)$ dans M et une partie droite H de A dans R pour remplacer X par H dans M :

$$N \ = \ (M - \{X\}) \cup h(H)$$

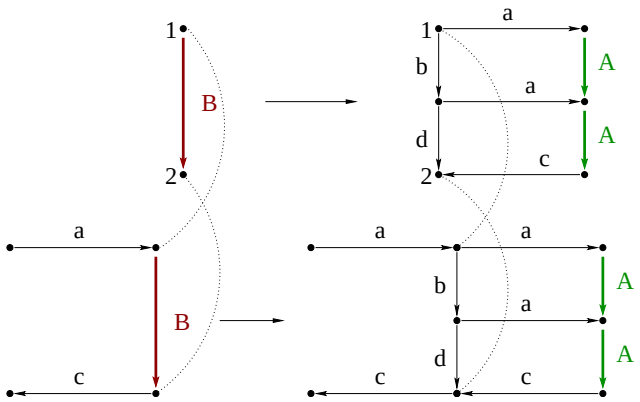


FIGURE 2.3 – Récriture pour la grammaire de la figure 2.2

pour une injection h qui associe le sommet 1 à s, le sommet 2 à t, et les autres sommets de H à des sommets autres que ceux de M. Cette récriture est notée par $M \underset{R,X}{\longrightarrow} N$.

La récriture $\underset{R,X}{\longrightarrow}$ d'un arc non-terminal X est étendue de manière évidente à la récriture $\underset{R,E}{\longrightarrow}$ de tout sous-ensemble E d'arcs non-terminaux. La *récriture parallèle* $\underset{R}{\Longrightarrow}$ est la récriture de l'ensemble des arcs non-terminaux : $M \underset{R}{\Longrightarrow} N$ si $M \underset{R,E}{\longrightarrow} N$ où E est l'ensemble des arcs non-terminaux de M.

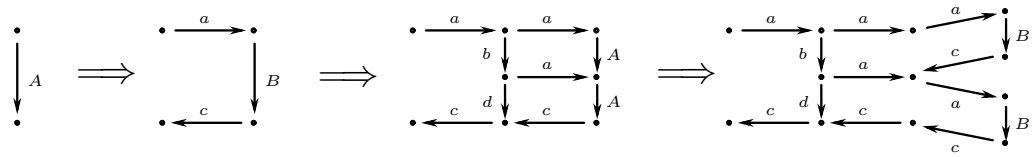


FIGURE 2.4 – Récritures parallèles pour la grammaire de la figure 2.2

Grâce aux deux arcs non-terminaux dans la partie droite de B pour la grammaire de la figure 2.2, nous obtenons après n récritures parallèles appliquées à partir de $H_0 = \{(1, A, 2)\}$, un graphe H_n ayant un nombre exponentiel d'arcs.

La *dérivation* $\underset{R}{\Longrightarrow}^*$ est la fermeture réflexive et transitive pour la composition de la récriture parallèle $\underset{R}{\Longrightarrow}$ *i.e.* $G \underset{R}{\Longrightarrow}^* H$ si H est obtenu de G par une suite consécutive de récritures parallèles.
Nous notons par $[M]$ l'ensemble des arcs terminaux d'un graphe M :

$$[M] := M \cap V_M \times T_R \times V_M.$$

Maintenant, nous supposons que toute grammaire R est déterministe : tout non-terminal $A \in N_R$ admet un unique membre droit.

Un graphe G est *engendré par une grammaire* R à partir d'un graphe H si G est isomorphe à un graphe dans l'ensemble $R^\omega(H)$ des graphes isomorphes suivant :

$$R^\omega(H) \;:=\; \{\, \bigcup_{n\geq 0} [H_n] \mid (H_0 = H) \wedge (\forall n \geq 0, H_n \underset{R}{\Longrightarrow} H_{n+1}) \,\}.$$

Par exemple, en itérant indéfiniment la dérivation de la figure 2.4, nous obtenons le graphe infini ci-dessous.

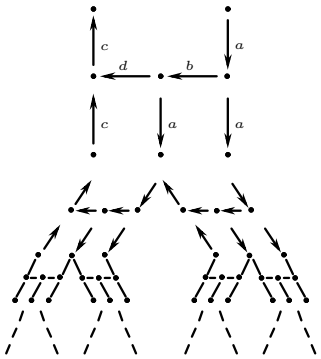


FIGURE 2.5 – Graphe engendré par la grammaire dans la figure 2.2

Le langage $L_R(A)$ des mots reconnus par R à partir du non-terminal A est le *langage reconnu* par $R^\omega(1 \xrightarrow{A} 2)$ du sommet 1 au sommet 2 :

$$L_R(A) \;=\; L(G, 1, 2) \;\text{ pour }\; G \in R^\omega(1 \xrightarrow{A} 2).$$

A toute grammaire algébrique G, on peut associer de façon effective une grammaire déterministe de graphes R qui soit sans circuit et dont l'ensemble des traces est le langage engendré par G.

Proposition 1 *On peut transformer toute grammaire algébrique* $G = (T, N, P)$ *en une grammaire de graphes* $R = \{\, (1, A, 2) \longrightarrow H_A \mid A \in N \,\}$ *sans circuit tel que* $L_G(A) = L_R(A)$ *pour tout non-terminal* $A \in N$.

Démonstration. Pour tout non-terminal $A \in N$ de G, on définit :

$$\begin{aligned}
H_A \;:=\;& \{\, 1 \xrightarrow{a} a.v \mid (A, av) \in P \wedge a \in N \cup T \wedge v \neq \epsilon \,\} \\
\cup\;& \{\, u.a \xrightarrow{a} 2 \mid (A, ua) \in P \wedge a \in N \cup T \wedge u \neq \epsilon \,\} \\
\cup\;& \{\, 1 \xrightarrow{a} 2 \mid (A, a) \in P \wedge a \in N \cup T \,\} \\
\cup\;& \{\, u.av \xrightarrow{a} ua.v \mid (A, uav) \in P \wedge a \in N \cup T \wedge u, v \neq \epsilon \,\}
\end{aligned}$$

Remarquons que l'ensemble V_R des sommets de R est

$$V_R := \{ u.v \mid uv \in Im(P) \wedge u, v \neq \epsilon \} \cup \{1, 2\}.$$

□

Par exemple, on considère la grammaire algébrique $G = (T, N, P)$ où $T = \{a, b, c, d\}$, $N = \{A, B\}$ et où les règles sont :

$$\begin{array}{lcl} A & \rightarrow & Bb \\ A & \rightarrow & aAB \\ B & \rightarrow & c \\ B & \rightarrow & BBd \end{array}$$

La construction de la proposition 1 associe à la grammaire algébrique G la grammaire de graphes R constituée des deux régles comme la figure 2.6.

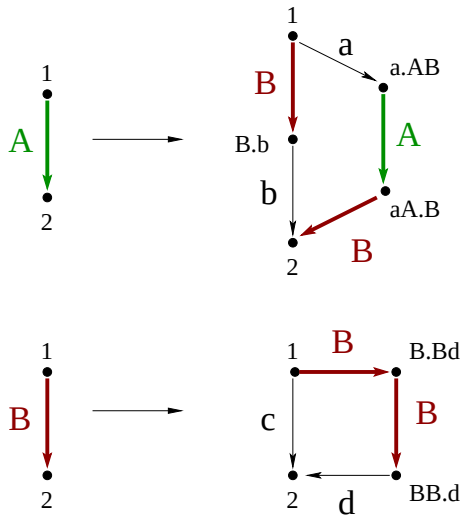


FIGURE 2.6 – Une grammaire de graphe.

La réciproque de la proposition 1 est vraie avec aussi une transformation de complexité linéaire.

Proposition 2 *On peut transformer toute grammaire R de graphes en une grammaire algébrique G telle que $L_R(A) = L_G(A)$ pour tout $A \in N_R$.*

Démonstration. Pour chaque non-terminal $A \in N_R$, nous définissons un renommage h_A des sommets du membre droit H_A : $h_A(1) = A$, $h_A(2) = \varepsilon$, et l'image

$Im(h_A) - \{\varepsilon\}$ est un ensemble de symboles avec $Im(h_A) \cap Im(h_B) = \{\varepsilon\}$ pour tout $B \in N_R - \{A\}$. Nous définissons la grammaire algébrique $G = (T, N', P)$ suivante :

$$P \;:=\; \{\; (h_A(s)\,,\, ah_A(t)) \mid A \in N_R \;\wedge\; s \xrightarrow[R(A)]{a} t \;\}.$$

Notons que chaque membre droit de P est un mot de longueur au plus 2, et le nombre de non-terminaux de G dépend de la longueur de description de R :

$$|N'| \;=\; \big(\sum_{A \in N_R} |V_{R(A)}|\big) \;-\; |N_R|\,.$$

□

Par exemple, la grammaire de graphes de la figure 2.2 est transformée en la grammaire algébrique suivante :

$$\begin{array}{lllllllllll}
A & = & aC & ; & B & = & aF + bE & ; & C & = & BD \\
D & = & c & ; & E & = & aG + d & ; & F & = & AG \\
G & = & AH & ; & H & = & c
\end{array}$$

3

Grammaires de graphes et langages algébriques

3.1 Lemme des paires itérantes

3.1.1 Introduction

Le lemme des paires itérantes pour les langages algébriques, connu aussi sous les noms de *lemme de pompage* ou de *lemme de Shamir*, est une propriété élémentaire des langages algébriques. On utilise essentiellement ce lemme pour établir que des langages ne sont pas algébriques parce qu'ils ne satisfont pas la propriété. Dans ce chapitre, nous allons présenter une démonstration de ce lemme grâce aux grammaires de graphes.

3.1.2 Lemme des paires itérantes

Un langage algébrique est aussi le langage reconnu par un automate à pile avec acceptation sur pile vide. L'usage d'une pile induit une propriété de pompage sur les mots reconnus suffisamment longs : on pourra itérer un empilement-dépilement pour obtenir d'autres mots du langage.

Lemme 3 (Lemme des paires itérantes) *Si L est un langage algébrique alors il existe un entier n tel que tout mot $z \in L$ de longueur au moins n se factorise en $z = uvwxy$ avec $|vwx| \leq n$ et $v \neq \epsilon$ de sorte que pour tout $i \geq 0$ on a $uv^iwx^iy \in L$.*

On utilise la contraposée du lemme des paires itérantes pour établir la non algébricité d'un langage, à savoir

si

pour tout entier $n \geq 0$, il existe un mot $z \in L$ de longueur au moins n
tel que pour toute factorisation $uvwxy = z$ avec $|vwx| \leq n$ et $v \neq \epsilon$,
il existe $i \geq 0$ tel que $uv^iwx^iy \notin L$

alors L est non-algébrique.

3.1.3 Démonstration géométrique du lemme des paires itérantes

Soit L un langage algébrique.

Alors $L - \{\varepsilon\} = L_G(S)$ pour une grammaire algébrique $G = (T, N, P)$ en *forme normale de Greibach* : toute production est de la forme

$$A \longrightarrow w \text{ avec } A \in N \text{ et } w \in T.N^*.$$

Soit R la grammaire de graphes obtenue à partir de G par la construction de la proposition 1.

Considérons un graphe H engendré par R à partir de l'arc $(1, S, 2)$:

$$H = \bigcup_{n\geq 0}[H_n] \text{ avec } \{(1, S, 2)\} = H_0 \underset{R}{\Longrightarrow} \ldots \underset{R}{\Longrightarrow} H_n \underset{R}{\Longrightarrow} \ldots$$

Soit r le nombre de règles de P (ou de R).

Soit q la taille maximale des graphes membres droits de R :

$$q = max_{A\in N} \sum_{(A,U)\in P} |U|$$

Aussi $|H_0| = 1$ et $|H_1| \leq q$.

De plus, $|H_2| \leq q + (q-1)(q-1) \leq q^2$.

Par récurrence sur $n \geq 0$, on obtient

$$|H_n| \leq 1 + (q-1) + \ldots (q-1)^n \leq q^n.$$

On choisit

$$n = q^r$$

Soit z un mot de L de longueur au moins n.

Il existe un chemin $1 = s_0 \xrightarrow[H]{z(1)} s_1 \ldots \xrightarrow[H]{z(|z|)} s_{|z|} = 2$.

Comme R est sans circuit, le chemin est élémentaire :

$$s_i \neq s_j \text{ pour tout } 1 \leq i < j \leq |z|.$$

Soit ℓ_0 le plus petit entier tel que H_{ℓ_0} contienne ce chemin :

$$\ell_0 \;=\; max_{0\leq i<|z|} min\{\; j \mid s_i \in V_{H_j} \;\}.$$

Aussi

$$q^r \;\leq |z| \leq q^{\ell_0} \quad \text{d'où} \quad \ell_0 \geq r.$$

Soit i_0 le rang d'un arc terminal de niveau maximal :

$$s_{i_0} \overset{z(i_0+1)}{\underset{H_{\ell_0} - H_{\ell_0-1}}{\longrightarrow}} s_{i_0+1}.$$

Pour tout rang $0 < i \leq \ell_0$, on pose

$$i_1 \;=\; max\{\; j \leq i_0 \mid s_j \in V_{H_i - H_{i-1}} \;\}$$
$$i_2 \;=\; min\{\; j > i_0 \mid s_j \in V_{H_i - H_{i-1}} \;\}$$

que l'on complète au rang 0 avec $0_1 = 0$ et $0_2 = |z|$.

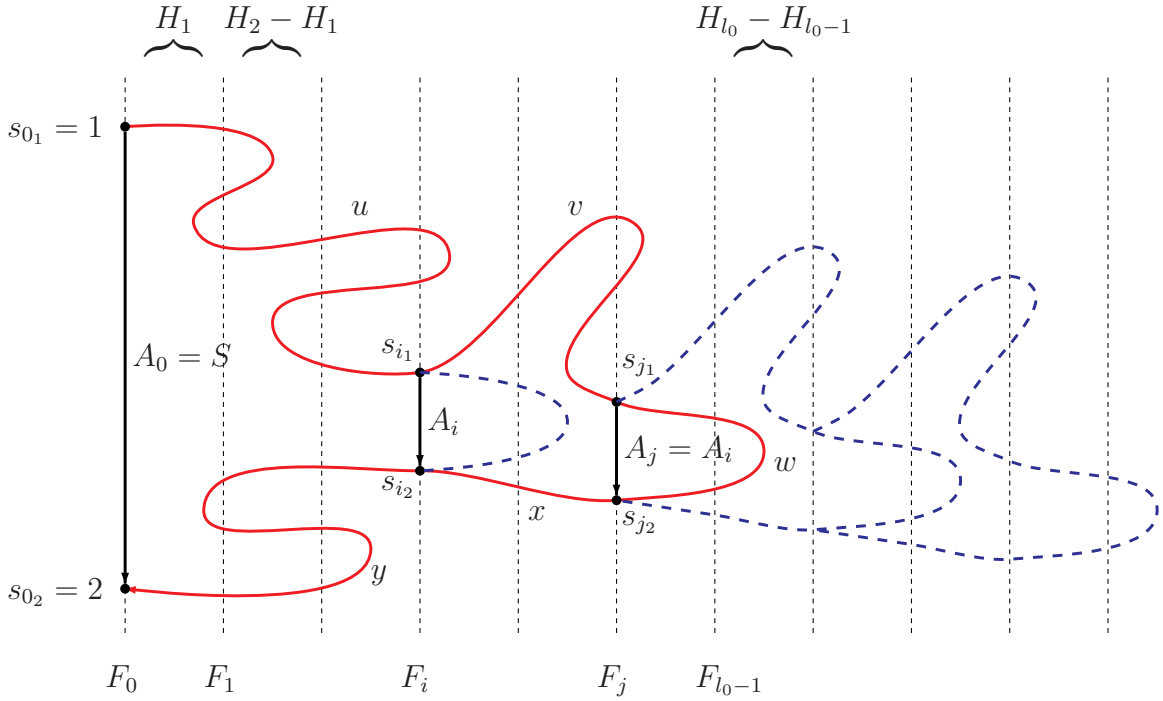


FIGURE 3.1 – Décomposition d'un mot accepté et itération.

Aussi et pour tout $0 \leq i \leq \ell_0$, il existe un non-terminal A_i tel que

$(s_{i_1}, A_i, s_{i_2}) \in H_i$

$s_{i_1} \overset{z(i_1+1)}{\longrightarrow} \ldots \overset{z(i_2)}{\longrightarrow} s_{i_2}$ est un chemin engendré par R à partir de $A_i s_{i_1} s_{i_2}$.

Comme $\ell_0 \geq r$, on prend $0 \leq i < j \leq r$ tel que $A_i = A_j$ avec i maximal.

Donc, le mot z peut être factorisé en $z = uvwxy$ où

$$u = z[1, i_1] \text{ avec } z[p, q] = z(p)\ldots z(q) \text{ pour } p \leq q,$$
$$v = z[i_1 + 1, j_1]\,;\, w = z[j_1 + 1, j_2]\,;\, x = z[j_2 + 1, i_2]\,;\, y = z[i_2 + 1, |z|].$$

Par maximalité de i, les non-terminaux $A_{i+1}, \ldots, A_{\ell_0}$ sont distincts 2 à 2, donc

$$\ell_0 - i \leq r.$$

Les lettres de w de rang ℓ_0 sont obtenues en remplaçant des arcs non-terminaux par des arcs terminaux. Aussi

$$|vwx| \leq q^{\ell_0 - i} \leq q^r = n.$$

Comme G est en forme normale de Greibach et $i \neq j$, on a $v \neq \varepsilon$.

Enfin et par remplacements itérés de A_i en vA_ix, on obtient

$$uv^m wx^m y \in L \text{ pour tout } m \geq 0.$$

□

3.2 Lemme de Parikh

3.2.1 Introduction

Comme le lemme des paires itérantes, le lemme de Parikh nous donne une condition nécessaire sur la distribution des lettres des mots d'un langage algébrique. Il a été introduit en 1966 [Pa 66]. Ce lemme a suscité un vif intérêt (voir [Pi 73]), et depuis presque quarante ans, différentes approches pour le démontrer ont été présentées (voir [HK 99] et [AEI 01]). Dans ce chapitre, nous présentons une démonstration géométrique du lemme de Parikh. Cette démonstration a été publiée dans [Di 06].

3.2.2 Lemme de Parikh

Pour commencer, nous rappelons les définitions des ensembles linéaires, des ensembles semi-linéaires, et également la définition de l'image commutative d'un langage (voir par exemple [Re 88] et [Ha 78]).

Un ensemble de la forme

$$\{\ \alpha_0 + k_1\alpha_1 + \cdots + k_n\alpha_n \mid k_1, \cdots, k_n \in \mathbb{N}\ \}$$

où $\alpha_0, \cdots, \alpha_n$ sont des élément de $\mathbb{N}^m$, est dit un sous-ensemble *linéaire* de $\mathbb{N}^m$. Un *ensemble semi-linéaire* est une union finie d'ensembles linéaires.

Comme $\mathbb{N}^m$ est un monoïde commutatif pour l'opération d'addition + composante à composante, ses ensembles semi-linéaires sont ses parties rationnelles.

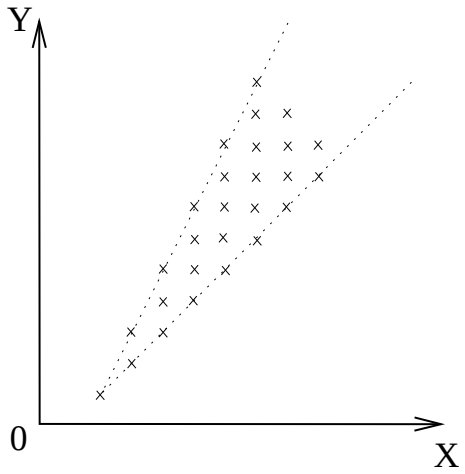


FIGURE 3.2 – Un ensemble linéaire.

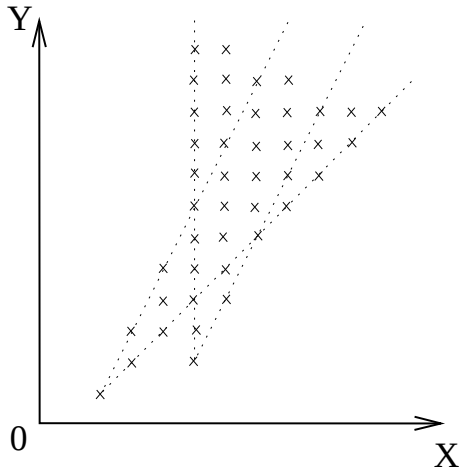


FIGURE 3.3 – Un ensemble semi-linéaire.

Aussi, ils forment une algèbre de Boole [GS 64] : la famille des ensembles semi-linéaires de $\mathbb{N}^m$ est fermée par union, intersection et complémentation.

Etant donné un alphabet $T = \{a_1, \cdots, a_m\}$, nous définissons l'application ψ de T^* dans $\mathbb{N}^m$ comme suit :

$$\psi(w) = (|w|_{a_1}, \cdots, |w|_{a_m})$$

où $|w|_{a_i}$ désigne le nombre d'occurrences de la lettre a_i dans le mot w.

On dit que $\psi(w)$ est l'*image de Parikh* du mot w.

Par exemple pour $T = \{a, b, c\}$, on a $\psi(abacac) = (3, 1, 2)$.

Soient $x, y \in T^*$ deux mots quelconques, nous avons

$$\begin{array}{rcl} \psi(x.y) & = & \psi(x) + \psi(y) \\ & = & \psi(y) + \psi(x) \\ & = & \psi(y.x) \end{array}$$

En particulier, ψ est un morphisme de T^* dans $\mathbb{N}^m$.

L'image de Parikh s'étend par union à tout langage $L \subseteq T^*$:

$$\psi(L) \ = \ \{\psi(x) \mid x \in L\}.$$

Comme la rationalité est préservée par morphisme, on a $\psi(L)$ rationnel pour tout langage rationnel L. La réciproque n'est pas vraie :

$$\psi^{-1}(\{(0,1) + k.(1,2) \mid k \geq 0\}) \cap a^*b^* \ = \ \{a^n b^{2n+1} | n \geq 0\}.$$

La semi-linéarité de l'image de Parikh d'un langage rationnel se généralise à tout langage algébrique [Pa 66].

Lemme 4 *(Lemme de Parikh). L'image de Parikh $\psi(L)$ de tout langage algébrique L est de façon effective un ensemble semi-linéaire.*

Soit G une grammaire algébrique qui engendre le langage L et soit $\psi(L)$ l'image de Parikh de L. Ainsi, $\psi(L)$ est un ensemble semi-linéaire de $\mathbb{N}^m$. De plus, une description de $\psi(L)$ sous la forme :

$$\psi(L) \ = \ Q_1 \cup Q_2 \cup \cdots \cup Q_n$$

où $Q_1, ..., Q_n$ sont des ensembles linéaires de $\mathbb{N}^m$, peut être obtenue par un algorithme (déterministe) à partir de G.

3.2.3 Démonstration géométrique du lemme de Parikh

Dorénavant R est une grammaire de graphes d'ensemble

$$N = \{B_1, ..., B_p\}$$

des non-terminaux. Soit

$$\mathrm{Tr} \ = \ \bigcup_{n \geq 0} H_n \ \text{ avec } \ \{(1, A, 2)\} \ = \ H_0 \underset{R}{\Longrightarrow} H_1 \ldots \underset{R}{\Longrightarrow} H_n \underset{R}{\Longrightarrow} \ldots$$

un graphe engendré par R à partir d'un axiome $A \in N$ et auquel on a adjoint ses arcs non-terminaux. On note

$$\mathrm{Tr}_{n+1} \;=\; H_{n+1} - H_n \;\text{ pour tout } n \geq 0.$$

L'ensemble des non-terminaux récrits pour engendrer un chemin β de Tr est noté $\Lambda(\beta)$; l'étiquette du chemin β se note $I(\beta)$.

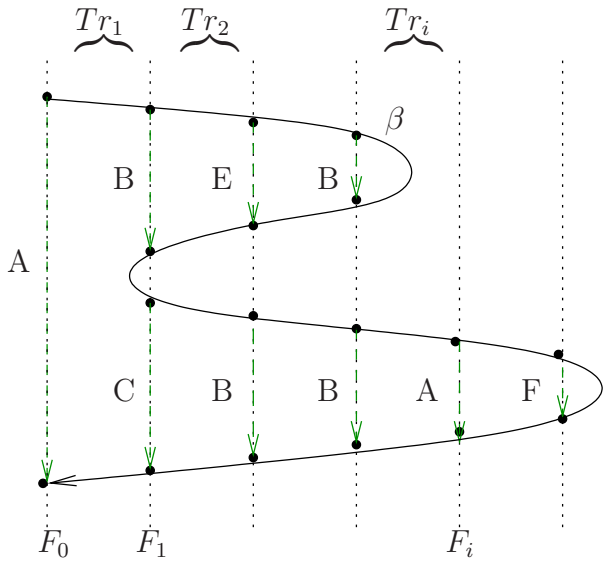


FIGURE 3.4 – Ensemble $\Lambda(\beta) = \{A, B, C, E, F\}$.

Une *réduction* d'un chemin α est l'opération qui enlève une partie β_B de α pour obtenir un autre chemin α', comme indiqué à la figure 3.5.

On dit que α *se réduit* en (α', β_B) et on note $\alpha \overset{reduire}{\longrightarrow} (\alpha', \beta_B)$; dans le cas où β_B n'est pas pertinent, on peut l'ignorer et écrire simplement $\alpha \overset{reduire}{\longrightarrow} \alpha'$.

On dit que α est un chemin *réductible* s'il peut être réduit, sinon α est dit *irréductible*.

Si α et α' appartiennent à un ensemble F de chemins, on dit que α est *réductible selon* F, sinon α est dit *irréductible selon* F.

Un ensemble F de chemins est dit *réductible* s'il contient au moins un chemin réductible selon F ; sinon, F est dit *irréductible*.

Remarque. Tout chemin irréductible engendré par une grammaire de graphes est de longueur bornée. Par conséquent, tout ensemble irréductible de chemins engendrés par une grammaire de graphes est fini.

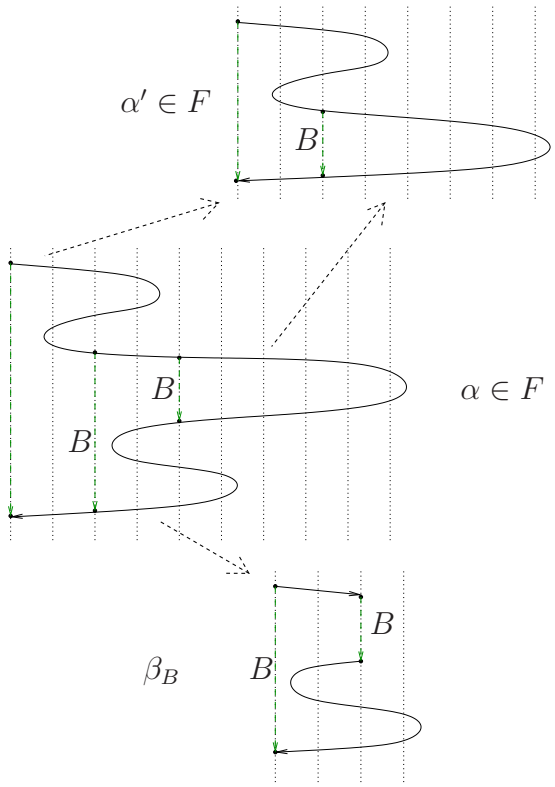


FIGURE 3.5 – Une réduction d'un chemin en un autre en retirant une répétition.

Pour tout sous-ensemble N' de N, on considère l'ensemble $\mathrm{Ch}_{N'}$ des chemins α de Tr dont l'ensemble $\Lambda(\alpha)$ est N'.

On note $L_{N'}$ l'ensemble des étiquettes des chemins de $\mathrm{Ch}_{N'}$:

$$L_{N'} \;=\; \{\, I(\beta) \mid \beta \in \mathrm{Ch}_{N'} \,\}.$$

On a

$$L = \bigcup_{N' \subseteq N} L_{N'}$$

Si l'axiome $A \notin N'$ alors $L_{N'}$ est vide. Aussi,

$$L = \bigcup_{N' \subseteq N, A \in N'} L_{N'}$$

Par conséquent, on a

$$\psi(L) = \bigcup_{N' \subseteq N, A \in N'} \psi(L_{N'})$$

Comme N est fini, c'est une union finie.

Par restrictions, nous pouvons nous contenter de ne montrer la semi-linéarité que de $\psi(L_N)$.

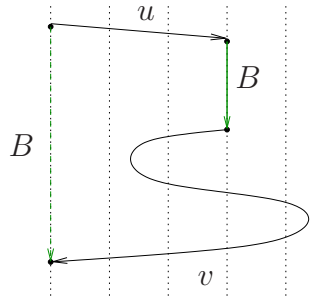


FIGURE 3.6 – Un chemin de l'ensemble R_B.

Pour chaque non-terminal B, nous définissons l'ensemble noté R_B des chemins β engendrés par R à partir de B et qui satisfont les deux conditions suivantes :

- β ne contient qu'un seul non-terminal B ;

- β n'est réductible qu'en B.

Ainsi, tout $\beta \in R_B$ est étiqueté par un mot de la forme

$$I(\beta) \;=\; uBv \;\text{ où }\; u, v \in T^*.$$

On note L_B le langage suivant :

$$L_B \;=\; \{\, uv \mid \exists\, \beta \in R_B \;(I(\beta) = uBv)\;\}.$$

Nous définissons l'ensemble Cht des chemins terminaux α engendrés par R à partir de l'axiome A de sorte que :

- tous les non-terminaux sont présents dans $\Lambda(\alpha)$;

- si $\alpha \overset{reduire}{\longrightarrow} \alpha'$ alors au moins un non-terminal n'est pas présent dans $\Lambda(\alpha')$.

On note L_A le langage des étiquettes des chemins dans Cht :

$$L_A \;=\; \{\, I(\beta) \mid \beta \in \text{Cht}\,\}.$$

Ainsi les langages L_{B_i} et L_A sont finis. Nous allons montrer que

$$\psi(L_N) \;=\; \psi(L_A.L_{B_1}^*.L_{B_2}^*...L_{B_p}^*) \tag{3.1}$$

Nous montrons d'abord que

$$\psi(L_A.L_{B_1}^*.L_{B_2}^*...L_{B_p}^*) \;\subseteq\; \psi(L_N) \tag{3.2}$$

puis l'inclusion inverse :

$$\psi(L_N) \subseteq \psi(L_A.L_{B_1}^*.L_{B_2}^*...L_{B_p}^*). \tag{3.3}$$

i) Vérifions l'inclusion (3.2).

Tout d'abord, nous avons $L_A \subseteq L_N$, donc

$$\psi(L_A) \subseteq \psi(L_N). \tag{3.4}$$

De plus, soit β un chemin dans R_B pour un $B \in N$. Soit α un chemin dans Cht.

Par définition de Cht, on a $\Lambda(\alpha) = N$, donc $B \in \Lambda(\alpha)$.

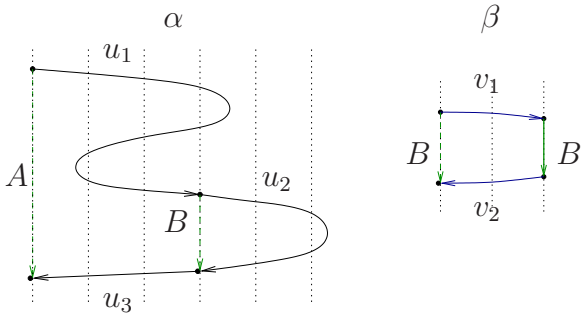


FIGURE 3.7 – Un chemin $\alpha \in$ Cht et un chemin $\beta \in R_B$.

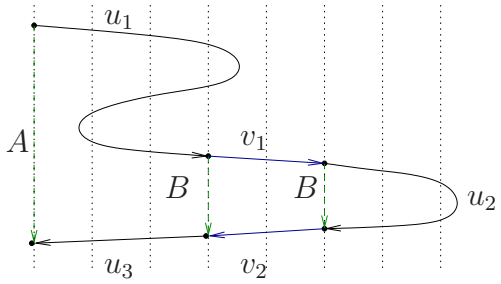


FIGURE 3.8 – Le chemin α' obtenu à partir de α par insertion de β.

Nous avons $I(\beta) = v_1.B.v_2$ et $I(\alpha) = u_1.u_2.u_3$.

Donc, nous pouvons construire un chemin α' obtenu à partir de α par insertion de β comme indiqué à la figure 3.8.

Alors $I(\alpha') = u_1.v_1.u_2.v_2.u_3$. Ainsi,

$$\Lambda(\alpha') = \Lambda(\alpha) \cup \Lambda(\beta) = N \cup \Lambda(\beta) = N.$$

Alors $\alpha' \in \mathrm{Ch}_N$, donc $I(\alpha') \in L_N$. Par conséquent,

$$\begin{array}{rcl} \psi(uv) & = & \psi(u_1.u_2.u_3.v_1.v_2) \\ & = & \psi(u_1.v_1.u_2.v_2.u_3) \\ & = & \psi(I(\alpha')) \\ & \in & \psi(L_N). \end{array}$$

Alors pour tout $u \in L_A, B \in N, v \in L_B$, on a

$$\psi(uv) \ \in \ \psi(L_N).$$

Donc,

$$\psi(L_A.L_B) \ \subseteq \ \psi(L_N) \ \text{ pour tout } B \in N. \tag{3.5}$$

De (3.4) et (3.5) découlent l'inclusion (3.2) :

$$\psi(L_A.L^*_{B_1}.L^*_{B_2}...L^*_{B_p}) \ \subseteq \ \psi(L_N).$$

ii) Vérifions l'inclusion (3.3).

Pour tout chemin $\alpha \in \mathrm{Ch}_N$, on note n_α le nombre maximum de réductions selon Ch_N à partir de α.

Montrons par récurrence sur $n_\alpha \geq 0$ que

$$\psi(I(\alpha)) \ \subseteq \ \psi(L_A.L^*_{B_1}.L^*_{B_2}...L^*_{B_p}) \ \text{ pour tout } \ \alpha \in \mathrm{Ch}_N\,.$$

$n_\alpha = 0$: On a $\alpha \in \mathrm{Cht}$ donc $I(\alpha) \in L_A$.

$n_\alpha > 0$: Le chemin α se récrit en (α', β) avec $\alpha' \in \mathrm{Ch}_N$ et $n_{\alpha'} = n_\alpha - 1$, $\beta \in R_B$ pour un non-terminal B.

Aussi $\beta = uBv$ avec $uv \in L_B$. Donc

$$\begin{array}{rcll} \psi(\alpha) & = & \psi(\alpha').\psi(uv) & \text{par définition de la récriture} \\ & \in & \psi(L_A.L^*_{B_1}.L^*_{B_2}...L^*_{B_p}).\psi(L_B) & \text{par hypothèse de récurrence} \\ & \in & \psi(L_A.L^*_{B_1}.L^*_{B_2}...L^*_{B_p})\,. & \end{array}$$

Ceci termine la récurrence, d'où l'inclusion (3.3) :

$$\psi(L_N) \ \subseteq \ \psi(L_A.L^*_{B_1}.L^*_{B_2}...L^*_{B_p}).$$

iii) De (i) et (ii), on a donc l'égalité (3.1) :

$$\psi(L_N) \;=\; \psi(L_A.L^*_{B_1}.L^*_{B_2}...L^*_{B_p})$$

Ainsi $\psi(L_N)$ est semi-linéaire.

Par restriction, $\psi(L_{N'})$ est semi-linéaire pour toute partie N' de N.

En définitive, nous avons obtenu le résultat suivant :

$$\psi(L) \;=\; \bigcup_{N' \subseteq N, A \in N'} \psi(L_{N'})$$

est semi-linéaire.

□

4

Plus courts chemins

4.1 Calculs avec des grammaires de graphes

Un graphe est un ensemble dénombrable d'arcs étiquetés. Dans ce chapitre, les étiquettes des arcs sont des éléments d'un demi-anneau idempotent et continu. La valeur d'un chemin est le produit de ses étiquettes successives. La valeur d'un graphe (à partir des sommets initiaux aux sommets finaux) est la somme des valeurs de ses chemins d'un sommet initial à un sommet final.

Nous voulons calculer les valeurs des graphes engendrés par une grammaire déterministe de graphes. On rappelle que nous nous sommes limités à des grammaires pour lesquelles chaque membre gauche est un arc étiqueté du sommet 1 au sommet 2, et il n'y a pas dans les membres droits d'arc sortant de 2, ni d'arc entrant en 1. La valeur de tout arc non-terminal du membre gauche est la valeur de 1 à 2 de son graphe engendré. La valeur d'une grammaire est le vecteur des valeurs de ses membres gauches.

En considérant une grammaire comme une fonction de et vers des demi-anneaux (à la puissance du nombre des règles), son application itérée à partir du plus petit élément donne par borne supérieure la valeur de la grammaire (théorème 5).

Rappelons qu'une structure algébrique $(K, +, \cdot, 0, 1)$ est un *demi-anneau* si :

- $(K, +, 0)$ est un monoïde commutatif ;
- $(K, \cdot, 1)$ est un monoïde ;
- la *multiplication* $\cdot$ est distributive par rapport à l'*addition* $+$;
- 0 est absorbant pour la multiplication $\cdot$: $0 \cdot a \ = \ a \cdot 0 \ = \ 0$ pour tout $a \in K$.

On dit qu'un demi-anneau K est *complet* si pour la relation suivante :

$$a \leq b \quad \text{si} \quad a \ + \ c \ = \ b \ \text{ pour un certain } c \in K,$$

toute suite croissante $a_0 \ \leq \ a_1 \ \leq \ \cdots \ \leq \ a_n \ \leq \ \cdots$ a une borne supérieure $\bigvee_n a_n$.

Cela implique que $\leq$ est un pré-ordre, 0 est le plus petit élément, et les opérations $+$ et $\cdot$ sont croissantes, à savoir :

$$(a \leq b \ \wedge \ a' \leq b') \quad \Longrightarrow \quad (a + a' \leq b + b' \ \wedge \ a \cdot a' \leq b \cdot b').$$

Cela permet aussi de définir la somme d'une suite $(a_n)_{n\geq 0}$ dans K par :

$$\sum_{n\geq 0} a_n \ := \ \bigvee_{n\geq 0} (\sum_{i=0}^{n} a_i).$$

On dit qu'un demi-anneau K est *idempotent* si $+$ est idempotent :

$a + a = a$ pour tout $a \in K$ ou de façon équivalente $1 + 1 = 1$.

Pour K complet et idempotent, nous avons $\sum_{n\geq 0} a_n \ = \ \sum_{n\geq 0} a_{\pi(n)}$ pour toute permutation π, et cette somme est également notée

$$\sum \{a_n | n \geq 0\}.$$

On dit qu'un demi-anneau complet K est *continu* si $+$ et $\cdot$ sont des opérations continues *i.e.* elles sont commutatives avec $\bigvee$:

pour toutes suites croisantes $(a_n)_{n\geq 0}$ et $(b_n)_{n\geq 0}$,

$$(\bigvee_{n\geq 0} a_n) + (\bigvee_{n\geq 0} b_n) \ = \ \bigvee_{n\geq 0} (a_n + b_n)$$

$$(\bigvee_{n\geq 0} a_n) \cdot (\bigvee_{n\geq 0} b_n) \ = \ \bigvee_{n\geq 0} (a_n \cdot b_n).$$

Donc,

$$(\sum_{n\geq 0} a_n) \ + \ (\sum_{n\geq 0} b_n) \ = \ \sum_{n\geq 0} (a_n + b_n), \quad \text{pour tout} \quad a_n, b_n \in K.$$

Pour K continu et idempotent et pour toutes suites $(a_n)_{n\geq 0}$ et $(b_n)_{n\geq 0}$, on a

$$(\sum_{n\geq 0} a_n) \ \cdot \ (\sum_{n\geq 0} b_n) \ = \ \sum \{a_i \cdot b_i \mid i, j \geq 0\}.$$

Désormais, K désigne un demi-anneau complet et idempotent.

Rappelons que $(K^*, \cdot, \epsilon)$ est le monoïde libre engendré par l'ensemble K *i.e.* $K^* \ = \ \bigcup_{p\geq 0} K^p$ où K^p est l'ensemble des p-uplets d'éléments de K, et $\cdot$ est la

concaténation dont l'élément neutre ϵ est le 0-uplet $()$. Tout p-uplet $(a_1, \cdots, a_p)$ est écrit $a_1 \dots a_p$.

La relation d'identité sur K est étendue en le morphisme $[\![\ ]\!]$ de $(K^*, \cdot, \epsilon)$ vers le monoïde $(K, \cdot, 1)$:

$$[\![\ \epsilon\ ]\!] = 1 \ \ ; \ \ \forall a \in K,\ [\![\ a\ ]\!] = a \ \ ; \ \ \forall u, v \in K^*,\ [\![\ u.v\ ]\!] \ = \ [\![\ u\ ]\!] \cdot [\![\ v\ ]\!].$$

Nous étendons par sommation l'application valeur $[\![\ ]\!]$ à tout langage :

$$[\![\ U\ ]\!] \ := \ \sum \{ [\![\ u\ ]\!] \mid u \in U \} \ \text{ pour tout } U \subseteq K^*.$$

Soit L un ensemble quelconque. Rappellons qu'un *L-graphe* est un ensemble d'arcs étiquetés dans L. Plus précisément, un L-graphe G est un sous-ensemble de $V \times L \times V$ où V est un ensemble quelconque tel que l'ensemble des *sommets* de G défini par

$$V_G \ := \ \{\, s \mid \exists a, t, (s, a, t) \in G \vee (t, a, s) \in G \,\}$$

est fini ou dénombrable, et l'ensemble de ses *étiquettes*

$$L_G \ := \ \{\, a \in L \mid \exists s, t, (s, a, t) \in G \,\} \ \text{ est fini.}$$

Tout (s, a, t) de G est un *arc étiqueté* de *source* s, de *but* t, et d'étiquette a ; il est identifié avec la transition étiquetée $s \xrightarrow[G]{a} t$, ou directement $s \xrightarrow{a} t$ si G est sous-entendu. La transformation d'un graphe G par une fonction h de V_G dans un ensemble V est le graphe

$$h(G) \ := \ \{\ h(s) \xrightarrow{a} h(t) \mid s \xrightarrow[G]{a} t \ \wedge \ s, t \in Dom(h)\ \}.$$

Rappelons qu'un *isomorphisme* h d'un graphe G sur un graphe H est une bijection de V_G sur V_H telle que $h(G) = H$. De plus, le *langage reconnu* par G à partir d'un ensemble I de sommets à un ensemble F de sommets est l'ensemble des étiquettes des chemins de I à F :

$$L(G, I, F) \ := \ \{\, a_1 \cdots a_n \mid n \geq 0 \ \wedge \ \exists\ s_0, \cdots, s_n$$
$$(s_0 \in I \ \wedge \ s_n \in F \ \wedge \ s_0 \xrightarrow{a_1} s_1 \cdots s_{n-1} \xrightarrow{a_n} s_n) \,\}.$$

En particulier, on a $\epsilon \in L(G, I, F)$ pour $I \cap F \neq \emptyset$.

Nous écrivons aussi $s \xrightarrow[G]{}{}^* t$ pour $L(G, s, t) \neq \emptyset$, et $s \xRightarrow[G]{u} t$ pour $u \in L(G, s, t)$.

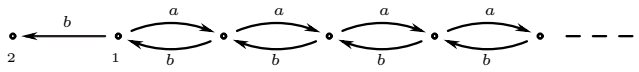
La *valeur d'un K-graphe* G de $I \subseteq V_G$ à $F \subseteq V_G$ est définie comme la valeur de son langage reconnu :

$$[\![\ G, I, F\ ]\!] \ := \ [\![\ L(G, I, F)\ ]\!].$$

Pour tout graphe G et en identifiant toute étiquette $a \in L_G$ avec $\{a\}$, nous pouvons prendre le demi-anneau $(2^{L_G^*}, \cup, \cdot, \emptyset, \{\epsilon\})$ des langages pour lesquels $[\![u]\!] = \{u\}$ pour tout $u \in L_G^*$. Pour ce demi-anneau, les valeurs de G sont les langages reconnus :

$$[\![G, I, F]\!] = L(G, I, F) \text{ pour tout } I, F \subseteq L_G^*.$$

Exemple. Nous considérons le graphe G ci-dessous :



a) En identifiant a avec $\{a\}$ et b avec $\{b\}$, la valeur $[\![G, 1, 2]\!]$ de G du sommet 1 au sommet 2 du demi-anneau $(2^{\{a,b\}^*}, \cup, \cdot, \emptyset, \{\epsilon\})$ est le *langage de Lukasiewicz*.

b) Maintenant, en prenant le demi-anneau $(2^{\mathbb{N}\times\mathbb{N}}, \cup, +, \emptyset, (0,0))$ avec $a = \{(1,0)\}$ et $b = \{(0,1)\}$, la valeur $[\![G, 1, 2]\!]$ est l'image de Parikh $\{(n, n+1) | n \geq 0\}$ du langage de Lukasiewicz.

c) En prenant $a, b \in \mathbb{R}$ et le demi-anneau $(\mathbb{R} \cup \{-\omega, \omega\}, Min, +, \omega, 0)$, la valeur

$$[\![G, 1, 2]\!] = \begin{cases} b & \text{si } a + b \geq 0, \\ -\omega & \text{sinon,} \end{cases}$$

est la plus petite valeur de l'étiquetage des chemins de 1 à 2.

d) Enfin ayant a, b dans l'ensemble $\mathbb{R}_+$ des nombres réels non négatifs et pour le demi-anneau $(\mathbb{R}_+ \cup \{\omega\}, Max, \times, 0, 1)$, la valeur de 1 à 2 est

$$[\![G, 1, 2]\!] = \begin{cases} b & \text{si } a \times b \leq 1, \\ \omega & \text{sinon.} \end{cases}$$

Pour tout demi-anneau idempotent et continu, nous voulons étendre aux graphes engendrés par les grammaires de graphes, les algorithmes de base calculant les valeurs pour des graphes finis.

Une grammaire R sur K signifie que $T_R \subseteq K$.

Pour toute grammaire R et tout non-terminal $A \in N_R$,

son membre droit est $R(\{(1, A, 2)\})$, également noté $R(A)$;

le graphe engendré par R à partir de A est $R^\omega(\{(1, A, 2)\})$, également noté $R^\omega(A)$;

et si $T_R \subseteq K$, la valeur de A par R est $[\![R^\omega(A), 1, 2]\!]$ également notée $[\![R^\omega(A)]\!]$.

Etant donnée une grammaire R sur K, nous voulons calculer $[\![R]\!]^\omega(A)$ pour tout non-terminal A.

Tout d'abord, nous mettons R sous la forme réduite S suivante :

$$N_S \;=\; \{\; A \in N_R \mid L(R^\omega(A), 1, 2) \neq \emptyset \;\}\;\; \text{et}\;\; 0 \notin T_S,$$

de sorte que pour tout $A \in N_S$, nous avons :

$$[\![\; S^\omega(A) \;]\!] \;=\; [\![\; R^\omega(A) \;]\!]\;\; \text{et}\;\; 1 \underset{s(A)}{\longrightarrow}{}^* s \underset{s(A)}{\longrightarrow}{}^* 2\;\; \text{pour tout } s \in V_{S(A)}.$$

Pour cela, nous commençons par supprimer les arcs étiquetés par 0 dans les membres droits de R. Ensuite, nous calculons l'ensemble

$$E \;=\; \{\; A \in N_R \mid L(R^\omega(A), 1, 2) \neq \emptyset \;\}$$

des non-terminaux ayant un chemin de 1 à 2 dans le graphe engendré. Cet ensemble E est le plus petit point fixe de l'équation suivante :

$$E \;=\; \{\, A \in N_R \mid \exists u \in (T_R \cup E)^*,\; 1 \underset{R(A)}{\overset{u}{\Longrightarrow}} 2) \,\}.$$

Cela nous permet de limiter les règles de R aux non-terminaux dans E et de supprimer les arcs étiquetés par un non-terminal n'appartenant pas à E dans les membres droits de la grammaire.

Enfin, nous obtenons S en limitant chaque membre droit aux sommets accessibles de 1 et co-accessibles de 2.

La complexité de cet algorithme est quadratique selon la longueur de description de R (due au calcul de E).

Désormais, nous supposons que toute grammaire R est sous forme réduite. Dans ce cas,

$$R \;\text{ est sans circuit} \quad\Longleftrightarrow\quad R^\omega(A) \;\text{ est sans circuit pour tout } A \in N_R.$$

Etant donnée une grammaire R sur K, nous ordonnons l'ensemble

$$N_R \;=\; \{A_1, \cdots, A_p\}$$

de ses p non-terminaux. Nous souhaitons déterminer *la valeur de la grammaire* $[\![\; R^\omega \;]\!]$ de R comme étant le p-uplet suivant dans K^p :

$$[\![\; R^\omega \;]\!] \;:=\; ([\![\; R^\omega(A_1) \;]\!], \cdots, [\![\; R^\omega(A_p) \;]\!]).$$

Une méthode sémantique habituelle est de définir l'*interprétation* $[\![\; R \;]\!]$ de R comme étant le morphisme :

$$\begin{array}{rccl} [\![\; R \;]\!] : & K^p & \longrightarrow & K^p \\ & (a_1, \cdots, a_p) & \longmapsto & ([\![\; R(A_1)[a_1, \cdots, a_p] \;]\!], \cdots, [\![\; R(A_p)[a_1, \cdots, a_p] \;]\!]) \end{array}$$

où $G[a_1, \cdots, a_p]$ est le graphe obtenu à partir du graphe G en remplaçant chaque étiquette A_i par a_i pour chaque $1 \leq i \leq p$. Cette interprétation $[\![R]\!]$ est un morphisme continu.

Comme K^p est un ensemble complet pour l'ordre produit dont le plus petit élément est $0 = (0, \cdots, 0)$, nous pouvons appliquer le théorème de Knaster-Tarski [Kn 28, Ta 55] : la borne supérieure de l'application itérée de $[\![R]\!]$ à partir de 0, est le plus petit point fixe de $[\![R]\!]$ noté $\mu[\![R]\!]$. Un premier résultat est que cette borne supérieure est également la valeur $[\![R^\omega]\!]$ de R.

Rappelons que K est un demi-anneau complet et idempotent.
Nous commençons par quelques propriétés de base des valeurs du graphe.

Lemme 4.1.1 *Nous avons les propriétés suivantes :*

a) $H \subseteq G \implies [\![H,I,F]\!] \leq [\![G, I, F]\!]$

b) $[\![\bigcup_{n\geq 0} G_n, I, F]\!] = \bigvee_{n\geq 0} [\![G_n, I, F]\!]$ *pour toute suite croissante de graphes* $G_0 \subseteq G_1 \subseteq \ldots \subseteq G_n \subseteq \ldots$

c) $[\![G, I, F]\!] = [\![G - H, I, F]\!]$ *pour tout* $H \subseteq V_G \times \{0\} \times V_G$.

Démonstration. i) Soit $H \subseteq G$. Donc $L(H, I, F) \subseteq L(G, I, F)$ et

$$\begin{aligned} [\![G, I, F]\!] &= \sum \{ [\![w]\!] \mid w \in L(G, I, F) \} \\ &= [\![H, I, F]\!] + \sum \{ [\![w]\!] \mid w \in L(G, I, F) - L(H, I, F) \}. \end{aligned}$$

Par conséquent, $[\![\mathrm{H,I,F}]\!] \leq [\![G, I, F]\!]$.

ii) Soit $G_0 \subseteq G_1 \subseteq \ldots \subseteq G_n \subseteq \ldots$

Donc $L(G_0, I, F) \subseteq L(G_1, I, F) \subseteq \ldots \subseteq L(G_n, I, F) \subseteq \ldots$

et sa borne supérieure est $\bigcup_{n\geq 0} L(G_n, I, F) = L(\bigcup_{n\geq 0} G_n, I, F)$.

Pour chaque $n \geq 0$, soit

$$a_n := \sum \{ [\![w]\!] \mid w \in L(G_n, I, F) - L(G_{n-1}, I, F) \}$$

avec $G_{-1} = \emptyset$. Ainsi

$$\begin{aligned} [\![\bigcup_{n\geq 0} G_n, I, F]\!] &= \sum \{ [\![w]\!] \mid w \in \bigcup_{n\geq 0} L(G_n, I, F) \} \\ &= \sum_{n\geq 0} a_n \\ &= \bigvee_{n\geq 0} \left(\sum_{i=0}^{n} a_i \right) \\ &= \bigvee_{n\geq 0} \sum \{ [\![w]\!] \mid w \in L(G_n, I, F) \} \\ &= \bigvee_{n\geq 0} [\![G_n, I, F]\!]. \end{aligned}$$

iii) Soit $H \subseteq V_G \times \{0\} \times V_G$.

Pour tout chemin $w \in L(G,I,F) - L(G-H,I,F)$ passant par au moins un arc de H, nous avons $[\![w]\!] = 0$ parce que 0 est absorbant pour l'opération $\cdot$. Alors

$$\begin{aligned} [\![G,I,F]\!] &= [\![G-H,I,F]\!] + \sum \{ [\![w]\!] \mid w \in L(G,I,F) - L(G-H,I,F) \} \\ &= [\![G-H,I,F]\!] \quad \text{car } 0 \text{ est l'identité pour l'opération } + . \end{aligned}$$

□

L'application d'évaluation $[\![\ ]\!]$ est un morphisme sur les langages finis, et sur tout les langages si le demi-anneau est continu.

Lemme 4.1.2 *Nous avons* $[\![U.V]\!] = [\![U]\!] \cdot [\![V]\!]$ *pour tout* $U, V \subseteq K^*$ *finis. Cette égalité est valable pour tous les langages où* K *est continu.*

Démonstration. En raison de la distributivité de $\cdot$ par rapport à $+$, nous avons

$$\Big(\sum_{i=1}^{m} a_i\Big) \cdot \Big(\sum_{j=1}^{n} b_j\Big) = \sum \{ a_i \cdot b_j \mid 1 \le i \le m \ \wedge \ 1 \le j \le n \}$$

pour tout $m, n \ge 0$ et $a_1, \ldots, a_m, b_1, \ldots, b_n \in K$. Ainsi

$$\begin{aligned} [\![U.V]\!] &= \sum \{ [\![uv]\!] \mid u \in U \ \wedge \ v \in V \} \\ &= \sum \{ [\![u]\!] \cdot [\![v]\!] \mid u \in U \ \wedge \ v \in V \} \\ &= \sum \{ [\![u]\!] \mid u \in U \} \cdot \sum \{ [\![v]\!] \mid v \in V \} \quad U, V \text{ finis et la distribut.} \\ &= [\![U]\!] \cdot [\![V]\!] . \end{aligned}$$

Si K est continu, l'égalité ci-dessus est vraie pour $m = n = \omega$.
□

Donnons quelques notions de base pour la substitution sur les graphes et sur les langages.

Soit L un ensemble quelconque d'étiquettes. Nous fixons un sous-ensemble $N = \{A_1, \ldots, A_p\} \subseteq L$ de symboles à remplacer.

La *substitution de graphe* $G[a_1, \ldots, a_p]$ est le graphe obtenu à partir d'un L-graphe G en remplaçant chaque étiquette A_i par $a_i \in L$ *i.e.*

$$\begin{aligned} G[a_1, \ldots, a_p] &:= \{ s \xrightarrow[G]{a} t \mid a \notin \{A_1, \ldots, A_p\} \} \\ &\cup \{ s \xrightarrow{a_i} t \mid 1 \le i \le p \ \wedge \ s \xrightarrow[G]{A_i} t \} . \end{aligned}$$

La *substitution de langage* $u[L_1, \ldots, L_p]$ est le langage obtenu à partir d'un mot $u \in L^*$ en remplaçant chaque lettre A_i par $L_i \subseteq L^*$ *i.e.*

$$\begin{array}{rrcl} \text{par morphisme} & (uv)[L_1, \ldots, L_p] & = & u[L_1, \ldots, L_p] \,.\, v[L_1, \ldots, L_p] \\ \text{avec} & \varepsilon[L_1, \ldots, L_p] & = & \varepsilon \\ & A_i[L_1, \ldots, L_p] & = & L_i \quad \text{pour chaque } \ 1 \leq i \leq p \\ & a[L_1, \ldots, L_p] & = & a \quad \text{pour chaque } \ a \in L - N. \end{array}$$

Nous étendons par union la substitution à tout langage $M \subseteq L^*$:

$$M[L_1, \ldots, L_p] \ := \bigcup \{ \ u[L_1, \ldots, L_p] \mid u \in M \ \}.$$

Le langage d'une substitution de graphe est la substitution du langage de graphe.

Lemme 4.1.3 *Pour tout* $a_1, \ldots, a_p \in L$,

$$L(G[a_1, \ldots, a_p], I, F) \ = \ L(G, I, F)\,[a_1, \ldots, a_p]\,.$$

Démonstration. L'inclusion directe résulte de

$$s \xrightarrow[G[a_1, \ldots, a_p]]{a} t \quad \Longrightarrow \quad \exists\, b \in L,\ b[a_1, \ldots, a_p] = a \ \wedge \ s \xrightarrow[G]{b} t\,.$$

L'inclusion inverse est une conséquence de $s \xrightarrow[G]{a} t \quad \Longrightarrow \quad s \xrightarrow[G[a_1, \ldots, a_p]]{a[a_1, \ldots, a_p]} t\,.$

□

La substitution permet d'exprimer le changement des langages reconnus en appliquant une récriture en parallèle.

Lemme 4.1.4 *Pour toute grammaire (déterministe de graphes)* R *avec* $N_R = \{A_1, \ldots, A_p\}$, *et pour tout* $G \underset{R}{\Longrightarrow} H$ *avec* $I, F \subseteq V_G$, *nous avons*

$$L(H, I, F) \ = \ L(G, I, F)\,[L(R(A_1), 1, 2), \ldots, L(R(A_p), 1, 2)]\,.$$

Démonstration. L'inclusion inverse est vérifiée car

$$\big(s \xrightarrow[G]{a} t \ \wedge \ a \notin N_R \ \Longrightarrow \ s \xrightarrow[H]{a} t\big)$$

et

$$\big(s \xrightarrow[G]{a} t \ \wedge \ a \in N_R \ \Longrightarrow \ s \xrightarrow[H]{L(R(a),1,2)} t\big).$$

L'inclusion directe résulte du fait que I et F sont des ensembles de sommets de

G, et que tout membre droit de R n'a pas d'arc de source 2, ni d'arc de but 1. Précisément, soit u un mot étiquetant un chemin dans H de I à F :

il existe un chemin $s_0 \xrightarrow[H]{a_1} s_1 \ \ldots\ s_{m-1} \xrightarrow[H]{a_m} s_m$

avec $u = a_1 \ldots a_m$, $s_0 \in I$ et $s_m \in F$.

Soit $I \ = \ \{\ 0 \leq i \leq m \mid s_i \in V_G\ \}$.

Pour deux nombres consécutifs $i, j \in I$ *i.e.* $(i < j \ \wedge\ \forall\, i < k < j,\ s_k \notin I)$, il existe

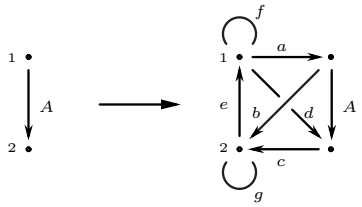
$$s_i \xrightarrow[G]{b_j} s_j \ \text{ tel que } \ (b_j \in N_R \ \wedge\ 1 \overset{a_i \ldots a_{j-1}}{\underset{R(b_j)}{\Longrightarrow}} 2) \ \vee\ (b_j \notin N_R \ \wedge\ j = i+1 \ \wedge\ b_j = a_i).$$

Pour $I \ = \ \{0\,,\, i_1\,,\, \ldots\,,\, i_n\}$ avec $0 < i_1 < \ldots < i_n = m$, le mot

$$v \ := b_{i_1} \ldots b_{i_n} \ \in L(G, I, F) \ \text{ et } \ v\,[L(R(A_1), 1, 2), \ldots, L(R(A_p), 1, 2)] \ = \ u.$$

□

Il est à noter que le lemme 4.1.4 peut être faux si nous permettons dans le membre droit de la grammaire un arc de but 1 ou de source 2. Par exemple, en prenant la grammaire R réduite à la règle suivante :



et en faisant la récriture parallèle $R(A) \underset{R}{\Longrightarrow} H$, nous avons

$$aec\,,\, afb\,,\, dgc \ \in \ L(H, 1, 2) \ - \ L(R(A), 1, 2)[L(R(A), 1, 2)].$$

La valeur d'une substitution peut être obtenue par calcul des valeurs des langages substitués.

Lemme 4.1.5 *Pour tout $M \subseteq (K \cup N)^*$ et pour tous langages finis $L_1, \ldots, L_p \subseteq K^*$, nous avons*

$$[\![\, M[L_1, \ldots, L_p] \,]\!] \ = \ [\![\, M\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!]\,.$$

Cette égalité est valable pour tous les langages si K est continu.

Démonstration. i) Montrons que

$$[\![\, u[L_1, \ldots, L_p] \,]\!] \;=\; [\![\, u\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] \quad \text{pour tout } u \in (K \cup N)^*.$$

La démonstration est faite par récurrence sur $|u| \geq 0$.

$|u| = 0$: Alors $u = \varepsilon$ et les deux termes sont égaux à $[\![\, \varepsilon \,]\!] = 1$.

$|u| = 1$: Soit $u \in K$ et les deux termes sont égaux à $[\![\, u \,]\!] = u$.

Soit $u = A_i$ pour un $1 \leq i \leq p$, et les deux termes sont égaux à $[\![\, L_i \,]\!]$.

$|u| > 1 : u = av$ pour $a \in K \cup N$. Alors

$$\begin{array}{rcl}
[\![\, u[L_1, \ldots, L_p] \,]\!] & = & [\![\, (av)[L_1, \ldots, L_p] \,]\!] \;=\; [\![\, a[L_1, \ldots, L_p] \,.\, v[L_1, \ldots, L_p] \,]\!] \\
& = & [\![\, a[L_1, \ldots, L_p] \,]\!] \cdot [\![\, v[L_1, \ldots, L_p] \,]\!] \quad \text{par le lemme 4.1.2} \\
& = & [\![\, a\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] \cdot [\![\, v\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] \quad \text{par hyp. réc.} \\
& = & [\![\, a\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,.\, v\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] \quad \text{par le lemme 4.1.2} \\
& = & [\![\, (av)\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] \;=\; [\![\, u\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] .
\end{array}$$

ii) Il s'en suit que

$$\begin{array}{rcl}
[\![\, M[L_1, \ldots, L_p] \,]\!] & = & \sum \{\, [\![\, u[L_1, \ldots, L_p] \,]\!] \mid u \in M \,\} \\
& = & \sum \{\, [\![\, u\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] \mid u \in M \,\} \quad \text{par } (i) \\
& = & [\![\, M\,[[\![\, L_1 \,]\!], \ldots, [\![\, L_p \,]\!]] \,]\!] .
\end{array}$$

□

Pour toute grammaire (déterministe de graphes) R et S telle que $N_R \subseteq N_S$, nous définissons la *composition* $R \circ S$ de R par S comme étant une grammaire telle que :

$$N_{R \circ S} \;=\; N_R \quad \text{et} \quad R(A) \underset{S}{\Longrightarrow} (R \circ S)(A) \;\text{ pour tout } A \in N_R .$$

Pour obtenir une grammaire unique, nous définissons pour tout $A \in N_R$,

$$(R \circ S)(A) \;:=\; [R(A)] \;\cup\; \bigcup \{\, S(B)_{(s,t)} \mid s \xrightarrow[R(A)]{B} t \;\wedge\; B \in N_R \,\}$$

où $S(B)_{(s,t)}$ est le renommage de sommets $h(S(B))$ de $S(B)$ défini par

$$h(1) = s \;,\; h(2) = t \;,\; h(x) = (s, t, x) \;\text{ pour tout }\; x \in V_{S(B)} - \{1, 2\} .$$

En prenant la grammaire R de la figure 2.2, la composition $R \circ R$ est la grammaire suivante :

Rappelons que la *composition fonctionnelle* de deux applications $f : B \longrightarrow C$ et $g : A \longrightarrow B$ est l'application $f \circ g : A \longrightarrow C$ définie par $(f \circ g)(a) = f(g(a))$ pour tout $a \in A$.
L'interprétation de la composition de deux grammaires est la composition fonctionnelle de leurs interprétations.

Proposition 4.1.6 *Pour toute grammaire R et S avec $N_R = N_S$ et S sans circuit,*

$$[\![R \circ S]\!] \;=\; [\![R]\!] \circ [\![S]\!].$$

Cette égalité est aussi vérifiée pour S non sans circuit lorsque K est continu.

Démonstration. Soit $N_R = \{A_1, \ldots, A_p\}$. Soient $a_1, \ldots, a_p \in K$ et $A \in N_R$.
i) Montrons que

$$[\![(R \circ S)(A)\,[a_1, \ldots, a_p]]\!] \;=\; [\![R(A)\,[[\![S(A_1)[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)[a_1, \ldots, a_p]]\!]]]\!].$$

Nous avons

$$\begin{aligned}
& [\![(R \circ S)(A)\,[a_1, \ldots, a_p]]\!] \\
= \;& [\![L\big((R \circ S)(A)\,[a_1, \ldots, a_p], 1, 2\big)]\!] && \text{par définition de } [\![\ ]\!] \\
= \;& [\![L\big((R \circ S)(A), 1, 2\big)\,[a_1, \ldots, a_p]]\!] && \text{par le lemme 4.1.3} \\
= \;& [\![L\big(R(A), 1, 2\big)\ [L(S(A_1), 1, 2), \ldots, L(S(A_p), 1, 2)]\,[a_1, \ldots, a_p]]\!] && \text{par le lemme 4.1.4} \\
= \;& [\![L\big(R(A), 1, 2\big)\ [L(S(A_1), 1, 2)\,[a_1, \ldots, a_p], \ldots, L(S(A_p), 1, 2)\,[a_1, \ldots, a_p]\ ]\!] \\
= \;& [\![L\big(R(A), 1, 2\big)\ [L(S(A_1)\,[a_1, \ldots, a_p], 1, 2), \ldots, L(S(A_p)\,[a_1, \ldots, a_p], 1, 2)]\ ]\!] && \text{par 4.1.3} \\
= \;& [\![L\big(R(A), 1, 2\big)\ [[\![L(S(A_1)\,[a_1, \ldots, a_p], 1, 2)]\!], \ldots, [\![L(S(A_p)\,[a_1, \ldots, a_p], 1, 2)]\!]]\ ]\!] && \text{par 4.1.5} \\
= \;& [\![L\big(R(A), 1, 2\big)\ [[\![S(A_1)\,[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)\,[a_1, \ldots, a_p]]\!]]\ ]\!] && \text{par définition de } [\![\ ]\!] \\
= \;& [\![L\big(R(A)\ [[\![S(A_1)\,[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)\,[a_1, \ldots, a_p]]\!]], 1, 2\big)]\!] && \text{par le lemme 4.1.3} \\
= \;& [\![R(A)\,[[\![S(A_1)[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)[a_1, \ldots, a_p]]\!]]]\!].
\end{aligned}$$

ii) Nous avons

$$
\begin{aligned}
& [\![R \circ S]\!] (a_1, \ldots, a_p) \\
= \quad & ([\![(R \circ S)(A_1) [a_1, \ldots, a_p]]\!], \ldots, [\![(R \circ S)(A_p) [a_1, \ldots, a_p]]\!]) \quad \text{par définition de } [\![\ ]\!] \\
= \quad & ([\![R(A_1) [[\![S(A_1)[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)[a_1, \ldots, a_p]]\!]]]\!], \ \ldots, \\
& \qquad\qquad [\![R(A_p) [[\![S(A_1)[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)[a_1, \ldots, a_p]]\!]]]\!] \quad \text{par } p \text{ fois } (i) \\
= \quad & [\![R]\!] \left([\![S(A_1)[a_1, \ldots, a_p]]\!], \ldots, [\![S(A_p)[a_1, \ldots, a_p]]\!] \right) \quad \text{par définition de } [\![R]\!] \\
= \quad & [\![R]\!] \left([\![S]\!] (a_1, \ldots, a_p) \right) \quad \text{par définition de } [\![S]\!] \\
= \quad & \left([\![R]\!] \circ [\![S]\!] \right) (a_1, \ldots, a_p) \ .
\end{aligned}
$$

□

Lorsque K est continu, la substitution de langage est une opération continue.

Lemme 4.1.7 *Soit K un demi-anneau idempotent et continu.*
Pour tout $M \subseteq (K \cup N)^*$ et toutes suites croissantes $(a_{1,n})_{n \geq 0}, \ldots, (a_{p,n})_{n \geq 0}$,

$$[\![M [\textstyle\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \ = \ \bigvee_n [\![M [a_{1,n}, \ldots, a_{p,n}]]\!] .$$

Démonstration. i) Montrons que pour tout $u \in (K \cup N)^*$,

$$[\![u [\textstyle\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \ = \ \bigvee_n [\![u [a_{1,n}, \ldots, a_{p,n}]]\!] .$$

La démonstration est faite par récurrence sur $|u| \geq 0$.

$|u| = 0$: Alors $u = \varepsilon$ et les deux termes sont égaux à $[\![\varepsilon]\!] \ = \ 1$.

$|u| = 1$: Soit $u \in K$ et les deux termes sont égaux à $[\![u]\!] \ = \ u$.

Soit $u = A_i$ pour un $1 \leq i \leq p$, et les deux termes sont égaux à $[\![\bigvee_n a_{i,n}]\!]$.

$|u| > 1 : u = av$ pour $a \in K \cup N$. Alors

$$
\begin{aligned}
& [\![u[\textstyle\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \\
= \quad & [\![(av)[\textstyle\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \\
= \quad & [\![a[\textstyle\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}] \ . \ v[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \\
= \quad & [\![a[\textstyle\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \cdot [\![v[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \qquad \text{car } [\![\ ]\!] \text{ est un morphisme} \\
= \quad & \left(\textstyle\bigvee_n [\![a [a_{1,n}, \ldots, a_{p,n}]]\!] \right) \cdot \left(\bigvee_n [\![v [a_{1,n}, \ldots, a_{p,n}]]\!] \right) \quad \text{par hypothèse de récurrence} \\
= \quad & \textstyle\bigvee_n \left([\![a [a_{1,n}, \ldots, a_{p,n}]]\!] \cdot [\![v [a_{1,n}, \ldots, a_{p,n}]]\!] \right) \qquad \text{car } \cdot \text{ est continue} \\
= \quad & \textstyle\bigvee_n [\![a [a_{1,n}, \ldots, a_{p,n}] \ . \ v [a_{1,n}, \ldots, a_{p,n}]]\!] \qquad \text{car } [\![\ ]\!] \text{ est un morphisme} \\
= \quad & \textstyle\bigvee_n [\![u [a_{1,n}, \ldots, a_{p,n}]]\!] .
\end{aligned}
$$

ii) Vérifions que $\sum$ est continue.

Pour toutes suites croissantes $(a_{0,n})_{n\geq 0}, \ldots, (a_{m,n})_{n\geq 0}, \ldots$, nous avons

$$\begin{array}{lll}
\sum_{m\geq 0} \bigvee_n a_{m,n} & = & \bigvee_{m\geq 0} \sum_{i=0}^{m} (\bigvee_n a_{i,n}) \\
& = & \bigvee_{m\geq 0} \bigvee_n \sum_{i=0}^{m} a_{i,n} \qquad \text{car } + \text{ est continue} \\
& = & \bigvee_n \bigvee_m \sum_{i=0}^{m} a_{i,n} \\
& = & \bigvee_n \sum_{m\geq 0} a_{m,n}\,.
\end{array}$$

iii) Il s'en suit que

$$\begin{array}{lll}
[\![M[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] & = & \sum \{\, [\![u[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] \mid u \in M \,\} \\
& = & \sum \{\, \bigvee_n [\![u\,[a_{1,n}, \ldots, a_{p,n}]]\!] \mid u \in M \,\} \qquad \text{par } (i) \\
& = & \bigvee_n \sum \{\, [\![u\,[a_{1,n}, \ldots, a_{p,n}]]\!] \mid u \in M \,\} \qquad \text{par } (ii) \\
& = & \bigvee_n [\![M\,[a_{1,n}, \ldots, a_{p,n}]]\!]\,.
\end{array}$$

□

L'interprétation de grammaire est une application continue.

Lemme 4.1.8 *Pour un demi-anneau K idempotent et continu et une K-grammaire R, l'opération $[\![R]\!]$ est continue.*

Démonstration. Soient $(a_{1,n})_{n\geq 0}, \ldots, (a_{p,n})_{n\geq 0}$ des suites croissantes d'éléments de K.

i) Pour tout non-terminal $A \in N_R$, nous avons

$$\begin{array}{lll}
 & [\![R(A)\,[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] & \\
= & [\![L(R(A)\,[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}], 1, 2)]\!] & \text{par déf. d'une valeur de graphe} \\
= & [\![L(R(A), 1, 2)\,[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}]]\!] & \text{par le lemme 4.1.3} \\
= & \bigvee_n [\![L(R(A), 1, 2)\,[a_{1,n}, \ldots, a_{p,n}]]\!] & \text{par le lemme 4.1.7} \\
= & \bigvee_n [\![L(R(A)\,[a_{1,n}, \ldots, a_{p,n}], 1, 2)]\!] & \text{par le lemme 4.1.3} \\
= & \bigvee_n [\![R(A)\,[a_{1,n}, \ldots, a_{p,n}]]\!] & \text{par déf. d'une valeur de graphe.}
\end{array}$$

ii) Alors

$$
\begin{array}{lll}
 & [\![R]\!] \left(\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}\right) & \\
= & \left([\![R(A_1) \left[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}\right]]\!], \ldots, [\![R(A_p) \left[\bigvee_n a_{1,n}, \ldots, \bigvee_n a_{p,n}\right]]\!]\right) & \\
= & \left(\bigvee_n [\![R(A_1) \left[a_{1,n}, \ldots, a_{p,n}\right]]\!], \ldots, \bigvee_n [\![R(A_p) \left[a_{1,n}, \ldots, a_{p,n}\right]]\!]\right) & \text{par } (i) \\
= & \bigvee_n \left([\![R(A_1) \left[a_{1,n}, \ldots, a_{p,n}\right]]\!], \ldots, [\![R(A_p) \left[a_{1,n}, \ldots, a_{p,n}\right]]\!]\right) & \\
= & \bigvee_n [\![R]\!](a_{1,n}, \ldots, a_{p,n}) \,. &
\end{array}
$$

□

La valeur d'un graphe engendré par une grammaire peut être calculée en appliquant itérativement l'interprétation de la grammaire à partir de l'élément le plus petit.

Théorème 5 *Pour tout demi-anneau K idempotent et continu et pour toute grammaire R sur K,*

$$[\![R^\omega]\!] \;=\; \bigvee_{n\geq 0} [\![R]\!]^n(0) \;=\; \mu[\![R]\!].$$

La première égalité est aussi valable pour K complet et idempotent, lorsque R est sans circuit.

Démonstration. Grâce au lemme 4.1.8, la seconde égalité est juste une application du théorème de Knaster-Tarski. Montrons la première égalité.

Par définitions de la composition de grammaires et de R^ω, nous avons :

$$\bigcup_{n\geq 0} [R^n(A)] \quad\in\quad R^\omega(A) \;\text{ pour tout } A \in N_R \,.$$

Alors

$$
\begin{array}{rcll}
[\![R^\omega(A)]\!] & = & [\![\bigcup_{n\geq 0} [R^n(A)]]\!] & \\
 & = & \bigvee_{n\geq 0} [\![[R^n(A)]]\!] & \text{par le lemme 4.1.1 } (b) \\
 & = & \bigvee_{n\geq 0} [\![R^n(A) \left[0, \ldots, 0\right]]\!] & \text{par le lemme 4.1.1 } (c) \,. \\
[\![R^\omega]\!] & = & \left(\bigvee_{n\geq 0} [\![R^n(A_1) \left[0, \ldots, 0\right]]\!], \ldots, \bigvee_{n\geq 0} [\![R^n(A_p) \left[0, \ldots, 0\right]]\!]\right) & \\
 & = & \bigvee_{n\geq 0} \left([\![R^n(A_1) \left[0, \ldots, 0\right]]\!], \ldots, [\![R^n(A_p) \left[0, \ldots, 0\right]]\!]\right) & \\
 & = & \bigvee_{n\geq 0} [\![R^n]\!](0, \ldots, 0) & \text{par définition de } [\![R^n]\!] \\
 & = & \bigvee_{n\geq 0} [\![R]\!]^n(0, \ldots, 0) & \text{par la proposition 4.1.6} \,.
\end{array}
$$

□

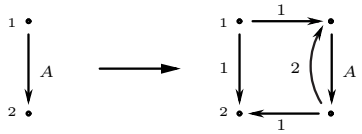
Dans le théorème 5, la continuité du demi-anneau est nécessaire pour des grammaires ayant un circuit. Par exemple, nous considérons le demi-anneau idempotent et complet $(\mathbb{N} \cup \{\omega, \overline{\omega}\}, Max, \times, 0, 1)$ dont les opérations Max et $\times$ sont respectivement les opérations maximum et multiplication habituelles sur les entiers, et que nous étendons comme suit :

$$\begin{array}{rcccll}
0 \times \omega & = & \omega \times 0 & = & 0 & \\
n \times \omega & = & \omega \times n & = & \omega & \text{pour tout entier non nul } n \\
 & & \omega \times \omega & = & \overline{\omega} & \\
a \times \overline{\omega} & = & \overline{\omega} \times a & = & \overline{\omega} & \text{pour tout élément } a \\
Max(a, \overline{\omega}) & = & Max(\overline{\omega}, a) & = & \overline{\omega} & \text{pour tout élément } a \\
Max(a, \omega) & = & Max(\omega, a) & = & \omega & \text{pour tout élément } a \neq \overline{\omega}\,.
\end{array}$$

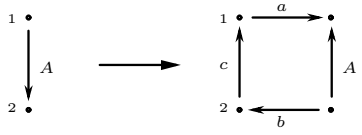
Ce demi-anneau n'est pas continu parce que

$$(\bigvee_{n\geq 0} n) \times (\bigvee_{n\geq 0} n) \;=\; \omega \times \omega \;=\; \overline{w} \qquad \text{mais} \qquad \bigvee_{n\geq 0} n \times n \;=\; \omega.$$

Pour ce demi-anneau K, nous considérons la K-grammaire réduite à la règle suivante :



Nous avons $[\![R]\!] = [\![R^{\omega}(A)]\!] = \omega$ mais $\bigvee_{n\geq 0} [\![R]\!]^n(0) = \overline{\omega}$ (la suite croissante est $0\,,\ 1\,,\ \omega\,,\ \overline{\omega}\,,\ \ldots$). Il est à noter que la première égalité du théorème 5 peut ne pas être vérifiée si l'on permet d'avoir un arc de but 1 ou de source 2 dans un membre droit de la grammaire. Par exemple, en prenant le demi-anneau $(2^{\{a,b,c\}^*}, \cup, ., \emptyset, \{\epsilon\})$ et la grammaire R restreinte à la règle suivante :



nous avons $\bigvee_{n\geq 0} [\![\ R\ ]\!]^n(\emptyset) \;=\; \emptyset \;\neq\; [\![\ R^{\omega}\ ]\!]$.

4.2 Algorithmes de calculs

Nous présentons deux algorithmes généraux pour calculer la valeur d'une grammaire R pour des demi-anneaux idempotents, continus et commutatifs. Sachant que $|R|$ désigne le nombre de règles, le premier algorithme compare les différences entre le $|R|$-*ième* approximant $[\![R]\!]^{|R|}(0)$ avec le suivant pour détecter des augmentations (strictement supérieures à 1) dans la valeur des graphes engendrés (cf. théorème 1). Le deuxième algorithme est une simple généralisation de la méthode de Hopkins-Kozen [HK 99] qui a été développée pour les grammaires algébriques (cf. corollaire 7). Bien que le premier algorithme soit plus efficace que le second, il travaille avec des demi-anneaux dont l'ordre est total et il admet un plus grand élément.

Étant donné un demi-anneau et un algorithme de calcul $[\![G, I, F]\!]$ pour tout graphe fini G (et les ensembles de sommets I et F) de complexité $C_{G,I,F}$, nous voulons utiliser cet algorithme pour calculer $[\![R]\!]^{\omega}$ pour toute grammaire R. La complexité est exprimée avec les paramètres suivants :

$|R| = |N_R|$ le nombre p de règles de R (son cardinal),

$l_R := \Sigma \{ |R(A)| \mid A \in N_R \}$ la longueur de description de R,

$C_R := \Sigma \{ C_{R(A)[0],1,2} \mid A \in N_R \}$ la complexité pour calculer $[\![R]\!](0)$.

Le théorème 5 nécessite des demi-anneaux idempotents et continus. Nos algorithmes exigent que ces demi-anneaux soient commutatifs : la multiplication $\cdot$ est commutative. Un *cci-demi-anneau* est une abréviation pour un demi-anneau idempotent, continu et commutatif.

Le premier algorithme a également besoin que le demi-anneau soit *total*, ce qui signifie que la relation $\leq$ est un ordre total. Nous allons décrire cet algorithme.

Pour toute grammaire R, le théorème 5 présente une méthode standard pour calculer $[\![R^{\omega}]\!]$: s'il existe n tel que $[\![R]\!]^n(0) = [\![R]\!]^{n+1}(0)$, alors $[\![R^{\omega}]\!] = [\![R]\!]^n(0)$ est vrai pour tout *demi-anneau borné* *i.e.* sans suite infinie croissante. Mais la suite $\left([\![R]\!]^n(0)\right)_n$ est strictement croissante en général.

Nous comparons les vecteurs $[\![R]\!]^p(0)$ et $[\![R]\!]^{p+1}(0)$ (avec $p = |R|$) et nous déterminons les rangs pour lesquels ils diffèrent :

$$E = \{ A_i \mid ([\![R]\!]^p(0))_i \neq ([\![R]\!]^{p+1}(0))_i \}.$$

Par un argument classique de pompage et pour chaque $A \in E$, nous pouvons trouver un incrément > 1 de la valeur de son graphe engendré.

En supposant que nous avons un plus grand élément $\top$ atteint par toute suite croissante stricte $(a^n)_{n \geq 0}$, nous obtenons $[\![R^{\omega}(A)]\!] = \top$.

Ceci est également vrai pour tout non-terminal ayant A dans son membre droit. Aussi, nous complétons E en l'ensemble $\overline{E}$ qui est le plus petit point fixe de l'équation suivante :

$$\overline{E} \;=\; E \,\cup\, \{\, A \in N_R \mid L_{R(A)} \cap \overline{E} \;\neq\; \emptyset \,\}\,.$$

On en déduit pour tout rang $1 \leq i \leq p$,

$$\big(\!\!\big[\, R^\omega \,]\!\!\big)_i \;=\; \begin{cases} \top & \text{si } A_i \in \overline{E}, \\ \big(\!\!\big[\, R \,]\!\!\big]^p(0)\big)_i & \text{sinon.} \end{cases}$$

La complexité en temps est la complexité du calcul de $\big[\!\!\big[\, R \,]\!\!\big]^p(0)$.

Nous écrivons $s \underset{G/H}{\overset{u}{\Longrightarrow}} t$ s'il existe un chemin $s_0 \xrightarrow[G]{a_1} s_1 \,\ldots\, s_{n-1} \xrightarrow[G]{a_n} s_n$ dans G de $s = s_0$ à $t = s_n$ étiqueté par $u = a_1 \ldots a_n$ dont au moins un arc n'est pas dans le graphe H : $\neg(s_i \xrightarrow[H]{a_i} s_{i+1})$ pour un $0 \leq i < n$.

Nous donnons une propriété fondamentale de la décomposition des chemins pour les graphes engendrés par une grammaire.

Lemme 4.2.1 *Pour toute grammaire R, tout non-terminal $A \in N_R$ et tout entier $n \geq 1$, si $1 \underset{R^{n+1}(A)/R^n(A)}{\overset{z}{\Longrightarrow}} 2$ alors il existe $uvw = z$ et $s \xrightarrow[R(A)]{B} t$ avec $B \in N_R$ tel que $1 \underset{R^{n+1}(A)}{\overset{u}{\Longrightarrow}} s$, $1 \underset{R^n(B)/R^{n-1}(B)}{\overset{v}{\Longrightarrow}} 2$, $t \underset{R^{n+1}(A)}{\overset{w}{\Longrightarrow}} 2$.*

Démonstration. Il y a un chemin

$$s_0 \xrightarrow[R^{n+1}(A)]{a_1} s_1 \,\ldots\, s_{m-1} \xrightarrow[R^{n+1}(A)]{a_m} s_m \;\text{ avec } z = a_1 \ldots a_m\,,\; s_0 = 1 \text{ et } s_m = 2.$$

Soit

$$I \;:=\; \{\, 0 \leq i < m \mid s_i \in V_{R(A)} \;\wedge\; \neg(s_i \xrightarrow[R(A)]{a_i} s_{i+1}) \,\}\,.$$

Pour tout $i \in I$, nous notons i' le successeur de i dans $V_{R(A)}$:

$$s_{i'} \in V_{R(A)} \;\wedge\; (i < i') \;\wedge\; (\forall\, i < k < i',\; s_k \notin V_{R(A)})\,.$$

Pour tout $i \in I$, il y a $s_i \xrightarrow[R(A)]{A_i} s_{i'}$ tel que $A_i \in N_R$ et $1 \underset{R(A_i)}{\overset{a_i \ldots a_{i'-1}}{\Longrightarrow}} 2$.

Comme $1 \underset{R^{n+1}(A)/R^n(A)}{\overset{z}{\Longrightarrow}} 2$, il y a un $k \in I$ tel que $1 \underset{R^n(A_k)/R^{n-1}(A_k)}{\overset{a_k \ldots a_{k'-1}}{\Longrightarrow}} 2$.

Il reste à prendre

$$s = s_k\;,\; t = s_{k'}\;,\; B = A_k\;,\; u = a_1 \ldots a_{k-1}\;,\; v = a_k \ldots a_{k'-1}\;,\; w = a_{k'} \ldots a_m\,.$$

□

Nous allons utiliser une construction similaire au lemme de pompage du chapitre 3 pour démontrer le théorème 1.

Tout demi-anneau total et idempotent satisfait la propriété de monotonicité ci-dessous.

Lemme 4.2.2 *Pour tout demi-anneau K idempotent et total, pour tout $M \subseteq K^*$ et $a \in K$,*

$$a < a + [\![M]\!] \quad \Longrightarrow \quad \exists\, u \in M,\ \ a < [\![u]\!] .$$

Démonstration. Par contraposée.

Supposons que $\neg(a < [\![u]\!])$ pour tout $u \in M$.

Comme $\leq$ est totale, $[\![u]\!] \leq a$ pour tout $u \in M$.

Comme $+$ est croissant et idempotent, nous avons,

$$[\![M]\!] \ = \ \sum \{\, [\![u]\!] \mid u \in M \,\} \ \leq \ \sum \{\, a \mid u \in M \,\} \ = \ a\,.$$

Ainsi, $[\![M]\!] + a \leq a + a = a$.

□

Comme pour le lemme de pompage, les différences entre $[\![R]\!]^{|R|}(0)$ et $[\![R]\!]^{|R|+1}(0)$ permettent de détecter des répétitions > 1 dans les graphes engendrés.

Théorème 1 *Pour tout cci-demi-anneau total K ayant un plus grand élément $\top$ tel que*

$$\textstyle\bigvee_n a^n = \top \ \text{ pour tout } \ a > 1 \quad \text{et} \quad a{\cdot}\top = \top \ \text{ pour tout } \ a \neq 0\,,$$

nous pouvons calculer $[\![R^\omega]\!]$ en $O(|R|\, C_R)$ pour toute K-grammaire R.

Démonstration. Pour tout $a_1, \ldots, a_p \in K$ et tout $A \in N_R$, nous notons

$$(a_1, \ldots, a_p)_A \ = \ a_i \ \text{ pour } \ A_i = A.$$

i) Supposons que $\big([\![R]\!]^p(0)\big)_A \ \neq \ \big([\![R]\!]^{p+1}(0)\big)_A$ pour un non-terminal $A \in N_R$.

Montrons que $\big([\![R^\omega]\!]\big)_A \ = \ \top$.

Comme pour le lemme de pompage, nous extrayons une répétition > 1 dans le graphe $R^\omega(A)$.

Comme la suite $[\![R]\!]^n(0)$ est croissante, $\big([\![R]\!]^p(0)\big)_A \ < \ \big([\![R]\!]^{p+1}(0)\big)_A$.

Pour chaque $n \geq 0$, nous avons :

$$\begin{array}{lcll} \big(\!\!\big[\, R \,\big]\!\!\big]^n(0)\big)_A & = & \big(\!\!\big[\, R^n \,\big]\!\!\big](0)\big)_A & \text{par le lemme 4.1.6} \\ & = & \big[\!\!\big[\, R^n(A)\,[0,\ldots,0] \,\big]\!\!\big] & \text{par définition de l'interprétation} \\ & = & \big[\!\!\big[\, [R^n(A)] \,\big]\!\!\big] & \text{par le lemme 4.1.1 (c).} \end{array}$$

Ainsi $\big[\!\!\big[\, [R^p(A)] \,\big]\!\!\big] < \big[\!\!\big[\, [R^{p+1}(A)] \,\big]\!\!\big]$. De plus,

$$\begin{array}{ll} & \big[\!\!\big[\, [R^{p+1}(A)] \,\big]\!\!\big] \\ = & \big[\!\!\big[\, L([R^{p+1}(A)],1,2) \,\big]\!\!\big] \quad \text{par définition de la valeur d'un graphe} \\ = & \big[\!\!\big[\, L(R^p(A),1,2)\,[L([R(A_1)],1,2),\ldots,L([R(A_p)],1,2)] \,\big]\!\!\big] \\ & \qquad \text{parce que } R^p(A) \underset{R}{\Longrightarrow} R^{p+1}(A) \text{ et par le lemme 4.1.4} \\ = & \big[\!\!\big[\, L([R^p(A)],1,2) \,\cup\, (L(R^p(A),1,2) - T_R^*)\,[L([R(A_1)],1,2),\ldots,L([R(A_p)],1,2)] \,\big]\!\!\big] \\ = & \big[\!\!\big[\, [R^p(A)] \,\big]\!\!\big] \,+\, \big[\!\!\big[\, (L(R^p(A),1,2) - T_R^*)\,[L([R(A_1)],1,2),\ldots,L([R(A_p)],1,2)] \,\big]\!\!\big] \end{array}$$

Par le lemme 4.2.2, il existe

$$z \in (L(R^p(A),1,2) - T_R^*)\,[L([R(A_1)],1,2),\ldots,L([R(A_p)],1,2)]$$

tel que $\big[\!\!\big[\, [R^p(A)] \,\big]\!\!\big] < \big[\!\!\big[\, z \,\big]\!\!\big]$. Aussi $1 \underset{R^{p+1}(A)/R^p(A)}{\overset{z}{\Longrightarrow}} 2$.

Comme pour le lemme de pompage et en appliquant p fois le lemme 4.2.1, nous obtenons une suite finie $(A_i, u_i, v_i, w_i)_{0 \le i \le p}$ telle que

$$(A_0, u_0, v_0, w_0) \;=\; (A, \varepsilon, z, \varepsilon)$$

et pour tout $1 \le i \le p$,

$$A_i \in N_R \quad \wedge \quad 1 \underset{R^{p+1-i}(A)/R^{p-i}(A)}{\overset{v_i}{\Longrightarrow}} 2 \quad \wedge \quad u_i v_i w_i = v_{i-1}.$$

Nous prenons la première répétition

$$i \;=\; min\{\; 0 \le k < m \mid \exists\, j > k,\; A_k = A_j \;\}.$$

Soit $j > i$ tel que $A_i = A_j$. Soient

$$u = u_1 \ldots u_i \;,\; v = u_{i+1} \ldots u_j \;,\; w = v_j \;,\; x = w_j \ldots w_{i+1} \;,\; y = w_i \ldots w_1 \,.$$

En itérant (ou en enlevant) cette répétition, nous obtenons

$$uv^n wx^n y \;\in\; L(R^\omega(A),1,2) \;\text{ pour tout } n \ge 0.$$

Par récriture parallèle $R^p(A) \underset{R}{\Longrightarrow} R^{p+1}(A)$, nous pouvons faire plusieurs récritures.

Il est donc possible d'avoir de nouveau $1 \underset{R^{p+1}(A)/R^p(A)}{\overset{uwy}{\Longrightarrow}} 2$.

Nous itérons cette décomposition qui aura une incidence sur u ou y jusqu'à ce que nous ayons

$$z \;=\; u_1v_1w_1x_1y_1 \,\ldots\, u_kv_kw_kx_ky_k \;\text{ avec }\; u_1w_1y_1 \,\ldots\, u_kw_ky_k \;\in L(R^p(A),1,2)$$

et

$$u_1v_1^{n_1}w_1x_1^{n_1}y_1 \,\ldots\, u_kv_k^{n_k}w_kx_k^{n_k}y_k \;\in\; L(R^\omega(A),1,2) \;\text{ pour tout } n_1,\ldots,n_k \geq 0.$$

Soient $a \;=\; [\![\, u_1w_1y_1 \,\ldots\, u_kw_ky_k \,]\!]$ et $b \;=\; [\![\, v_1x_1 \,\ldots\, v_kx_k \,]\!]$.

Alors $a{\cdot}b \;=\; [\![\, z \,]\!] \;>\; [\![\, [R^p(A)] \,]\!] \;\geq\; a$.

Comme $a < a{\cdot}b$, nous avons $a \neq 0$ et $b > 1$.

En particulier, la suite $(a{\cdot}b^n)_n$ est croissante.

Donc, par continuité et d'après l'hypothèse, on a

$$\bigvee_n a{\cdot}b^n \;=\; a{\cdot}\Big(\bigvee_n b^n\Big) \;=\; a{\cdot}\top \;=\; \top\,.$$

Pour tout $n \geq 0$, on obtient

$$a{\cdot}b^n \;=\; [\![\, u_1v_1^nw_1x_1^ny_1 \,\ldots\, u_kv_k^nw_kx_k^ny_k \,]\!] \;\leq\; [\![\, R^\omega(A) \,]\!],$$

par conséquent, $[\![\, R^\omega(A) \,]\!] \;=\; \top$.

ii) Soient $A, B \in N_R$ avec $[\![\, R^\omega(A) \,]\!] \;=\; \top$ et $A \in L_{R(B)}$.

Montrons que $[\![\, R^\omega(B) \,]\!] \;=\; \top$.

Il y a un arc $s \xrightarrow{A} t$ dans le graphe $R(B)$.

Comme R est réduit, on a

$$1 \underset{R(B)}{\overset{u}{\Longrightarrow}} s \;\text{ et }\; t \underset{R(B)}{\overset{v}{\Longrightarrow}} 2 \;\text{ avec }\; u, v \in (K - \{0\}) \cup N_R\,.$$

Nous prenons le morphisme (substitution) σ de $(T_R \cup N_R)^*$ à 2^{K^*} défini par

$\sigma(a) = \{a\}$ pour tout $a \in T_R$ et $\sigma(C) = L(R^\omega(C),1,2)$ pour tout $C \in N_R$.

Ainsi, $\sigma(uAv) \;\subseteq\; \sigma(B)$, d'où $[\![\, \sigma(uAv) \,]\!] \;\leq\; [\![\, \sigma(B) \,]\!]$.

Par conséquent,

$$[\![\, \sigma(u) \,]\!] \cdot [\![\, R^\omega(A) \,]\!] \cdot [\![\, \sigma(v) \,]\!] \;\leq\; [\![\, R^\omega(B) \,]\!] \;\; i.e. \;\; [\![\, \sigma(u) \,]\!] \cdot \top \cdot [\![\, \sigma(v) \,]\!] \;\leq\; [\![\, R^\omega(B) \,]\!].$$

Comme R est réduit,

$$[\![\, \sigma(u) \,]\!] \neq 0 \;\text{ et }\; [\![\, \sigma(v) \,]\!] \neq 0\,, \;\text{ ainsi }\; [\![\, R^\omega(B) \,]\!] \;=\; \top\,.$$

iii) Soit

$$E \;\;=\;\; \{\; A \in N_R \mid ([\![\, R \,]\!]^p(0))_A \neq ([\![\, R \,]\!]^{p+1}(0))_A \;\}.$$

Par co-accessibilité, nous complétons E en le plus petit point fixe $\overline{E}$ de l'équation suivante :

$$\overline{E} \;\;=\;\; E \;\cup\; \{\; A \in N_R \mid L_{R(A)} \cap \overline{E} \;\neq\; \emptyset \;\}.$$

Par (i) et (ii), nous avons

$$[\![\, R^\omega \,]\!] \;=\; \top \;\text{ pour chaque }\; A \in \overline{E}.$$

Nous limitons R aux non-terminaux qui ne sont pas dans E :

$$R_{/\overline{E}} \;\;:=\;\; \{\; ((1, A, 2)\,,\, H) \in R \mid A \notin \overline{E} \;\}.$$

Donc $R_{/\overline{E}}$ est une K-grammaire satisfaisant :

$$\big([\![\, R_{/\overline{E}} \,]\!]^n(0)\big)_A \;\;=\;\; \big([\![\, R \,]\!]^n(0)\big)_A \;\text{ pour chaque }\; A \in N_R - \overline{E} \;\text{ et }\; n \geq 0.$$

Par conséquent,

$$[\![\, R_{/\overline{E}} \,]\!]^p(0) \;\;=\;\; [\![\, R_{/\overline{E}} \,]\!]^{p+1}(0) \;\;=\;\; \bigvee_n [\![\, R_{/\overline{E}} \,]\!]^n(0).$$

Donc pour tout $A \in N_R - \overline{E}$,

$$\big([\![\, R^\omega \,]\!]\big)_A \;=\; \big(\bigvee_n [\![\, R \,]\!]^n(0)\big)_A \;=\; \big(\bigvee_n [\![\, R_{/\overline{E}} \,]\!]^n(0)\big)_A \;=\; \big([\![\, R_{/\overline{E}} \,]\!]^p(0)\big)_A \;=\; \big([\![\, R \,]\!]^p(0)\big)_A.$$

Ainsi, nous avons pour tout $A \in N_R$,

$$\big([\![\, R^\omega \,]\!]\big)_A \;\;=\;\; \begin{cases} \top & \text{si } A \in \overline{E}, \\ \big([\![\, R \,]\!]^p(0)\big)_A & \text{sinon.} \end{cases}$$

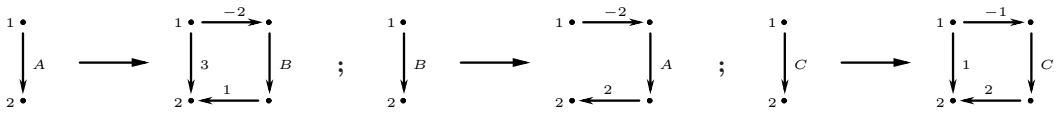
□

Appliquons le théorème 1 pour le demi-anneau $(\mathbb{R} \cup \{-\omega, \omega\}, Min, +, \omega, 0)$ de l'exemple 4.1 (c).

Cela nous permet de résoudre le problème du plus court chemin sur tout graphe engendré par une grammaire. Nous appliquons l'algorithme de Floyd (sans circuit de coût négatif) sur les graphes finis. Pour toutegrammaire R, nous avons C_R en $O(\ell_R^3)$, donc la complexité pour calculer $[\![R^\omega]\!]$ est en $O(|R|\, \ell_R^3)$.

Lorsque R est sans circuit, on peut appliquer l'algorithme de Bellman sur les graphes finis de sorte que C_R est en $O(\ell_R)$ et la complexité pour calculer $[\![R^\omega]\!]$ est en $O(|R|\, \ell_R)$, donc en temps quadratique.

Par exemple, en prenant la grammaire R suivante :



nous avons les approximants suivants :

n	0	1	2	3	4
$\big([\![R]\!]^n(0)\big)_A$	ω	3	3	2	2
$\big([\![R]\!]^n(0)\big)_B$	ω	ω	3	3	2
$\big([\![R]\!]^n(0)\big)_C$	ω	1	1	1	1

donnant par le théorème 1 la valeur : $[\![R^\omega]\!] = (-\omega, -\omega, 1)$ *i.e.*

$$[\![R^\omega(A)]\!] = [\![R^\omega(B)]\!] = -\omega \quad \text{et} \quad [\![R^\omega(C)]\!] = 1.$$

Notons que nous pouvons aussi appliquer le théorème 1 pour le demi-anneau $(\mathbb{R}_+ \cup \{\omega\}, Max, \times, 0, 1)$ de l'exemple 4.1 (d).

Notons que grâce aux deux transformations de la proposition 2, nous pouvons transformer en temps linéaire toute grammaire de graphes en une grammaire équivalente sans circuit. Ensuite, nous pouvons appliquer le théorème 1 avec l'algorithme de Bellman pour le problème du plus court chemin.

Corollaire 6 *Pour le demi-anneau* $(\mathbb{R} \cup \{-\omega, \omega\}, Min, +, \omega, 0)$*, pour toute grammaire (normalisée)* R *et pour tout non-terminal* A*, le problème du plus court chemin* $[\![R^\omega(A12), 1, 2]\!]$ *peut être résolu en* $O(\ell_R^2)$*, et en* $O(|R|\, \ell_R)$ *lorsque* $R^\omega(A12)$ *est sans circuit.*

En particulier, pour toute grammaire algébrique P, le problème du plus court chemin peut être résolu en $O(|P|\, \ell_P)$.

A partir de tout automate à pile R et en ajoutant des ε-transitions, il est facile de transformer R en un automate à pile R' reconnaissant le même langage (par états finaux et/ou pile vide) tel que chaque membre droit est un état suivi par au plus deux lettres de pile ; le nombre de règles $|R'|$ et donc sa longueur de description $\ell_{R'}$ sont en $O(\ell_R)$. Puis, nous appliquons à R' la transformation habituelle pour obtenir une grammaire algébrique équivalent P : $|P|$ et ℓ_P sont en $O(|R'|^3)$. Ainsi, pour tout automate à pile R, le problème du plus court chemin peut être résolu en $O(\ell_R^3)$.

Dans le paragraphe suivant, le corollaire 6 sera étendu en le corollaire 9 à partir de toute grammaire généralisée de graphes.

Notons également que les deux transformations de la proposition 2 et par la première égalité du théorème 5, nous pouvons calculer la valeur de toute grammaire de graphes pour tout demi-anneau non continu (mais complet et idempotent).

Une K-grammaire algébrique est une grammaire algébrique P telle que $T_P \subseteq K$. En ordonnant l'ensemble des non-terminaux : $N_P = \{A_1, \ldots, A_p\}$, la *valeur d'une grammaire algébrique* est

$$[\![L(P)]\!] := ([\![L(P, A_1)]\!], \ldots, [\![L(P, A_p)]\!]).$$

Pour le demi-anneau $(2^{\mathbf{N}^q}, \cup, +, \emptyset, (0, \ldots, 0))$ (exemple 4.1 (b)) des langages commutatifs sur q terminaux de la forme $\{(0, \ldots, 0, 1, 0, \ldots, 0)\}$, une première solution pour déterminer $[\![L(P)]\!]$ a été donnée par Parikh [Pa 66]. Cette méthode a été affinée dans [Pi 73] et généralisée dans [HK 99], et convient pour tout *cci*-demi-anneau. Cette dernière méthode est présentée ci-dessous.

La *dérivée* d'un langage $E \subseteq (N_P \cup T_P)^*$ par $A \in N_P$ est le langage suivant :

$$\frac{\partial E}{\partial A} := \{\, UV \mid UAV \in E \,\}$$

et comme $+$ est idempotent, nous pouvons nous limiter à enlever seulement la première occurrence de A, *i.e.* $|U|_A = 0$.

La *matrice Jacobienne* de P est

$$P' := \Big(\frac{\partial P(A_i)}{\partial A_j}\Big)_{1 \le i,j \le p}$$

où $P(A) = \{\, U \mid (A, U) \in P \,\}$ est l'image de $A \in N_P$ par P.

L'*interprétation* $[\![P']\!]$ de P' est l'application

$$\begin{array}{rccc} [\![P']\!] : & K^p & \longrightarrow & K^{p \times p} \\ & (a_1, \ldots, a_p) & \longmapsto & \big([\![\frac{\partial P(A_i)}{\partial A_j} [a_1, \ldots, a_p]]\!]\big)_{1 \le i,j \le p} \end{array}$$

où $E[a_1, \ldots, a_p]$ est le langage sur K qui est obtenu à partir du langage $E \subseteq (N_P \cup T_P)^*$ en remplaçant dans les mots de E chaque étiquette A_i par a_i pour tout $1 \leq i \leq p$.

La *transformation de Hopkins-Kozen* est le morphisme :

$$\begin{array}{rccc} \mathrm{HK}: & K^p & \longrightarrow & K^p \\ & (a_1, \ldots, a_p) & \longmapsto & \big(\llbracket P' \rrbracket(a_1, \ldots, a_p)\big)^* \times (a_1, \ldots, a_p) \end{array}$$

où $M \times \vec{v} := \big(M \cdot (\vec{v})^t\big)^t$ avec ^{t} pour vecteur de transposition , $\cdot$ pour la multiplication de la matrice et $*$ pour sa fermeture de Kleene.

En appliquant itérativement cette transformation $(\llbracket P(A_1)[0] \rrbracket, \ldots, \llbracket P(A_p)[0] \rrbracket)$, nous obtenons une suite croissante qui atteint sa borne supérieure $\llbracket L(P) \rrbracket$ après un nombre fini d'itérations [HK 99], et même après p itérations [EKL 07].

Théorème 2 [HK 99] [EKL 07] *Pour tout cci-demi-anneau K et pour toute K-grammaire algébrique P, nous avons*

$$\llbracket L(P) \rrbracket \;=\; HK^{|P|}\big(\llbracket P(A_1) \cap T^* \rrbracket, \ldots, \llbracket P(A_p) \cap T^* \rrbracket\big).$$

Le théorème 2 reste vrai même si les langages $P(A_1), \ldots, P(A_p)$ sont infinis [EKL 07]. En appliquant la transformation de la proposition 2 (a) à toute K-grammaire R de graphes, nous obtenons en temps linéaire une K-grammaire algébrique $\widehat{R}$ telle que $\llbracket R^\omega \rrbracket = \llbracket L(\widehat{R}) \rrbracket$, ce que l'on peut calculer d'après le théorème 2 pour tout cci-demi-anneau K. Toutefois, l'ensemble des non-terminaux $N_{\widehat{R}}$ dépend de ℓ_R et non pas de $|R|$. Il est donc plus efficace d'étendre le théorème 2 directement aux grammaires de graphes.

Soit R une grammaire de K-graphes avec l'ensemble $N_R = \{A_1, \ldots, A_p\}$ des non-terminaux.

Nous étendons la dérivée à tout $(K \cup N_R)$-graphe G avec $1, 2, \in V_G$.

Soit $A \in N_R$. Nous prenons un nouveau symbole A_0 et nous définissons le produit de synchronisation suivant :

$$\begin{array}{rrcl} & H & := & \big(G \cup \{\ s \xrightarrow{A_0} t \mid s \xrightarrow[G]{A} t\ \}\big) \times S_A \\ \text{avec} & S_A & := & \{\ 1 \xrightarrow{a} 1 \mid a \in K \cup N_P - \{A\}\ \} \cup \{1 \xrightarrow{A_0} 2\} \\ & & \cup & \{\ 2 \xrightarrow{a} 2 \mid a \in K \cup N_P\ \}\,. \end{array}$$

Prenant le renommage h de sommets de H défini par

$$h(1,1) = 1, h(2,2) = 2, \quad \text{et} \quad h(x) = x \quad \text{pour tout } x \in V_H - \{(1,1)\,, (2,2)\},$$

et en restreignant $h(H)$ au sous-graphe $\overline{H}$ des sommets accessibles à partir de 1 et co-accessibles à partir de 2, la *dérivée* du graphe G par A est le graphe :

$$\frac{\partial G}{\partial A} \quad := \quad \big(\overline{H} - V_{\overline{H}} \times A_0 \times V_{\overline{H}}\big) \cup \{\ s \xrightarrow{1} t \mid s \xrightarrow[\overline{H}]{A_0} t\ \}.$$

La *matrice Jacobienne* de R est

$$R' \quad := \quad \Big(\frac{\partial R(A_i)}{\partial A_j}\Big)_{1 \leq i,j \leq p}$$

et son interprétation est définie par

$$[\![\, R' \,]\!](a_1, \ldots, a_p) \quad := \quad \big([\![\, \tfrac{\partial R(A_i)}{\partial A_j}\, [a_1, \ldots, a_p] \,]\!]\big)_{1 \leq i,j \leq p} \ \text{ pour tout } a_1, \ldots, a_p \in K.$$

Ainsi, $[\![\, R' \,]\!] = [\![\, P' \,]\!]$ pour la grammaire algébrique infinie P définie par :

$$P(A) = L(R(A), 1, 2) \ \text{ pour tout } A \in N_R\,.$$

Remarquons que P est finie pour R sans circuit.

Pour toute grammaire R de K-graphes, nous associons la K-grammaire algébrique infinie suivante :

$$\breve{R} \quad := \quad \{\ (A, U) \mid U \in L(R(A), 1, 2)\ \}$$

de sorte que l'interprétation de sa matrice jacobienne soit la même que celle de R.

Lemme 4.2.3 *Pour toute grammaire R de graphes, nous avons $[\![\, R' \,]\!] = [\![\, \breve{R}' \,]\!]$.*

Démonstration. La dérivée $\frac{\partial G}{\partial A}$ pour tout $(K \cup N_R)$-graphe G par $A \in N_R$ a été définie de sorte que

$$\ll L(\tfrac{\partial G}{\partial A}, 1, 2) \gg \quad = \quad \ll \tfrac{\partial L(G,1,2)}{\partial A} \gg$$

avec $\ll U \gg$ le langage obtenu par l'élimination de 1 dans les mots de $U \subseteq K^*$.

Pour tout $a_1, \ldots, a_p \in K$, nous en déduisons que

$$[\![\, L(\tfrac{\partial G}{\partial A}, 1, 2)\, [a_1, \ldots, a_p] \,]\!] \quad = \quad [\![\, \tfrac{\partial L(G,1,2)}{\partial A}\, [a_1, \ldots, a_p] \,]\!]$$

et il s'en suit pour tout $1 \leq i, j \leq p$,

$$
\begin{array}{lll}
 & \big(\llbracket R' \rrbracket(a_1,\ldots,a_p)\big)_{i,j} & \\
= & \llbracket \frac{\partial R(A_i)}{\partial A_j}\,[a_1,\ldots,a_p] \rrbracket & \text{par définition de } \llbracket R' \rrbracket \\
= & \llbracket L(\frac{\partial R(A_i)}{\partial A_j}\,[a_1,\ldots,a_p],1,2) \rrbracket & \text{par définiton d'une valeur de graphe.} \\
= & \llbracket L(\frac{\partial R(A_i)}{\partial A_j},1,2)\,[a_1,\ldots,a_p] \rrbracket & \text{par le lemme 4.1.3} \\
= & \llbracket \frac{\partial \breve{R}(A_i)}{\partial A_j}\,[a_1,\ldots,a_p] \rrbracket & \text{par définition de } \breve{R} \\
= & \big(\llbracket \breve{R}' \rrbracket\,(a_1,\ldots,a_p)\big)_{i,j} & \text{par définition de } \llbracket \breve{R}' \rrbracket .
\end{array}
$$

Ainsi, nous avons $\llbracket R' \rrbracket = \llbracket \breve{R}' \rrbracket$.
□

Appliquons le théorème 2 avec le lemme 4.2.3.

Corollaire 7 *Pour tout cci-demi-anneau K et pour toute K-grammaire de graphes R, nous avons* $\llbracket R^\omega \rrbracket = HK^{|R|}\big(\llbracket R \rrbracket(0)\big)$.
En supposant que les opérations $+, \cdot, *$ dans K sont en $O(1)$, la complexité est en $O(|R|^4 + |R|^2 C_R)$.

Contrairement au théorème 1, ce corollaire peut être utilisé pour tout *cci*-demi-anneau.

Le calcul de la matrice Jacobienne R' est en $O\big(\sum_{A\in N_R} |R||R(A)|\big) = O(|R|\ell_R)$.

Nous calculons $\llbracket R' \rrbracket(a_1,\ldots,a_p)$ en $O\big(\sum_{A\in N_R} |R| C_{R(A)[a_1,\ldots,a_p],1,2}\big) = O(|R| C_R)$.

Ayant une matrice carrée $M \in K^{p\times p}$ et un vecteur $\vec{a} \in K^p$, la valeur de $\vec{b} = M^*.\vec{a}$ correspond à la résolution du système linéaire $\vec{b} = \vec{a} + M.\vec{b}$ de p équations avec p variables $\{b_1,\ldots,b_p\}$. Par la méthode de Gauss, nous utilisons $O(p^3)$ opérations $+$ et $\cdot$, et $O(p)$ opérations $*$.

Pour le demi-anneau de l'exemple 4.1 (c) et en utilisant l'algorithme de Floyd sur les graphes finis, nous obtenons une complexité en $O(|R|^2\,\ell_R^{\,3})$ donc en $O(\ell_R^{\,5})$ pour le calcul de $\llbracket R^\omega \rrbracket$ pour toute K-grammaire R. Il s'agit d'une complexité plus grande que celle obtenue en appliquant le théorème 1.

4.3 Calculs sur des graphes réguliers

On étend la notion de grammaire de graphes en permettant des hyperarcs non-terminaux, c'est-à-dire que les arcs étiquetés par un non-terminal peuvent joindre plus que deux sommets (ou un seul). Ces grammaires généralisées engendrent les

graphes dits réguliers. La famille des graphes réguliers contient les graphes des transitions des automates à pile. Précisément, les graphes réguliers de degré fini sont les restrictions régulières des graphes des transitions des automates à pile. En découpant les hyperarcs non-terminaux en arcs, nous fournissons une transformation polynomiale de toute grammaire généralisée en une grammaire de graphes définissant les mêmes langages des traces (Proposition 8). Nous appliquons ensuite les résultats du paragraphe précédent pour obtenir des algorithmes polynomiaux pour résoudre le problème du plus court chemin sur les graphes réguliers (Corollaire 9).

A toute lettre a, on associe un entier non nul $\varrho(a) \geq 1$ appelé l'*a*rité de a. Tout mot $as_1 \ldots s_{\varrho(a)}$ est un *hyperarc* étiquetée par a et reliant dans l'ordre les sommets $s_1, \ldots, s_{\varrho(a)}$. Un hyperarc ast dont l'étiquette a est d'arité $\varrho(a) = 2$ est un arc (s, a, t). Un *hypergraphe* G est un ensemble d'hyperarcs, et sa *longeur* est $\ell_G := \sum \{ |u| \mid u \in G \}$ pour G fini.
Une *grammaire généralisée* R de graphes est un ensemble fini de règles de la forme :

$$A\,1 \ldots \varrho(A) \longrightarrow H$$

où $A\,1 \ldots \varrho(A)$ est un hyperarc étiqueté par A reliant les sommets $1, \ldots, \varrho(A)$, et H est un hypergraphe fini.

Les étiquettes des membres gauches forment l'ensemble N_R des non-terminaux, et les autres étiquettes qui sont dans les membres droits forment l'ensemble T_R des terminaux.

Comme nous voulons seulement engendrer des graphes, nous supposons que tout terminal est d'arité 2.

Nous étendons naturellement à toute grammaire généralisée R de graphes les notions de récriture $\underset{R}{\longrightarrow}$ et de récriture parallèle $\underset{R}{\Longrightarrow}$.

Désormais, *une grammaire généralisée* est une grammaire déterministe généralisée de graphes.

Il s'agit d'une K-grammaire généralisée R si $T_R \subseteq K$.

Comme pour les grammaires de graphes, nous définissons le graphe $R^\omega(H)$ engendré par une grammaire généralisée R à partir d'un hypergraphe H.

Nous donnons ci-dessous une grammaire généralisée restreinte à une règle unique, et son graphe engendré.
Pour toute grammaire généralisée R et pour tout non-terminal $A \in N_R$,

son membre droit dans R est $R(A\,1 \ldots \varrho(A))$, également désigné par $R(A)$,

le graphe engendré par R à partir de de A est noté $R^\omega(A\,1 \ldots \varrho(A))$ ou $R^\omega(A)$,

$|R| = |N_R|$ est le nombre de règles de R (son cardinal),

$\ell_R := \sum \{ |R(A)| + \varrho(A) \mid A \in N_R \}$ est la *longeur* de R,

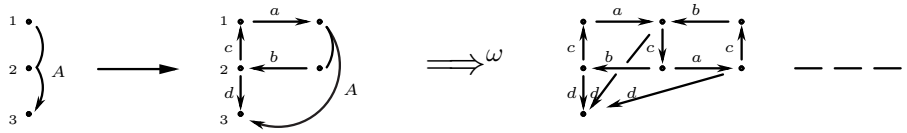


FIGURE 4.1 – Grammaire généralisée et graphe engendré.

$\varrho_R \;:=\; \sum \{\, \varrho(A) \mid A \in N_R \,\}$ est l'*arité* de R.

Un *graphe régulier* est un graphe engendré par une grammaire généralisée à partir d'un hyperarc non-terminal.

Nous transformons toute grammaire généralisée R en une grammaire (non généralisée) définissant les mêmes langages de traces. Pour cela, on découpe tout hyperarc non-terminal en tous les arcs possibles.

Nous supposons que 0 n'est pas un sommet de R et nous prenons un nouvel ensemble de symboles

$$\{\, A_{i,j} \mid A \in N_R \;\wedge\; 1 \leq i,j \leq \varrho(A) \,\}.$$

Nous définissons le *découpage* $\prec G \succ$ de tout $(T_R \cup N_R)$-hypergraphe G comme étant le graphe :

$$\begin{aligned}
\prec G \succ \quad &:= \quad \{\, s \xrightarrow{a} t \mid ast \in G \;\wedge\; a \in T_R \,\} \\
&\cup \quad \{\, s \xrightarrow{A_{i,j}} t \mid A \in N_R \;\wedge\; 1 \leq i,j \leq \varrho(A) \;\wedge \\
&\qquad\qquad \exists\, s_1,\ldots,s_{\varrho(A)},\; As_1\ldots s_{\varrho(A)} \in H \;\wedge\; s = s_i \;\wedge\; t = s_j \,\}
\end{aligned}$$

et pour $p,q \in V_G$ et $P \subseteq V_G$ avec $0 \notin V_G$, nous définissons

$$\begin{aligned}
G_{p,q,P} \quad &:= \quad \{\, s \xrightarrow[\prec G \succ]{a} t \mid t \neq p \;\wedge\; s \neq q \;\wedge\; s,t \notin P \,\} \quad \text{pour} \quad p \neq q \\
\text{et} \quad G_{p,p,P} \quad &:= \quad \{\, s \xrightarrow[\prec G \succ]{a} t \mid t \neq p \;\wedge\; s,t \notin P \,\} \cup \{\, s \xrightarrow{a} 0 \mid s \xrightarrow[\prec G \succ]{a} p \,\}.
\end{aligned}$$

Cela nous permet de définir le découpage de R comme étant la grammaire (non généralisée) suivante :

$$\prec R \succ \;:=\; \{\, (1, A_{i,j}, 2) \longrightarrow h_{i,j}(R(A)_{i,j,\{1,\ldots,\varrho(A)\}-\{i,j\}}) \mid A \in N_R \wedge 1 \leq i,j \leq \varrho(A) \,\}$$

où $h_{i,j}$ est le renommage des sommets de $R(A)_{i,j}$ défini par

$h_{i,j}(i) = 1\,,\; h_{i,j}(j) = 2\,,\; h_{i,j}(x) = x$ sinon, pour $i \neq j$

$h_{i,i}(i) = 1\,,\; h_{i,i}(0) = 2\,,\; h_{i,i}(x) = x$ sinon.

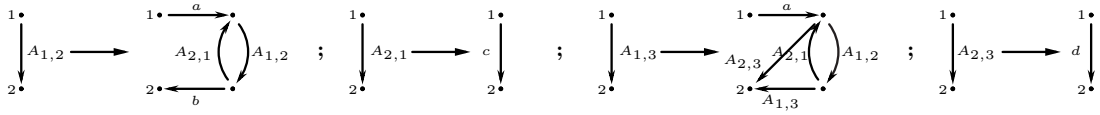


FIGURE 4.2 – Découpage sous forme réduite
de la grammaire généralisée de la figure 4.1 .

Nous mettons ensuite $\prec R\succ$ sous la forme réduite définie au paragraphe précédent.
Notons que

$$\begin{array}{rcccl} |\prec R\succ| & = & \sum_{A\in N_R} \varrho(A)^2 & \leq & \varrho_R^2 \\ \ell_{\prec R\succ} & \leq & \sum_{A\in N_R} \varrho(A)^2\,|R(A)|^2 & \leq & \varrho_R^2\,\ell_R^2\,. \end{array}$$

Ainsi, la transformation de découpage est polynomiale et elle définit les mêmes langages de traces.

Proposition 8 *Pour toute grammaire généralisée R, pour tout $(T_R \cup N_R)$-hypergraphe H et pour tout $s,t \in V_H$, nous avons* $L(R^\omega(H),s,t) \;=\; L(\prec R\succ^\omega(\prec H\succ),s,t)\,.$

Démonstration. Pour tout $A \in N_R$, $1 \leq i,j \leq \varrho(A)$ et $0 \leq n \leq \omega$, nous notons

$$\begin{array}{rcll} L_n(A,i,j) & := & L(R^n(A)_{i,j,\{1,\ldots,\varrho(A)\}-\{i,j\}},i,j) & \text{pour } i \neq j \\ L_n(A,i,i) & := & L(R^n(A)_{i,i,\{1,\ldots,\varrho(A)\}-\{i\}},i,0) & . \end{array}$$

i) Montrons que pour tout $A \in N_R$ et $1 \leq i,j \leq \varrho(A)$, nous avons

$$L_\omega(A,i,j) \;=\; L(\prec R\succ^\omega(A_{i,j}),1,2)\,.$$

Nous montrons par récurrence sur $n \geq 0$ que

$$L_n(A,i,j) \;=\; L(\prec R\succ^n(A_{i,j}),1,2)\,.$$

$n=0$: nous avons $L_0(A,i,j) \;=\; \{A_{i,j}\} \;=\; L(A_{i,j},1,2)$.

$n=1$: immédiat.

$n \implies n+1$:

$$\begin{array}{rl} & L_{n+1}(A,i,j) \\ = & L_1(A,i,j)\,[L_n(B,x,y)/B_{x,y}] \\ = & L(\prec R\succ(A_{i,j}),1,2)\,[L(\prec R\succ^n(B),1,2)/B_{x,y}] \qquad \text{par hypothèse de récurrence} \\ = & L(\prec R\succ^{n+1}(A_{i,j}),1,2). \end{array}$$

ii) Pour tout $(T_R \cup N_R)$-hypergraphe H et pour tout $s, t \in V_H$, nous avons

$$\begin{aligned} L(R^\omega(H), s, t) &= L(\prec H \succ, s, t)\,[L_\omega(A, i, j)/A_{i,j}] \\ &= L(\prec H \succ, s, t)\,[L(\prec R \succ^\omega(A_{i,j}), 1, 2)/A_{i,j}] \qquad \text{d'après } (i) \\ &= L(\prec R \succ^\omega(\prec H \succ), s, t). \end{aligned}$$

□

Illustrons la construction de la proposition 8. Nous considérons la grammaire généralisée R restreinte à la règle ci-dessous.

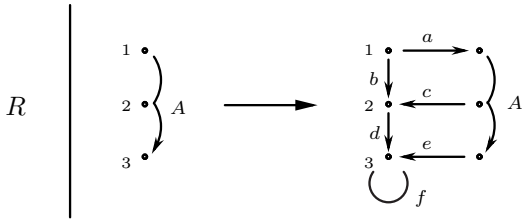


FIGURE 4.3 – Grammaire généralisée

Son graphe engendré $R^\omega(A)$ à partir de son non-terminal A est représenté à la figure 4.4.

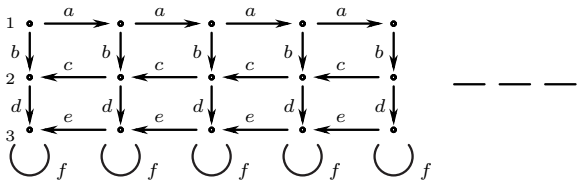


FIGURE 4.4 – Graphe engendré par la grammaire généralisée de la figuree 4.3.

On découpe la grammaire généralisée R en la grammaire sous forme réduite $\prec R \succ$ de la figure 4.5.

Prenant pour $\prec R \succ$ l'axiome réduit $\prec A123 \succ$ représenté à la figure 4.6,

nous obtenons le graphe engendré $\prec R \succ^\omega \prec A123 \succ$ de la figure 4.7.

Evidemment, les graphes $R^\omega(A123)$ et $\prec R \succ^\omega \prec A123 \succ$ ne sont pas isomorphes, mais par la proposition 8, ils reconnaissent les mêmes langages entre les entrées.

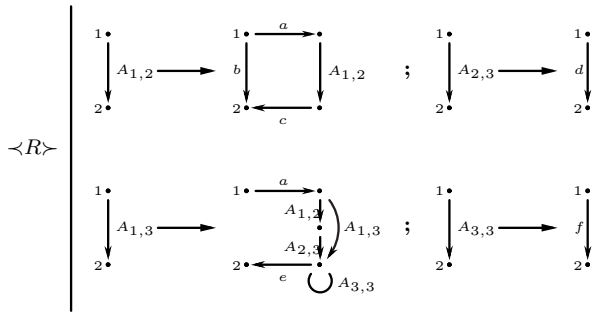


FIGURE 4.5 – Découpage sous forme réduite
de la grammaire généralisée de la figure 4.3 .

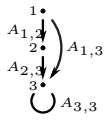


FIGURE 4.6 – Découpage de l'hyperarc $A123$
pour la grammaire de la figure 4.3.

Précisément,

$$
\begin{array}{lcllcl}
L(R^\omega(A123)\,,\,1\,,\,2) & = & L(\prec R\succ^\omega \prec A123\succ\,,\,1\,,\,2) & = & \{\ a^n bc^n \mid n \geq 0\ \} \\
L(R^\omega(A123)\,,\,2\,,\,3) & = & L(\prec R\succ^\omega \prec A123\succ\,,\,2\,,\,3) & = & df^* \\
L(R^\omega(A123)\,,\,3\,,\,3) & = & L(\prec R\succ^\omega \prec A123\succ\,,\,3\,,\,3) & = & f^* \\
L(R^\omega(A123)\,,\,1\,,\,3) & = & L(\prec R\succ^\omega \prec A123\succ\,,\,1\,,\,3) & = & \{\ a^{m+n} bc^m d(ef^*)^n \mid m,n \geq 0\ \} \\
L(R^\omega(A123)\,,\,i\,,\,j) & = & L(\prec R\succ^\omega \prec A123\succ\,,\,i\,,\,j) & = & \emptyset \quad \text{sinon.}
\end{array}
$$

La proposition 8 et la proposition 2 impliquent le fait bien connu que les langages reconnus (de et vers un sommet) par les graphes réguliers sont exactement les langages algébriques.

Mais la pertinence de la proposition 8 est que nous pouvons appliquer le théorème 1 et le corollaire 7 pour calculer les valeurs de graphes réguliers.

Etendons le corollaire 6.

Corollaire 9 *Pour le demi-anneau* $(\mathbb{R} \cup \{-\omega, \omega\}, Min, +, \omega, 0)$*, pour toute grammaire généralisée* R*, pour tout membre gauche* X *et pour tous sommets* $i, j \in V_X$ *, le problème du plus court chemin* $[\![\, R^\omega(X), i, j \,]\!]$ *peut être résolu en* $O(\varrho_R^{\,4}\, \ell_R^{\,4})$*, et en* $O(\varrho_R^{\,4}\, \ell_R^{\,2})$ *quand* $R^\omega(X)$ *est sans circuit.*

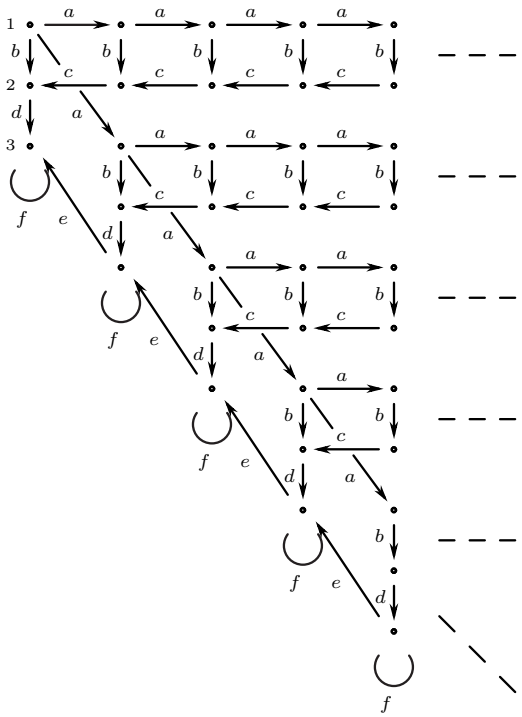


FIGURE 4.7 – Graphe engendré par la grammaire de la figure 4.5 à partir de l'axiome de la figure 4.6.

Bien sûr, le corollaire 9 est juste un problème spécifique de cheminement qui peut être résolu avec une complexité polynomiale en utilisant la proposition 8 avec le théorème 1 ou le corollaire 7.

5

Régularité et algébricité des systèmes de récriture

5.1 Introduction

Un problème central dans l'analyse de l'accessibilité d'un système R de récriture de mots est de déterminer l'ensemble $\xrightarrow[R]{*}(L)$ de tous les mots qui peuvent dériver selon R à partir de mots d'un langage L donné. Bien que pour L fini, cet ensemble $\xrightarrow[R]{*}(L)$ ne soit pas en général récursif, divers travaux de recherche ont dégagé des familles générales de systèmes de récriture de mots dont la relation de dérivation $\xrightarrow[R]{*}$ a de bonnes propriétés de fermeture : elle préserve des classes générales de langages. En particulier, on dit qu'un système R préserve la régularité (resp. l'algébricité) si, pour tout langage régulier (resp. algébrique) L, le langage $\xrightarrow[R]{*}(L)$ est aussi régulier (resp. algébrique).

On peut trouver dans la littérature de nombreuses classes de systèmes de récriture préservant la régularité ou l'algébricité. Par exemple, la dérivation préfixe de tout système fini de récriture préserve la régularité [Bu 64, Ca 90]. De plus, les systèmes algébriques (tout membre gauche est de longueur au plus 1) préservent l'algébricité et leurs dérivations inverses préservent la régularité (voir par exemple [BO 93]). Dans [HW 04], Hofbauer et Waldmann ont montré que la dérivation de tout système effaçant fini préserve la régularité sachant que Hibbard [Hi 74] avait établi que la dérivation inverse préserve l'algébricité. Au lieu d'utiliser des règles de découpage et d'élimination, ils ont fourni une décomposition astucieuse de la relation de dérivation en une substitution finie suivie par la dérivation d'un système algébrique inverse et une restriction à l'alphabet original. De là, ils ont pu déduire de nombreux résultats

déjà connus de préservation de la régularité et/ou de l'algébricité. La contribution principale de ce chapitre est de simplifier l'idée de la décomposition de dérivation de Hofbauer et Waldmann dans [HW 04] et de l'étendre à des familles plus générales de systèmes de récriture de mots.

Notre construction est basée sur une observation simple que l'on va décrire. Etant donné un mot $u = a_1 \dots a_n$, nous écrivons $\overleftarrow{u} = \overleftarrow{a_n} \dots \overleftarrow{a_1}$ et $\overrightarrow{u} = \overrightarrow{a_n} \dots \overrightarrow{a_1}$ où $\overleftarrow{a}$ et $\overrightarrow{a}$ sont de nouvelles lettres pour chaque lettre a d'origine. Pour tout système de récriture R et pour toute règle $u \to v$ de R, nous considérons les mots de la forme $\overleftarrow{u_1} v \overrightarrow{u_2}$ avec $u_1 u_2 = u$. Le sens implicite est que pour appliquer cette règle de récriture, le côté droit v peut être inséré à une certaine position i dans un mot à condition que u_1 puisse être effacé à la gauche de la position i et u_2 à la droite. En d'autres termes, l'application de la règle $u \to v$ à un mot peut être simulée en insérant d'abord un des facteurs $\overleftarrow{u_1} v \overrightarrow{u_2}$ à une position appropriée, puis en effaçant les facteurs de la forme $u\overleftarrow{u}$ ou $\overrightarrow{u}u$. Cette procédure en deux phases peut être effectuée par une substitution suivie par l'application de règles algébriques inverses de la forme $\overrightarrow{a}a \to \varepsilon$ et $a\overleftarrow{a} \to \varepsilon$ (constituant ce que nous appelons le système de récriture de Dyck, voir [KKO 06]).

En vertu de certains critères syntaxiques, cette étape de simulation peut être utilisée pour éliminer entièrement les règles de récriture du système initial. Plus précisément, nous sommes capable de décomposer la dérivation d'un système R en une substitution (régulière) dont le rôle est d'insérer des facteurs (tels que décrits ci-dessus) correspondants à un sous-ensemble R' de R, suivie par la dérivation d'un système $S \cup D$, où S est tout simplement $R - R'$ et D désigne le système de Dyck. On dit que R est décomposable en S. En conséquence, la dérivation de R (resp. R^{-1}) préserve la régularité (resp. l'algébricité) si c'est le cas pour la dérivation de $S \cup D$ (resp. son inverse). Cela reste vrai même pour les systèmes infinis lorsque le système R' est reconnaissable.

Ce résultat peut être utilisé pour caractériser quelques familles de systèmes dont la dérivation préserve la régularité ou l'algébricité. Premièrement, nous observons que dans le cas des systèmes effaçants, la décomposition produit un S vide (i.e toutes les règles peuvent être simulées et éliminées de R). Comme le système de Dyck est algébrique inverse, cela étend en effet le résultat de [HW 04] aux systèmes reconnaissables. En outre et contrairement à [HW 04], notre décomposition utilise uniquement un système algébrique élémentaire inverse, à savoir D. Notons cependant que de nombreux autres systèmes peuvent être directement décomposés en le système vide comme par exemple les systèmes de récriture préfixe (qui codent la relation de transition des systèmes à pile), leur variante bifixe, et les systèmes à gauche ou à droite. Dans le cas fini, la plupart de ces systèmes peuvent également être simulés par des systèmes effaçants, comme cela a été montré dans [HW 04].

Comme exemple, les systèmes multi-piles tels que ceux définis dans [BMT 05] peuvent être considérés comme des systèmes gauche-droite, et il en résulte que leur

relation de transition préserve l'algébricité et que leur inverse préserve la régularité.

Notre principale application concerne les systèmes marqués, qui généralisent les systèmes de récriture préfixe et suffixe. Etant donné un ensemble de symboles spéciaux appelés marques, que nous séparons en marques préfixes et marques suffixes, nous considérons des règles de la forme $\#u \longrightarrow \#'v$, où $\#, \#'$ sont des marques préfixes et u ne contient aucun marque. Nous permettons également des règles suffixes, bifixes, et démarquées qui sont définies de façon similaire. Puisque v peut contenir des marques, cela étend strictement les notions antérieures des systèmes marqués définis dans [Al 08]. Si l'ensemble des règles marquées est reconnaissable et si l'ensemble des règles non marquées est algébrique, nous montrons que notre décomposition s'applique, et qu'il en résulte que la dérivation d'un tel système (resp. son inverse) préserve l'algébricité (resp. la régularité). Ce résultat est encore vrai lorsque nous ne séparons pas les marques en préfixes et suffixes, mais en imposant que les marques dans le membre gauche d'une règle soient celles de son membre droit. Ceci étend les propriétés connues de préservation des langages réguliers et algébriques pour des systèmes marqués plus simples [Al 08].

Le reste de ce chapitre est organisé comme suit. Après quelques notations et définitions élémentaires, le paragraphe 5.2 présente notre théorème de décomposition de la dérivation, et son lien avec la classe des systèmes effaçants. Le paragraphe 5.3 détaille plusieurs classes de systèmes de récriture dont les résultats de préservation connus peuvent être récupérés à l'aide de notre technique (les systèmes préfixes, suffixes et bifixes du paragraphe 5.3.1) ou pour lesquels de nouveaux résultats de préservation peuvent être obtenus : les systèmes gauche-droite au paragraphe 5.3.2, et les systèmes marqués au paragraphe 5.3.3.

5.2 Décomposition de la dérivation

Dans ce paragraphe, nous nous focalisons sur la préservation de la régularité et de l'algébricité pour la dérivation des systèmes de récriture de mots. Après avoir analysé les résultats connus (paragraphe 5.2.2), nous généralisons la méthode de décomposition de la dérivation des systèmes effaçants [HW 04] à tous les systèmes de récriture de mots (paragraphe 5.2.3), ce qui nous permet d'obtenir de nouvelles sous-familles de systèmes préservant la régularité ou l'algébricité. Nous commençons par rappeler quelques définitions de base et nous introduisons quelques notations.

5.2.1 Notations

Pour simplifier, un singleton $\{x\}$ est souvent identifié avec x. Pour tout ensemble E, nous écrivons $|E|$ son cardinal, et 2^E l'ensemble de ses parties. Une *relation*

binaire R d'un ensemble E dans un ensemble F est une partie de $E\times F$. Toute couple $(x,y) \in R$ est également désigné par $x\,R\,y$. L'*inverse* de R est la relation $R^{-1} = \{\,(y,x) \mid x\,R\,y\,\}$. Le domaine de R est l'ensemble $\mathrm{Dom}(R) = \{\,x \mid \exists\, y\,(x\,R\,y)\,\}$ de ses premières composantes, et l'image de R est l'ensemble $\mathrm{Im}(R) = \{\,y \mid \exists\, x\,(x\,R\,y)\,\} = \mathrm{Dom}(R^{-1})$ de ses deuxièmes composantes. L'image par R d'un sous-ensemble $P \subseteq E$ est $R(P) = \{\,y \mid \exists\, x \in P\ (x\ R\ y)\,\}$; en particulier, $\mathrm{Im}(R) = R(E) = R(\mathrm{Dom}(R))$. L'identité sur E est la relation $\mathrm{Id}_E = \{\,(x,x) \mid x \in E\,\}$. La *composition* de $R \subseteq E\times F$ avec $S \subseteq F\times G$ est $R\circ S = \{\,(x,z) \mid \exists\, y\ (x\ R\ y\ \wedge\ y\ S\ z)\,\}$.

Soit N un alphabet. Nous notons $\mathrm{Alph}(u) = \{\,u(i) \mid 1 \le i \le |u|\,\}$ l'ensemble des lettres d'un mot $u \in N^*$. Ceci est étendu par union à tout langage $P \subseteq N^*$: $\mathrm{Alph}(P) = \{\,a \mid \exists\, u \in P,\, a \in \mathrm{Alph}(u)\,\}$. La *concaténation* de relations binaires R et S sur N^* est $R.S = \{\,(ux,vy) \mid u\,R\,v \wedge x\,S\,y\,\}$, et la concaténation à gauche et à droite de R par un langage $P \subseteq N^*$ est $R.P = R.\mathrm{Id}_P = \{\,(uw,vw) \mid u\,R\,v \wedge w \in P\,\}$ et $P.R = \mathrm{Id}_P.R$.

Un *automate* A sur E est un sous-ensemble $A \subseteq Q\times E\times Q \cup \{\iota,o\}\times Q$ où Q est l'ensemble de ses *états* et ι, o sont des symboles : on dit que p est une *entrée* pour $(\iota,p) \in A$ et que p est une *sortie* pour $(o,p) \in A$. Toute triplet $(p,a,q) \in A$ est une *transition* étiquetée par a de l'état p à l'état q. Un automate A reconnaît l'ensemble $L(A)$ des étiquettes de ses chemins acceptants : $L(A) = \{\,a_1\ldots a_n \mid n \ge 0\ \wedge\ \exists\, p_0,\ldots,p_n\ (\iota,p_0),(p_0,a_1,p_1),\ldots,(p_{n-1},a_n,p_n),(o,p_n) \in A\,\}$. Un *langage régulier* sur N est le langage reconnu par un automate fini étiqueté dans N (ou N^*). Une *substitution* h sur N est une relation binaire sur N^* définie par l'image $h(a)$ de chaque lettre $a \in N$, et qui est étendue par morphisme à tous les mots : $h(a_1\ldots a_n) = h(a_1)\ldots h(a_n)$ pour tout entier $n \ge 0$ et pour toutes lettres $a_1,\ldots,a_n \in N$. La substitution h est dite finie (resp. régulière) si $h(a)$ est un langage fini (resp. régulier) pour toute lettre $a \in N$. Une *relation reconnaissable* R sur N^* est une union finie de produits binaires de langages réguliers : $R = U_1\times V_1 \cup \ldots \cup U_p\times V_p$ pour $p \ge 0$ et pour des langages réguliers $U_1,V_1,\ldots,U_p,V_p$. Un *transducteur* A sur N est un automate étiqueté sur $N^*\times N^*$ dont le langage est interprété comme une relation binaire, appelée *relation rationnelle*.

Rappelons qu'un *système de récriture de mots* (ou tout simplement un *système*) R sur un alphabet N est une relation binaire sur N^* et est considérée comme un ensemble de règles (u,v) ; nous ne supposons pas que R soit fini. Soit $\mathrm{Alph}(R)$ l'ensemble des lettres de R. La *relation de récriture* selon R est la relation binaire $\underset{R}{\longrightarrow} = N^*.R.N^*$, *i.e.* $xuy \underset{R}{\longrightarrow} xvy$ pour tout $u\,R\,v$ et $x,y \in N^*$; nous écrirons aussi parfois $xuy \underset{R,|x|}{\longrightarrow} xvy$ pour désigner la position $|x|$ où la règle est appliquée. Notons que $(\underset{R}{\longrightarrow})^{-1} = \underset{R^{-1}}{\longrightarrow}$. La relation de *dérivation* $\overset{*}{\underset{R}{\longrightarrow}}$ de R est la fermeture réflexive et transitive (par composition) de $\underset{R}{\longrightarrow}$, *i.e.* $u \overset{*}{\underset{R}{\longrightarrow}} v$ s'il existe un entier $n \ge 0$ et des mots $u_0,\ldots,u_n \in N^*$ tels que $u = u_0 \underset{R}{\longrightarrow} u_1 \ldots \underset{R}{\longrightarrow} u_n = v$. Notons

que $(\xrightarrow[R]{*})^{-1} = \xrightarrow[R^{-1}]{*}$. Rappelons qu'une grammaire algébrique sur N est une relation finie $R \subseteq M \times (M \cup N)^*$ pour un alphabet M disjoint de N ; elle engendre à partir de $u \in (M \cup N)^*$ le *langage algébrique* $L(R,u) = \{\, v \in N^* \mid u \xrightarrow[R]{*} v \,\}$.

5.2.2 Propriétés de préservation

Un premier résultat bien connu est que toute relation rationnelle R (et son inverse) préserve la régularité : l'image $R(L)$ d'un langage régulier L reste régulier. Il est également bien connu en théorie des langages formels que la famille des langages algébrique est fermée par relation rationnelle.

Lemme 5.2.1 *Toute relation rationnelle préserve la régularité et l'algébricité.*

Démonstration. Précisément soit A un transducteur sur N.

Aussi A est un automate étiqueté dans $N^* \times N^*$; il reconnaît la relation binaire $[\![L(A)]\!]$ sur N^* dont $[\![\ ]\!]$ est le morphisme de $(N^* \times N^*)^*$ dans $N^* \times N^*$ de la concaténation composante à composante :

$$[\![(u_1, v_1) \ldots (u_n, v_n)]\!] = (u_1 \ldots u_n, v_1 \ldots v_n)$$

étendue par union à tout langage $L \subseteq (N^* \times N^*)^*$: $[\![L]\!] = \{\, [\![w]\!] \mid w \in L \,\}$.

A toute étiquette $(u,v) \in E_A$, nous associons un nouveau symbole $[u,v]$ et nous notons $E = \{\, [u,v] \mid (u,v) \in E_A \,\}$ l'ensemble de ces nouveaux symboles.

Nous transformons A en l'automate sur E :

$$[A] = \{\, p \xrightarrow{[u,v]} q \mid p \xrightarrow[A]{(u,v)} q \,\}$$

L'image par $[\![L(A)]\!]$ de n'importe quel langage $P \subseteq N^*$ est

$$[\![L(A)]\!](P) = \pi_2\big(\pi_1^{-1}(P) \cap L([A])\big)$$

où π_1 et π_2 sont respectivement la première et la deuxième projections étendues par morphisme de E^* dans N^* :

$$\pi_1([u,v]) = u \quad \text{et} \quad \pi_2([u,v]) = v \quad \text{pour tout } [u,v] \in E.$$

Pour A fini, $[A]$ est fini. De plus, la régularité des langages et l'algébricité des langages sont préservées par morphisme direct, par morphisme inverse et par intersection avec un langage régulier.

Ainsi, toute relation rationnelle $[\![L(A)]\!]$ préserve la régularité et l'algébricité.
□

Puisque les relations reconnaissables et les substitutions régulières sont des cas particuliers de relations rationnelles, il s'en suit que la famille des langages réguliers et celle des langages algébriques sont également fermées par images directe et inverse pour toute relation reconnaissable et pour toute substitution régulière.

Dans ce chapitre, nous nous intéressons à la caractérisation de classes de systèmes de récriture dont la relation de dérivation préserve la régularité ou l'algébricité. On va utiliser une méthode de décomposition générale qui sera détaillée au paragraphe suivant. Une façon simple de simuler l'application d'une règle de récriture (uv, w) à un mot x consiste à insérer à la position appropriée dans x, un facteur $\overleftarrow{u}\, w \overrightarrow{v}$ signifiant qu'à cette position, w peut apparaître après l'application de la règle si u peut être supprimé sur la gauche et v sur la droite. Après avoir inséré ce mot, les suppressions appropriées sont effectuées en utilisant un seul système de récriture appelé le système de Dyck. Par conséquent, les propriétés de préservation des langages pour ce système de récriture joue un rôle central dans notre étude. Formellement, soit $R \subseteq N^* \times N^*$ un système de récriture, et considérons un nouvel alphabet $\overleftrightarrow{N} = \overrightarrow{N} \cup \overleftarrow{N}$ composé de deux copies disjointes de N, avec $\overrightarrow{N} = \{\overrightarrow{a} \mid a \in N\}$ et $\overleftarrow{N} = \{\overleftarrow{a} \mid a \in N\}$. Cette notation est étendue à tout mot sur N comme suit : $\overrightarrow{a_1 \dots a_n} = \overrightarrow{a_n} \dots \overrightarrow{a_1}$ et $\overleftarrow{a_1 \dots a_n} = \overleftarrow{a_n} \dots \overleftarrow{a_1}$ pour tout $n \geq 0$ et $a_1, \dots, a_n \in N$. Le *système de Dyck* $D = N{\downarrow} \cup {\downarrow}N$ défini sur $\overline{N} = N \cup \overleftrightarrow{N}$ est l'union du *système droit de Dyck* $N{\downarrow} = \{(\overrightarrow{a}\, a, \varepsilon) \mid a \in N\}$ et du *système gauche de Dyck* ${\downarrow}N = \{\ (a \overleftarrow{a}\,, \varepsilon) \mid a \in N\}$.

Pour toute règle (uv, w) dans R, le mot xwy obtenu par récriture de $xuvy$ peut également être dérivé du mot $xu\overleftarrow{u}\, w \overrightarrow{v}\, vy$ en utilisant D.

Notons que lorsque $u = \varepsilon$ (resp. $v = \varepsilon$), il suffit d'utiliser $N{\downarrow}$ (resp. ${\downarrow}N$). C'est un résultat classique et largement utilisé que la relation de dérivation de $N{\downarrow}$ préserve la régularité [Be 69], mais pas l'algébricité. Un exemple [JKLP 87] est de prendre les langages algébriques L et M qui sont respectivement les solutions des équations $L = \overrightarrow{a}\, La \cup M$ et $M = b \cup aMM\overrightarrow{a}$. Alors $\xrightarrow[N{\downarrow}]{*}(L)$ n'est pas algébrique :

$$\xrightarrow[N{\downarrow}]{*}(L) \cap b^* \ = \ \{b^{2^n} \mid n \geq 0\}.$$

En outre, la dérivation de $N{\downarrow}^{-1}$ préserve l'algébricité, mais pas la régularité :

$$\xrightarrow[N{\downarrow}^{-1}]{*}(\varepsilon) \cap \overrightarrow{a}^* a^* \ = \ \{\overrightarrow{a}^n a^n \mid n \geq 0\}$$

qui n'est pas régulier (mais algébrique). Nous appelons donc le système $N{\downarrow}$ rég/alg-préservant, comme défini ci-dessous.

Définition. Un système R est *reg/alg-préservant* si sa relation de dérivation préserve la régularité et sa dérivation inverse préserve l'algébricité.
Un système R est *alg/reg-préservant* si R^{-1} est reg/alg-préservant.

On peut étendre la reg/alg-préservation du système (droit) de Dyck pour des classes plus générales de systèmes de récriture de mots. On dit qu'une relation binaire R sur N^* est un *système algébrique* si $R \subseteq (N \cup \{\varepsilon\}) \times N^*$ avec $R(a)$ un langage algébrique pour tout $a \in N \cup \{\varepsilon\}$.
Le système D^{-1} est un système algébrique n'ayant qu'un nombre fini de règles.

Proposition 10 ([BJW 82]) *Les systèmes algébriques sont alg/reg-préservants.*

Démonstration. i) Tout d'abord, nous nous restreignons à

$$R \subseteq N \times N^* \quad \text{avec} \quad R(a) \in \mathrm{Alg}(N^*) \ \text{ pour tout } a \in N.$$

Soit $L \in \mathrm{Alg}(N^*)$. Montrons que $\xrightarrow[R]{*}(L)$ reste algébrique.

Nous prenons un nouveau symbole $i \notin N$ et nous complétons R en la relation binaire

$$\overline{R} = R \cup \{(i, L)\} \ \text{ sur } \overline{N}^* \ \text{ avec } \ \overline{N} = N \cup \{i\}.$$

Pour tout $a \in \overline{N}$, il existe une grammaire algébrique $R_a \subset N_a \times (N \cup N_a)^*$ sur un nouvel alphabet N_a de non-terminaux, engendrant $\overline{R}(a)$ à partir d'un axiome $I_a \in N_a$:

$$L(R_a, I_a) = \overline{R}(a).$$

A un renommage près, on peut supposer que les alphabets non-terminaux sont disjoints : $N_a \cap N_b = \emptyset$ pour $a \neq b \in \overline{N}$, et que pour tout $a \in \overline{N}$, l'axiome I_a n'apparaît pas dans les membres droits de R_a .

Soient $\widehat{N} = \bigcup\{ N_a \mid a \in \overline{N} \}$ et $h : (N \cup \widehat{N})^* \longrightarrow \widehat{N}^*$ le morphisme défini par

$$h(a) = I_a \ \text{ pour tout } \ a \in N, \ \text{ et } \ h(x) = x \ \text{ pour tout } \ x \in \widehat{N}.$$

Nous définissons la grammaire algébrique :

$$S = \{ (A, h(U)) \mid \exists\, a \in \overline{N},\ (A, U) \in R_a \} \cup \{ (I_a, a) \mid a \in N \}.$$

Ainsi, S engendre à partir de I_i le langage $\xrightarrow[R]{*}(L)$ qui est donc algébrique.

ii) Soit R une relation algébrique quelconque.

Déduisons de (i) que $\xrightarrow[R]{*}$ préserve l'algébricité.

Nous prenons un nouveau symbole $\# \notin N$, le morphisme $h : N^* \longrightarrow (N\#)^*$ défini par $h(a) = a\#$ pour tout $a \in N$, et le morphisme $\pi : (N \cup \{\#\})^* \longrightarrow N^*$ défini par $\pi(\#) = \varepsilon$ et $\pi(a) = a$ pour tout $a \in N$.

Nous transformons R en la relation suivante :

$$S \;=\; \{\; (a, h(u)) \mid a \in N \;\wedge\; a \;R\; u \;\} \;\cup\; \{\; (\#, \#h(u)) \mid \varepsilon \;R\; u \;\}.$$

Ainsi S est algébrique avec $\mathrm{Dom}(S) \;=\; N \cup \{\#\}$. En outre, pour tout $L \subseteq N^*$,

$$\xrightarrow[R]{*}(L) \quad = \quad \pi\big(\xrightarrow[S]{*}(\#h(L))\big).$$

Par (i), on obtient que $\xrightarrow[R]{*}$ préserve l'algébricité.

iii) Soient $R \;\subseteq\; (N \cup \{\varepsilon\}) \times N^*$ et $L \in \mathrm{Reg}(N^*)$.

Montrons que $\xrightarrow[R^{-1}]{*}(L)$ reste régulier, et de façon effective pour R algébrique.

Soit A un automate fini sur N reconnaissant $L(A) = L$.

Nous complétons A en un automate B qui est le plus petit point fixe de l'équation suivante :

$$B \;=\; A \;\cup\; \{\; p \xrightarrow{a} q \mid \exists\, u \;(a \;R\; u \;\wedge\; p \xRightarrow[B]{u} q) \;\}.$$

Cela signifie que $B \;=\; \bigcup_{n \geq 0} B_n$ avec

$$\begin{array}{rcl} B_0 & = & A \\ B_{n+1} & = & B_n \;\cup\; \{\; p \xrightarrow{a} q \mid \exists\, u \;(a \;R\; u \;\wedge\; p \xRightarrow[B_n]{u} q) \;\} \qquad \text{pour tout } \; n \geq 0. \end{array}$$

Comme A est fini, cette saturation s'arrête toujours

$$\exists\, m \;(B_m \;=\; B_{m+1}) \quad \text{donc} \quad B \;=\; B_m.$$

En fait, B a les mêmes états que A, et ses étiquettes sont dans $N \cup \{\varepsilon\}$.

Pour R algébrique, cette construction est effective : $\mathrm{Alg}(N^*)$ a un problème du vide décidable, et est effectivement fermé par intersection avec un langage régulier.

En outre, pour tous les états p, q et pour tout mot $u \in N^*$, nous avons

$$p \xRightarrow[B]{u} q \quad \Longleftrightarrow \quad \exists\, v \;(p \xRightarrow[A]{v} q \;\wedge\; u \xrightarrow[R]{*} v)$$

donc $L(B) \;=\; \xrightarrow[R^{-1}]{*}(L(A))$ qui est régulier.

□

Une autre classe de systèmes alg/reg-préservants est définie dans [Hi 74]. Un système R est appelé *contexte limité* s'il existe un ordre partiel $<$ sur N tel que pour toute règle $(u, v) \in R$, toute lettre de u a une lettre dans v strictement plus grande : $\forall\ a \in \mathrm{Alph}(u)\ \exists\ b \in \mathrm{Alph}(v)\ a < b$. Il a été montré que la relation de dérivation de tout système contexte-limité fini préserve l'algébricité des langages [Hi 74]. En plus, l'inverse R^{-1} d'un système contexte-limité de R est appelé *système effaçant*, et la relation de dérivation de tout système effaçant fini préserve la régularité [HW 04].

Proposition 11 ([Hi 74, HW 04]) *Les systèmes finis contexte-limités sont alg/reg-préservants.*

Cette proposition résulte de la décomposition, présentée en [HW 04], de la relation de dérivation d'un système effaçant fini R en une substitution finie h sur un alphabet étendu, composée avec la dérivation de l'inverse d'un système algébrique fini S, et suivie par une restriction à l'alphabet d'origine :

$$\xrightarrow[R]{*} \ = \ \big(h \circ \xrightarrow[S^{-1}]{*}\big) \cap N^* \times N^*.$$

Au paragraphe suivant, nous étendons cette décomposition à tout système de récriture. Nous verrons en particulier que dans le cas des systèmes effaçants, S^{-1} peut toujours être choisi pour être le système de Dyck.

5.2.3 Décomposition

Dans ce paragraphe, nous reprenons le principe de décomposition de la proposition 11 pour définir une décomposition de la dérivation de tout système de récriture. Comme déjà esquissé au paragraphe précédent, l'application d'une règle de récriture simple (uv, w) à un mot x peut être simulée par l'insertion du facteur $\overleftarrow{u}\, w \overrightarrow{v}$ dans x, puis par la suppression des lettres fléchées en utilisant le système de Dyck. Nous nous servons de cette idée en identifiant les règles qui dans la dérivation peuvent être simulées par ce processus.

Plus précisément pour tout système de récriture R sur un alphabet N, nous identifions un sous-ensemble de règles $R' \subseteq R$ tel que

$$\xrightarrow[R]{*} \ = \ \big(h \circ \xrightarrow[(R-R') \cup D]{*}\big) \cap N^* \times N^*$$

où h est une substitution ajoutant des facteurs de la forme $\overleftarrow{u}\, w \overrightarrow{v}$. Cette décomposition est effectuée en éliminant la récursion à gauche ou à droite du système. Formellement, pour toute relation $R \subseteq N^* \times N^*$ et pour tout $M \subseteq N$, nous définissons le sous-système

$$R_M \ = \ \{(u, v) \in R \mid \mathrm{Alph}(uv) \cap M \neq \emptyset\}$$

composé de toutes les règles de R ayant au moins une lettre dans M ; donc $R - R_M$ est le sous-système maximal de R sur l'alphabet $N - M$. Nous voulons décomposer la dérivation de R en la composition d'une substitution h, suivie de la dérivation du système $(R - R_M) \cup D$ pour un sous-ensemble approprié M de N.

Définition. Un ensemble $M \subseteq \mathrm{Alph}(R)$ est appelé un *sous-alphabet préfixe* de R si

$$R \subseteq MN^* \times M(N-M)^* \cup N^* \times (N-M)^*.$$

Cette définition signifie que pour chaque règle $(u, av) \in R$ avec $a \in N$, v n'a pas de lettre dans M, et si $a \in M$ alors u doit commencer par une lettre de M (voir l'exemple 5.2.3). Notons que l'ensemble des sous-alphabets préfixes de R est fermé par union, et que l'on peut calculer son élément maximal (pour l'inclusion). Pour tout sous-alphabet préfixe M de R, on définit sur $\overline{N}$ le langage :

$$P = \{ \overleftarrow{u}\, w\, \overrightarrow{v} \mid uv\, R\, w \wedge u \in (N-M)^* \wedge v \in MN^* \}$$

et la substitution

$$h_M : \overline{N} \longrightarrow 2^{\overline{N}^*} \quad \text{avec} \quad h_M(a) = P^* a \quad \text{si} \quad a \in M \quad \text{et} \quad h_M(a) = a \quad \text{sinon.}$$

Le langage P et la substitution h_M sont réguliers lorsque R_M est reconnaissable. On a besoin d'une opération binaire sur les langages.

L'*insertion* d'un langage $M \subseteq N^*$ dans un langage $L \subseteq N^*$ est le langage

$$\begin{aligned} L[M] &= \{\, u_1 v_1 \ldots u_n v_n u_{n+1} \mid n \geq 0 \ \wedge\ u_1 \ldots u_{n+1} \in L \ \wedge\ v_1, \ldots, v_n \in M \,\} \\ &= \bigcup \{\, M^* a_1 M^* \ldots a_n M^* \mid n \geq 0 \ \wedge\ a_1, \ldots, a_n \in N \ \wedge\ a_1 \ldots a_n \in L \,\} \end{aligned}$$

obtenu en insérant des mots de M dans les mots de L. Pour tout $u \in N^*$, on écrit

$$\|u\|_{L,M} = \min\{\, n_0 + \ldots + n_p \mid p, n_0, \ldots, n_p \geq 0 \ \wedge\ \exists\, a_1, \ldots, a_p \in N \\ (a_1 \ldots a_p \in L \ \wedge\ u \in M^{n_0} a_1 M^{n_1} \ldots a_p M^{n_p}) \,\}$$

le nombre minimal d'insertions de mots de M pour obtenir u à partir de L.

En particulier $(\|u\|_{L,M} = \infty \iff u \notin L[M])$ et $(\|u\|_{L,M} = 0 \iff u \in L)$.

Cette opération peut être exprimée en utilisant une substitution et un morphisme inverse.

On prend un nouveau symbole $\# \notin N$, le morphisme $\pi : (N \cup \{\#\})^* \longrightarrow N^*$ et la substitution $h : (N \cup \{\#\})^* \longrightarrow 2^{N^*}$ définie par

$$\pi(a) = h(a) = a \ \text{ pour tout } \ a \in N\,, \quad \pi(\#) = \varepsilon \ \text{ et } \ h(\#) = M.$$

Aussi $L[M] = h(\pi^{-1}(L))$ d'où pour M régulier, $L[M]$ est régulier (resp. algébrique) pour L régulier (resp. algébrique).

Pour tout $R \subseteq N^* \times N^*$, on note

$$\overleftrightarrow{R} = \{ \overleftarrow{u}\, w\, \overrightarrow{v} \mid (uv, w) \in R \}.$$

Même en insérant au hasard des facteurs de $\overleftrightarrow{R}$ dans un mot, on n'augmente pas l'ensemble des mots de N^* obtenus par dérivation en utilisant $R \cup D$.

Lemme 5.2.2 *Pour tout $R \subseteq N^* \times N^*$ et pour tout $u \in \overline{N}^*$,*

$$\xrightarrow[R \cup D]{*}\big(u[\overleftrightarrow{R}]\big) \cap N^* \subseteq \xrightarrow[R \cup D]{*}(u).$$

Démonstration. On écrit $R' = R \cup D$.
i) Tout d'abord, nous avons besoin d'établir que chaque fois qu'un facteur $\overrightarrow{v}$ (resp. $\overleftarrow{v}$) peut être supprimé au cours d'une dérivation selon R', on peut toujours réorganiser la dérivation pour que v apparaisse à la droite (resp. gauche) de $\overrightarrow{v}$ (resp. $\overleftarrow{v}$), puis que le facteur résultant $\overrightarrow{v}v$ (resp. $v\overleftarrow{v}$) soit effacé en utilisant D.

Formellement, pour tout $u, w \in \overline{N}^*$ et pour tout $v, z \in N^*$,

$$w\overrightarrow{v}u \xrightarrow[R']{*} z \implies \exists\, \overline{w},\ u \xrightarrow[R']{*} v\overline{w} \wedge w\overline{w} \xrightarrow[R']{*} z,$$
$$u\overleftarrow{v}\,w \xrightarrow[R']{*} z \implies \exists\, \overline{w},\ u \xrightarrow[R']{*} \overline{w}v \wedge \overline{w}w \xrightarrow[R']{*} z.$$

Vérifions la première implication par récurrence sur $n \geq 0$ pour $w\overrightarrow{v}u \xrightarrow[R']{n} z$.
$n = 0$: on a $w\overrightarrow{v}u = z \in N^*$.

Aussi $v = \varepsilon$ et $wu = z$. Donc $\overline{w} = u$ convient.

$n \Longrightarrow n+1$: soit $w\overrightarrow{v}u \xrightarrow[R']{n+1} z$.

On distingue les trois cas complémentaires ci-dessous.

Cas 1 : il existe u' tel que $u \xrightarrow[R']{} u'$ et $w\overrightarrow{v}u' \xrightarrow[R']{n} z$.
Par hypothèse de récurrence, il existe $\overline{w}$ tel que

$$u' \xrightarrow[R']{*} v\overline{w} \text{ et } w\overline{w} \xrightarrow[R']{*} z.$$

Aussi $u \xrightarrow[R']{*} v\overline{w}$.

Cas 2 : il existe w' tel que $w \xrightarrow[R']{} w'$ et $w'\overrightarrow{v}u \xrightarrow[R']{n} z$.
Par hypothèse de récurrence, il existe $\overline{w}$ tel que

$$u \xrightarrow[R']{*} v\overline{w} \text{ et } w'\overline{w} \xrightarrow[R']{*} z.$$

Aussi $w\overline{w} \xrightarrow[R']{*} z$.

Cas 3 : il existe $b \in N$ tel que $u = bu'$, $v = bv'$ et $w\overrightarrow{v'}u' \xrightarrow[R']{n} z$.

Par hypothèse de récurrence, il existe $\overline{w}$ tel que

$$u' \xrightarrow[R']{*} v'\overline{w} \text{ et } w\overline{w} \xrightarrow[R']{*} z.$$

Aussi $u = bu' \xrightarrow[R']{*} bv'\overline{w} = v\overline{w}$.

ii) Soit $x \xrightarrow[R']{*} y$ avec $x \in u[\overleftrightarrow{R}]$, $u \in \overline{N}^*$ et $y \in N^*$.

Montrons que $u \xrightarrow[R']{*} y$ par récurrence sur $\|x\|_{u,\overleftrightarrow{R}} \geq 0$.

$\|x\|_{u,\overleftrightarrow{R}} = 0$: $x = u$ d'où $u \xrightarrow[R']{*} y$.

$\|x\|_{u,\overleftrightarrow{R}} > 0$: il existe uv R w, $x' \in u'[\overleftrightarrow{R}]$ et $x'' \in u''[\overleftrightarrow{R}]$ tel que

$$u = u'u'' \text{ et } x = x'\overleftarrow{u}\, w\, \overrightarrow{v}x''.$$

Comme $x'\overleftarrow{u}\,(w\,\overrightarrow{v}x'') \xrightarrow[R']{*} y \in N^*$ et par (i), il existe $\overline{w}$ tel que

$$x' \xrightarrow[R']{*} \overline{w}u \text{ et } \overline{w}w\,\overrightarrow{v}x'' \xrightarrow[R']{*} y.$$

Comme $(\overline{w}w)\,\overrightarrow{v}x'' \xrightarrow[R']{*} y \in N^*$ et par (i), il existe $\overline{\overline{w}}$ tel que

$$x'' \xrightarrow[R']{*} v\overline{\overline{w}} \text{ et } \overline{w}w\overline{\overline{w}} \xrightarrow[R']{*} y.$$

Aussi $x'x'' \xrightarrow[R']{*} \overline{w}uv\overline{\overline{w}} \xrightarrow[R]{} \overline{w}w\overline{\overline{w}} \xrightarrow[R']{*} y$.

Comme $\|x'x''\|_{u,\overleftrightarrow{R}} = \|x\|_{u,\overleftrightarrow{R}} - 1$ et $x'x'' \in u[\overleftrightarrow{R}]$, $u \xrightarrow[R']{*} y$ par hyp. de réc.

□

Quand M est un sous-alphabet préfixe de R, on peut décomposer $\xrightarrow[R]{*}$ en supprimant R_M de R.

Lemme 5.2.3 *Pour tout sous-alphabet préfixe M de R et pour tout $u \in \overline{N}^*$,*

$$\xrightarrow[R\cup D]{*}(u) \cap N^* = \xrightarrow[(R-R_M)\cup D]{*}\big(h_M(u)\big) \cap N^*.$$

Démonstration. Ecrivons $S = (R - R_M) \cup D$ et $h = h_M$.

$\supseteq$ **:** Soit $u \in \overline{N}^*$. On a $h(u) \subseteq u[\overleftrightarrow{R}]$ et par le lemme 5.2.2, nous obtenons

$$\xrightarrow[S]{*}(h(u)) \cap N^* \subseteq \xrightarrow[R\cup D]{*}\big(u[\overleftrightarrow{R}]\big) \cap N^* \subseteq \xrightarrow[R\cup D]{*}(u) \cap N^*.$$

$\subseteq$: Soit $u \xrightarrow[R\cup D]{*} v$ avec $u \in \overline{N}^*$. Montrons que $h(v) \subseteq \xrightarrow[S]{*}\big(h(u)\big)$.

Pour démontrer cette inclusion, nous devons faire en sorte que le processus d'insertion réalisé par h ne fait pas obstacle à un mot dérivable de u selon $R \cup D$ de façon à ce qu'il soit aussi dérivable de $h(u)$ selon S. Intuitivement, cela est garanti par la définition de l'ensemble P et de la substitution h_M qui insère des facteurs à des positions spécifiques.

Par récurrence sur la longueur de la dérivation de u à v, il reste à vérifier l'inclusion pour $u \xrightarrow[R\cup D]{} v$.

Soient $u = xu_0y$ et $v = xv_0y$ pour un $(u_0, v_0) \in R \cup D$.

Nous distinguons les trois cas complémentaires ci-dessous.

Cas 1 : $(u_0, v_0) \notin R_M \cup D$.

Par définition $u_0, v_0 \in (N - M)^*$.

Cela signifie que u_0 et v_0 ne sont pas affectés par h : nous avons $h(u) = h(x)u_0h(y)$ et $h(v) = h(x)v_0h(y)$. Donc

$$h(v) \;=\; h(x)v_0h(y) \;\subseteq\; \xrightarrow[\{(u_0,v_0)\}]{}\big(h(x)u_0h(y)\big) \;\subseteq\; \xrightarrow[S]{*}\big(h(u)\big).$$

Cas 2 : $(u_0, v_0) \in D$.

Par définition, $v_0 = \varepsilon$. Ainsi

$$h(v) \;=\; h(x)h(y) \;\subseteq\; \xrightarrow[D]{}\big(h(x)u_0h(y)\big) \;\subseteq\; \xrightarrow[S]{*}\big(h(xu_0y)\big) \;=\; \xrightarrow[S]{*}\big(h(u)\big).$$

Cas 3 : $(u_0, v_0) \in R_M$.

Cette règle peut être de deux types correspondants aux deux sous-cas ci-dessous.

Cas 3.1 : $u_0 \in MN^*$ et $v_0 \in M(N - M)^*$.

Nous avons $h(x)P^*u_0h(y) \subseteq h(u)$ et $h(v) = h(x)P^*v_0h(y)$.

Comme $v_0\overrightarrow{u_0} \in P$, $h(x)P^*v_0\overrightarrow{u_0}u_0h(y) \subseteq h(u)$. Donc

$$h(v) \;=\; h(x)P^*v_0h(y) \;\subseteq\; \xrightarrow[D]{*}(h(x)P^*v_0\overrightarrow{u_0}u_0h(y)) \;\subseteq\; \xrightarrow[S]{*}(h(u)).$$

Cas 3.2 : $u_0 \in N^*MN^*$ et $v_0 \in (N - M)^*$.

Nous avons $u_0 = u_0'\#u_0''$ avec $u_0' \in (N - M)^*$, $\# \in M$, $u_0'' \in N^*$, et $\overleftarrow{u_0'}v_0\#\overrightarrow{u_0''} \in P$.

Donc $h(x)u_0'P^*\#u_0''h(y) \subseteq h(u)$, ce qui implique que

$$h(x)u_0'\overleftarrow{u_0'}v_0\overrightarrow{u_0''}\overrightarrow{\#}\#u_0''h(y) \;\subseteq\; h(u).$$

Enfin, nous obtenons

$$h(v) = h(x)v_0h(y) \;\subseteq\; \xrightarrow[D]{*}\big(h(x)u_0'\overleftarrow{u_0'}v_0\overrightarrow{u_0''}\overrightarrow{\#}\#u_0''h(y)\big) \;\subseteq\; \xrightarrow[S]{*}(h(u)).$$

Ceci conclut la démonstration que $h(v) \subseteq \xrightarrow[S]{*}(h(u))$ pour $u \xrightarrow[R \cup D]{}{}^* v$.

Comme $v \in h(v)$, nous obtenons finalement $\xrightarrow[R \cup D]{}{}^*(u) \subseteq \xrightarrow[S]{*}(h(u))$.

□

Une décomposition similaire peut également être obtenue en utilisant les sous-alphabets suffixes au lieu de préfixes. Un sous-ensemble M de N est un *sous-alphabet suffixe* de R, si

$$R \subseteq N^*M{\times}(N-M)^*M \cup N^*{\times}(N-M)^*.$$

Il peut également être considéré comme un sous-alphabet préfixe de $\widetilde{R} = \{(\widetilde{u}, \widetilde{v}) \mid u\,R\,v\}$ où $\widetilde{u} = u(|u|)\ldots u(1)$ est le *miroir* du mot u. Le lemme 5.2.3 reste vrai pour n'importe quel sous-alphabet suffixe M, à la différence que

$$h_M(x) = xQ^* \text{ pour tout } x \in M$$

avec

$$Q = \{\, \overleftarrow{u}\, w\, \overrightarrow{v} \mid uv\,R\,w \wedge u \in N^*M \wedge v \in (N-M)^* \,\}.$$

En utilisant des sous-alphabets préfixes et suffixes, nous pouvons maintenant effectuer une itération de ce mécanisme de décomposition, tant qu'il reste un sous-alphabet préfixe ou suffixe.

Nous disons que $R \subseteq N^*{\times}N^*$ est *u-décomposable* pour $u \in (2^N)^*$ si $u = \varepsilon$, ou $u = Mv$ avec M un sous-alphabet préfixe ou suffixe de R, et $R - R_M$ est v-décomposable.

Pour tout $u \in (2^N)^*$, nous définissons le sous-système R_u de R comme

$$R_u = R_{u(1)} \cup \ldots \cup R_{u(|u|)} = R_{u(1) \cup \ldots \cup u(|u|)}$$

composé du sous-ensemble de règles de R ayant au moins une lettre dans $u(1) \cup \ldots \cup u(|u|)$.

Lorsque les lettres de u sont des sous-alphabets préfixes et suffixes, nous définissons la substitution $h_u : \overline{N} \longrightarrow 2^{\overline{N}^*}$ par

$$h_u = h_{u(1)} \circ \ldots \circ h_{u(|u|)}$$

dont pour tout $1 \leq i \leq |u|$, $h_{u(i)}$ est la substitution associée au sous-alphabet préfixe $u(i)$ de R, ou suffixe mais pas préfixe.

Notons que si R_u est reconnaissable, h_u est une substitution régulière.

Itérons maintenant la décomposition du lemme 5.2.3.

Proposition 12 *Si R est u-décomposable alors*

$$\xrightarrow[R]{*} \;=\; \big(h_u \circ \xrightarrow[(R-R_u)\cup D]{*}\big) \cap N^*\times N^*.$$

Démonstration. Nous avons

$$\begin{aligned}
\xrightarrow[R]{*} \;&=\; \xrightarrow[R\cup D]{}{}^* \;\cap\; N^*\times N^* \quad \text{(pour le cas } u=\varepsilon\text{)}\\
&=\; \big(h_{u(1)} \circ \xrightarrow[(R-R_{u(1)})\cup D]{*}\big) \;\cap\; N^*\times N^* \quad \text{par le lemme 5.2.3}\\
&=\; \big(h_{u(1)} \circ \ldots \circ h_{u(|u|)} \circ \xrightarrow[(R-R_{u(1)}\ldots-R_{u(|u|)})\cup D]{*}\big) \;\cap\; N^*\times N^*\\
&=\; \big(h_u \circ \xrightarrow[(R-R_u)\cup D]{*}\big) \;\cap\; N^*\times N^*.
\end{aligned}$$

□

On dit que R est *décomposable* en S si R est u-décomposable pour un certain u avec $R - R_u = S$. Cette relation de décomposition est réflexive et transitive. Illustrons ce mécanisme dans l'exemple ci-dessous.

Exemple. Considérons le système de récriture

$$R = \{(abb, ab), (a, \varepsilon), (cb, cc)\}$$

et la dérivation

$$caabb \underset{R}{\longrightarrow} caab \underset{R}{\longrightarrow} cab \underset{R}{\longrightarrow} cb \underset{R}{\longrightarrow} cc.$$

Comme $\{a\}$ est un sous-alphabet préfixe de R et par le lemme 5.2.3, nous avons

$$\xrightarrow[R]{*} \;=\; \big(h_a \circ \xrightarrow[R'\cup D]{*}\big) \cap N^*\times N^*$$

avec $R' = \{(cb, cc)\}$ et $h_a(a) = \{\overrightarrow{a}, ab\,\overrightarrow{b}\,\overrightarrow{b}\,\overrightarrow{a}\}^*a$.
Comme $\{b\}$ est un sous-alphabet préfixe de R' (mais pas de R), nous avons

$$\xrightarrow[R'\cup D]{*} \cap\, \overline{N}^*\times N^* \;=\; \big(h_b \circ \xrightarrow[D]{*}\big) \cap\; \overline{N}^*\times N^*$$

avec $h_b(b) = \{\overleftarrow{c}\,cc\,\overrightarrow{b}\}^*b$.

Par conséquent, R est $\{a\}\{b\}$-décomposable en $\emptyset$ et

$$\xrightarrow[R]{*} \;=\; \big(h_{ab} \circ \xrightarrow[D]{*}\big) \cap N^* \times N^*$$

avec $h_{ab}(a) \;=\; h_b(h_a(a)) \;=\; \{\overrightarrow{a}, a\{\overleftarrow{c}\, cc\, \overrightarrow{b}\}^* b\, \overrightarrow{b}\, \overrightarrow{b}\, \overrightarrow{a}\}^* a$

et $h_{ab}(b) \;=\; h_b(h_a(b)) \;=\; h_b(b) \;=\; \{\overleftarrow{c}\, cc\, \overrightarrow{b}\}^* b$. Ainsi

$$u \;=\; c.\overrightarrow{a}\, a.\overrightarrow{a}\, a\overleftarrow{c}\, cc\, \overrightarrow{b}\, b\, \overrightarrow{b}\, \overrightarrow{b}\, \overrightarrow{a}\, a.b.b \in h_{ab}(caabb) \quad \text{et} \quad u \xrightarrow[D]{*} cc.$$

Enfin, R est nœthérien (pas de dérivation infinie) bien que R ne soit pas « match-bounded » [GHW 04].

Pour toute lettre $a \in N - \mathrm{Im}(R)$ qui n'apparaît pas dans les membres droits des règles de R, $\{a\}$ est un sous-alphabet préfixe (ou suffixe) de R, et nous appelons a une *lettre réductible* de R.

On dit que R est *réductible* en S si R est *u-décomposable* en S pour un mot u composé uniquement de lettres réductibles (pour les sous-relations restantes successives). Notons que ceci n'est pas le cas pour le système de récriture de l'exemple précédent, même si il est décomposable en le système vide.

Les systèmes qui peuvent être réduits en $\emptyset$ ou en $\{(\varepsilon, \varepsilon)\}$ sont exactement les systèmes effaçants.

Proposition 13 *R est effaçant $\iff$ R est réductible en $\emptyset$ ou en $\{(\varepsilon, \varepsilon)\}$.*

Démonstration. On peut supposer que $(\varepsilon, \varepsilon) \notin R$ puisque

$$\begin{array}{rcl} R \text{ est effaçant} & \iff & R - \{(\varepsilon,\varepsilon)\} \text{ est effaçant,} \\ R \text{ est réductible en } \emptyset \text{ ou en } \{(\varepsilon,\varepsilon)\} & \iff & R - \{(\varepsilon,\varepsilon)\} \text{ est réductible en } \emptyset. \end{array}$$

$\Longrightarrow$: Par récurrence forte sur $|\mathrm{Alph}(\mathrm{Dom}(R))| \geq 0$.

Cas de base : $|\mathrm{Alph}(\mathrm{Dom}(R))| = 0$. Aussi $R = \emptyset = R_\varepsilon$.

Cas récurrent : $|\mathrm{Alph}(\mathrm{Dom}(R))| > 0$.

R est effaçant pour un certain ordre partiel $<$.

Soit $a \in \mathrm{Max}\big(\mathrm{Alph}(\mathrm{Dom}(R)\big)$.

Comme R est effaçant, $a \notin \mathrm{Alph}(\mathrm{Im}(R))$ *i.e.* a est une lettre réductible de R.

Alors $\mathrm{Alph}(R - R_a) \;=\; \mathrm{Alph}(R) - \{a\}$, et $R - R_a$ reste effaçant pour $<$.

Par hypothèse de récurrence $R - R_a \;=\; R_v$ pour un certain mot v de lettres réductibles de R. Ainsi $R = R_{av}$.

$\Longleftarrow$: Soit $R = R_u$ pour un certain mot u de lettres réductibles de R.

On peut supposer que u n'a que des lettres distinctes.

Nous prenons l'ordre partiel $<$ défini par

$$a < u(|u|) < \ldots < u(1) \text{ pour tout } a \in \mathrm{Alph}(R) - \mathrm{Alph}(u).$$

Pour tout $(v, w) \in R$, il existe un unique $1 \leq i \leq |u|$ tel que

$$(v, w) \in R_{u(i)} - R_{u(1)\ldots u(i-1)}.$$

Alors $u(i) \in \mathrm{Alph}(v)$ et $u(1), \ldots, u(i) \notin \mathrm{Alph}(w)$.

Ainsi $a < u(i)$ pour tout $a \in \mathrm{Alph}(w)$. Finalement R est effaçant pour $<$.

□

Lorsque R est décomposable en S et $R - S$ est reconnaissable, nous disons qu'il y a une *décomposition reconnaissable* de R en S. Appliquons la proposition 12 à toute décomposition reconnaissable.

Théorème 14 *Pour toute décomposition reconnaissable de R en S,*

$$\xrightarrow[S\cup D]{*} \textit{ préserve la régularité} \implies \xrightarrow[R]{*} \textit{ préserve la régularité,}$$
$$\xrightarrow[S^{-1}\cup D^{-1}]{*} \textit{préserve l'algébricité} \implies \xrightarrow[R^{-1}]{*} \textit{ préserve l'algébricité.}$$

Démonstration. Par la proposition 12, il y a une substitution régulière h telle que pour tout $L \subseteq N^*$,

$$\xrightarrow[R]{*}(L) = \xrightarrow[S\cup D]{}{}^*(h(L)) \cap N^* \quad \text{et} \quad \xrightarrow[R^{-1}]{*}(L) = h^{-1}\Big(\xrightarrow[S^{-1}\cup D^{-1}]{}{}^*(L)\Big) \cap N^*.$$

Comme h est une substitution régulière, ceci prouve le théorème.

□

Notons que nous ne pouvons pas effacer D ou D^{-1} dans le théorème 14, et que les réciproques des implications sont fausses. En effet, le système $S = \{(\#a, bb\#), (b\&, \&a)\}$ a une dérivation rationnelle, donc sa dérivation préserve la régularité et l'algébricité, mais $\xrightarrow[S\cup D]{*}$ ne préserve pas la régularité :

$$\xrightarrow[S\cup D]{*}\Big((\#\overrightarrow{\&})^*\#a(\overleftarrow{\#}\&)^*\Big) \cap \#a^* = \{\#a^{2^n} \mid n \geq 0\}$$

et $\xrightarrow[S\cup D^{-1}]{*}$ ne préserve pas l'algébricité :

$$\xrightarrow[S\cup D^{-1}]{*}(a) \cap (\overrightarrow{\#}\&)^* a^* (\overleftarrow{\&}\#)^* \;=\; \{(\overrightarrow{\#}\&)^n a^{2^n} (\overleftarrow{\&}\#)^n \mid n \geq 0\}.$$

Pour conclure ce paragraphe, nous appliquons le théorème 14 avec la proposition 10 pour transférer les propriétés de préservation de la régularité et de l'algébricité des systèmes algébriques à une classe plus générale de systèmes de récriture.

Proposition 15 *Pour toute décomposition reconnaissable de R^{-1} en S^{-1} avec S un système algébrique, le système R est alg/reg-préservant.*

Par la proposition 13, cette proposition généralise strictement la proposition 11 en permettant un ensemble reconnaissable des règles. En effet, tout système reconnaissable effaçant a une décomposition reconnaissable en le système vide (ou en $\{(\varepsilon, \varepsilon)\}$) et dont l'inverse est trivialement un système algébrique. Cela implique que les systèmes contexte-limités reconnaissables sont alg/reg-préservant.

Au paragraphe suivant, nous donnons quelques autres applications du théorème 14 et de la proposition 15.

5.3 Applications

Dans ce paragraphe, nous donnons plusieurs conséquences et applications de la technique de décomposition qui a été présentée au paragraphe précédent. En particulier, nous montrons comment déduire du théorème 14 les propriétés de préservation pour les classes des systèmes préfixes, suffixes et bifixes ainsi que leurs variantes avec ajout de marques.

5.3.1 Systèmes préfixes, suffixes et bifixes

La proposition 11 a déjà été appliquée en [HW 04] pour la dérivation préfixe des systèmes finis. En utilisant le théorème 14, elle peut être étendue à tout système reconnaissable.

La *récriture préfixe* d'un système R est la relation binaire

$$\underset{R}{\longmapsto} \;=\; R.N^* \;=\; \underset{R,0}{\longrightarrow}$$

à savoir que $uy \underset{R}{\longmapsto} vy$ pour toute règle $u\,R\,v$ et pour tout mot $y \in N^*$.

La *dérivation préfixe* $\xrightarrow[R]{*}$ de R est la fermeture réflexive et transitive pour la composition de la relation de récriture préfixe.

Pour tout système fini, la régularité de l'ensemble des mots atteints par dérivation préfixe à partir d'un mot donné [Bu 64] est un cas particulier de la régularité de la dérivation préfixe ; cela reste vrai pour n'importe quel système reconnaissable.

Proposition 16 ([Ca 90]) *La dérivation préfixe de tout système reconnaissable est une relation rationnelle.*

Démonstration. Soit R un système de récriture reconnaissable,

$$R \;=\; \bigcup_{i=1}^{n} U_i \times V_i$$

Soit $\# \notin N$ un nouveau symbole. On considére le système

$$\#R \;=\; \{(\#u, \#v) \mid u\, R\, v\}.$$

Par définition, $\{\#\}$ est un sous-alphabet préfixe de $\#R$, qui est reconnaissable et $\#$-décomposable en $\emptyset$.

Pour tout $L \subseteq N^*$,

$$\xrightarrow[R]{*}(L) \;=\; {}^{\#}{}^{-1}\big(\xrightarrow[\#R]{*}(\#L)\big)$$

qui, par le théorème 14, est régulier dès que L est un langage régulier.

Par la proposition 12, la dernière égalité est équivalente à

$$\xrightarrow[R]{*}(L) \;=\; \xrightarrow[N\downarrow]{*}\big(\{v\,\overrightarrow{u} \mid u\, R\, v\}^* L\big) \cap N^*.$$

Puisque $\xrightarrow[R^{-1}]{*}(U)$ et $\xrightarrow[R]{*}(V)$ restent réguliers pour tous langages réguliers U et V, la relation

$$\overline{R} \;=\; \bigcup_{i=1}^{n} \xrightarrow[R^{-1}]{*}(U_i) \times \xrightarrow[R]{*}(V_i)$$

est reconnaissable, donc

$$\xrightarrow[R]{*} \;=\; \mathrm{Id}_{N^*} \cup \longmapsto_{\overline{R}} \;=\; \mathrm{Id}_{N^*} \cup \overline{R}.N^*$$

est reconnue par un transducteur fini.

□

Les règles d'un système R peuvent également être appliquées uniquement par suffixe. La *récriture suffixe* de R est la relation binaire

$$\underset{R}{\longrightarrow\!\!\!\!\!\rightarrow} \;=\; N^*.R$$

à savoir que $wu \underset{R}{\longrightarrow\!\!\!\!\!\rightarrow} wv$ pour tout $u\,R\,v$ et $w \in N^*$.

Notons que la relation de récriture d'un système suffixe est isomorphe à celle d'un système préfixe :

$$u \underset{R}{\longrightarrow\!\!\!\!\!\rightarrow} v \quad\Longleftrightarrow\quad \widetilde{u} \underset{\widetilde{R}}{\longmapsto} \widetilde{v} \quad \text{o} \quad \widetilde{R} \;=\; \{(\widetilde{u},\widetilde{v}) \mid u\,R\,v\}.$$

Donc, la proposition 16 est vraie pour les systèmes suffixes. Enfin, nous autorisons d'appliquer les règles à la fois par préfixe et par suffixe. La *relation de récriture bifixe* de R est

$$\underset{R}{\longmapsto\!\!\!\!\!\rightarrow} \;=\; \underset{R}{\longmapsto} \cup \underset{R}{\longrightarrow\!\!\!\!\!\rightarrow}.$$

Il existe une généralisation de la proposition 16 à ce type de récriture.

Proposition 17 ([KKO 06]) *La dérivation bifixe d'un système reconnaissable est une relation rationnelle.*

Démonstration. Nous prenons deux nouveaux symboles $\#, \& \notin N$ et on définit le système reconnaissable

$$S \;=\; \#R \cup R\& \;=\; \{\,(\#u, \#v) \mid u\,R\,v\,\} \cup \{\,(u\&, v\&) \mid u\,R\,v\,\}.$$

Les ensembles $\{\#\}$ et $\{\&\}$ sont respectivement des sous-aphabets préfixe et suffixe de S. Ainsi S est $\#\&$-décomposable en $\emptyset$. Pour tout $L \subseteq N^*$,

$$\overset{*}{\underset{R}{\longmapsto\!\!\!\!\!\rightarrow}}(L) \;=\; \#^{-1}\big(\overset{*}{\underset{S}{\longrightarrow}}(\#L\&)\big)\&^{-1}$$

qui d'après le théorème 14, est régulier dès que L est régulier. Par la proposition 12, la dernière égalité est équivalente à

$$\overset{*}{\underset{R}{\longmapsto\!\!\!\!\!\rightarrow}}(L) \;=\; \overset{*}{\underset{D}{\longrightarrow}}\big(\{v\,\overrightarrow{u} \mid u\,R\,v\}^* L \{\overleftarrow{u}\,v \mid u\,R\,v\}^*\big) \cap N^*.$$

Il s'agit de la première étape de la construction donnée en [KKO 06], d'un transducteur fini reconnaissant $\overset{*}{\underset{R}{\longmapsto\!\!\!\!\!\rightarrow}}$.

□

5.3.2 Dérivation gauche-droite

Appliquons le théorème 14 à une autre restriction de la dérivation. La *dérivation gauche-droite* $\overset{*}{\underset{R}{\hookrightarrow}}$ d'un système R est définie par

$$u \overset{*}{\underset{R}{\hookrightarrow}} v \iff u_0 \underset{R,p_1}{\longrightarrow} u_1 \ldots \underset{R,p_n}{\longrightarrow} u_n \text{ avec } p_1 \leq \ldots \leq p_n, u_0 = u \text{ et } u_n = v.$$

La dérivation gauche-droite et la dérivation la plus à gauche sont incomparables. En particulier, l'application d'une règle de récriture à une position i pourrait dans une dérivation la plus à gauche permettre d'appliquer une autre règle à une position strictement plus petite que i. Toutefois dans une dérivation gauche-droite, les positions successives des récritures sont croissantes.

Proposition 18 *La dérivation gauche-droite d'un système reconnaissable préserve l'algébricité, et son inverse préserve la régularité.*

Démonstration. Nous considérons un nouveau symbole $\# \notin N$ et le système

$$S = \{ (u, \#v) \mid u \, R \, v \}.$$

En choisissant $\{\#\}$ comme sous-alphabet préfixe, on a une décomposition reconnaissable de S^{-1} en $\emptyset$. Notons que S^{-1} est effaçant pour R fini.
Par la proposition 15, $\overset{*}{\underset{S}{\longrightarrow}}$ préserve l'algébricité et $\overset{*}{\underset{S^{-1}}{\longrightarrow}}$ préserve la régularité.
De plus, la dérivation $\overset{*}{\underset{S}{\longrightarrow}}$ peut être effectuée de gauche à droite :

$$\overset{*}{\underset{S}{\longrightarrow}} = \overset{*}{\underset{S}{\hookrightarrow}}.$$

Soit π le morphisme défini par $\pi(\#) = \varepsilon$ et $\pi(a) = a$ pour tout $a \in N$. Nous avons

$$\overset{*}{\underset{R}{\hookrightarrow}} = \{(u, \pi(v)) \mid u \in N^* \wedge u \overset{*}{\underset{S}{\hookrightarrow}} v\}.$$

Ainsi, pour chaque $L \subseteq N^*$,

$$\overset{*}{\underset{R}{\hookrightarrow}}(L) = \pi\big(\overset{*}{\underset{S}{\hookrightarrow}}(L)\big) = \pi\big(\overset{*}{\underset{S}{\longrightarrow}}(L)\big).$$

Donc $\overset{*}{\underset{R}{\hookrightarrow}}(L)$ est algébrique lorsque L est algébrique. Enfin, pour tout $L \subseteq N^*$,

$$\big(\overset{*}{\underset{R}{\hookrightarrow}}\big)^{-1}(L) = \overset{*}{\underset{S^{-1}}{\longrightarrow}}\big(\pi^{-1}(L)\big) \cap N^*$$

qui est régulier lorsque L est régulier.
□

Notons que la proposition 10 devient un corollaire de la proposition 18 lorsqu'elle est limitée aux systèmes reconnaissables de récriture de mots. En effet, pour tout $R \subseteq (N \cup \{\varepsilon\}) \times N^*$, la dérivation $\xrightarrow[R]{*}$ est égale à $\underset{R}{\overset{*}{\hookrightarrow}}$. Notons également que les propriétés de préservation inverse ne sont pas vraies en général. Par exemple, lorsque $R = \{(a, bab)\}$, nous avons $\underset{R}{\overset{*}{\hookrightarrow}}(a) = \{b^n a b^n \mid n \geq 0\}$ donc $\underset{R}{\overset{*}{\hookrightarrow}}$ ne préserve pas la régularité. Inversement $\xrightarrow[N\downarrow^{-1}]{*} = \underset{N\downarrow^{-1}}{\overset{*}{\hookrightarrow}}$ donc $\left(\underset{N\downarrow^{-1}}{\overset{*}{\hookrightarrow}}\right)^{-1} = \xrightarrow[N\downarrow]{*}$ ne préserve pas l'algébricité.

Nous concluons ce paragraphe sur la dérivation gauche-droite en montrant que la dérivation de gauche à droite de tout système de récriture de mots peut être décrite en utilisant la dérivation préfixe. Pour tout $R \subseteq N^* \times N^*$, on associe le système de transitions étiquetées (*i.e.* le graphe étiqueté $\widehat{R}$ sur $N \cup \{\varepsilon\}$ défini par

$$\widehat{R} \;=\; \{\, u \xrightarrow{\varepsilon} v \mid u\, R\, v \,\} \cup \{\, a \xrightarrow{a} \varepsilon \mid a \in N \,\}$$

et son *graphe de transition préfixe*

$$\widehat{R}.N^* \;=\; \{\, uw \xrightarrow{\varepsilon} vw \mid u\, R\, v \wedge w \in N^* \,\} \cup \{\, aw \xrightarrow{a} w \mid a \in N \wedge w \in N^* \,\}.$$

Les mots obtenus par dérivation gauche-droite selon R à partir d'un mot u sont précisément les mots v qui étiquettent les chemins $u \underset{\widehat{R}.N^*}{\overset{v}{\Longrightarrow}} \varepsilon$ (en d'autres termes, *reconnus par* $\widehat{R}.N^*$) du sommet u au sommet ε.

Lemme 5.3.1 *Pour tout système R, $u \underset{R}{\overset{*}{\hookrightarrow}} v$ si et seulement si $u \underset{\widehat{R}.N^*}{\overset{v}{\Longrightarrow}} \varepsilon$.*

Démonstration. $\Longrightarrow$: On veut montrer que pour tout $n \geq 0$,

$$u \underset{R}{\overset{n}{\hookrightarrow}} v \quad\Longrightarrow\quad u \underset{\widehat{R}.N^*}{\overset{v}{\Longrightarrow}} \varepsilon.$$

Ceci est fait par récurrence sur $n \geq 0$.
$n = 0 : u = v$. Comme $\{\, a \xrightarrow{a} \varepsilon \mid a \in N \,\} \subseteq \widehat{R}$, on obtient $u \underset{\widehat{R}.N^*}{\overset{u}{\Longrightarrow}} \varepsilon$.
$n \Longrightarrow n+1$: soit $u \underset{R}{\overset{n+1}{\hookrightarrow}} v$. Il existe x, y, z, w, w' tel que

$$u \;=\; wxw' \;\wedge\; x\; R\; y \;\wedge\; yw' \underset{R}{\overset{n}{\hookrightarrow}} z \;\wedge\; wz = v.$$

Par hypothèse de récurrence, $yw' \underset{\widehat{R}.N^*}{\overset{z}{\Longrightarrow}} \varepsilon$ d'où

$$u \;=\; wxw' \underset{\widehat{R}.N^*}{\overset{w}{\Longrightarrow}} xw' \xrightarrow[\widehat{R}.N^*]{\varepsilon} yw' \underset{\widehat{R}.N^*}{\overset{z}{\Longrightarrow}} \varepsilon \quad \text{avec} \quad wz = v.$$

$\Longleftarrow$: Plus généralement, vérifions que $u \overset{*}{\underset{R}{\hookrightarrow}} vw$ pour $u \overset{v}{\underset{\widehat{R}.N^*}{\Longrightarrow}} w$.
La démonstration est faite par récurrence sur la longueur n de la dérivation

$$u \left(\overset{v}{\underset{\widehat{R}.N^*}{\Longrightarrow}}\right)^n w.$$

$n = 0 : v = \varepsilon$ et $u = w$ aussi $u \overset{*}{\underset{R}{\hookrightarrow}} u = vw$.
$n \Longrightarrow n+1$: soit $u \left(\overset{v}{\underset{\widehat{R}.N^*}{\Longrightarrow}}\right)^{n+1} w$.
On distingue les deux cas complémentaires ci-dessous.
Cas 1 : $u \underset{\widehat{R}.N^*}{\longrightarrow} \left(\overset{v'}{\underset{\widehat{R}.N^*}{\Longrightarrow}}\right)^n w$ avec $av' = v$ et $a \in N$.

Aussi $u = au'$ et $u' \overset{v'}{\underset{\widehat{R}.N^*}{\Longrightarrow}} w$ d'où par récurrence $u' \overset{*}{\underset{R}{\hookrightarrow}} v'w$.

Aussi $u = au' \overset{*}{\underset{R}{\hookrightarrow}} av'w = vw$.

Cas 2 : $u \underset{\widehat{R}.N^*}{\overset{\varepsilon}{\longrightarrow}} \left(\overset{v}{\underset{\widehat{R}.N^*}{\Longrightarrow}}\right)^n w$.

Aussi $u = xz$ pour un $x \, R \, y$ et $yz \left(\overset{v}{\underset{\widehat{R}.N^*}{\Longrightarrow}}\right)^n w$.

Par hypothèse de récurrence, $yz \overset{*}{\underset{R}{\hookrightarrow}} vw$ d'où $u = xz \overset{*}{\underset{R}{\hookrightarrow}} vw$.
□

Par la proposition 18 et le lemme 5.3.1, nous obtenons le langage

$$\overset{*}{\underset{R}{\hookrightarrow}}(L) = L(\widehat{R}.N^*, L, \varepsilon)$$

des mots étiquetant les chemins dans le graphe $\widehat{R}.N^*$ entre les sommets de L et le sommet ε. C'est un langage algébrique lorsque L est algébrique et R est reconnaissable [Ca 90]. Lorsque R est fini et en prenant un nouveau symbole p représentant un *état de contrôle*, le système

$$p\widehat{R} = \{ pu \overset{\varepsilon}{\longrightarrow} pv \mid u \, R \, v \} \cup \{ pa \overset{a}{\longrightarrow} p \mid a \in N \}$$

peut être considéré comme un automate à pile avec l'alphabet de pile N. En effet, les règles ne sont pas sous la forme standard : tout membre gauche est un état suivi d'une seule lettre de pile. Cependant, la mise sous cette forme normale est standard par ajout de nouveaux états et de nouvelles règles. Ainsi, nous venons de décrire la construction effective d'un automate à pile reconnaissant le langage $\overset{*}{\underset{R}{\hookrightarrow}}(L)$ par pile vide et de pile initiale un mot de L.

5.3.3 Systèmes marqués

Maintenant, nous allons appliquer le théorème 14 à la dérivation de systèmes généralisant la dérivation bifixe. Nous considérons un ensemble fini M de symboles

spéciaux appelés *marques*. Les systèmes bifixes peuvent être facilement simulés et étendus en ajoutant des marques à la première et à la dernière position des membres des règles, tout en imposant l'identité de la première et/ou de la dernière marque de chaque membre [Al 08].

Définition. Etant donnés des ensembles disjoints M et N, un *système bifixe marqué* sur $M \cup N$ est un système

$$R \subseteq (M \cup \{\varepsilon\})N^*(M \cup \{\varepsilon\}) \times (M \cup N)^*$$

tel que pour toute règle $(u, v) \in R$,

$$\begin{array}{rcl} u(1) \in M & \vee & u(|u|) \in M \\ u(1) \in M & \Longrightarrow & u(1) = v(1) \\ u(|u|) \in M & \Longrightarrow & u(|u|) = v(|v|). \end{array}$$

Dans un tel système, toutes les marques sont conservées par récriture. Sans cette condition, on peut transformer tout système fini R sur N en le système marqué :

$$\begin{aligned} R_\bullet = \{ (\bullet u, \bullet v) \mid u\,R\,v \} \cup \{ (\bullet a, \#_a) \mid a \in N \} \cup \{ (\#_a, a\bullet) \mid a \in N \} \\ \cup \{ (a\bullet, \&_a) \mid a \in N \} \cup \{ (\&_a, \bullet a) \mid a \in N \} \end{aligned}$$

avec $M = \{\#_a \mid a \in N\} \cup \{\&_a \mid a \in N\} \cup \{\bullet\}$. Nous avons

$$u \xrightarrow[R]{*} v \iff \bullet u \xrightarrow[R_\bullet]{*} \bullet v \quad \text{pour tout} \quad u, v \in N^*.$$

Puisque $\xrightarrow[R]{*}$ n'est pas récursif en général, $\xrightarrow[R_\bullet]{*}$ ne l'est pas non plus. La rationalité de la dérivation bifixe peut être étendue à la dérivation des systèmes bifixes marqués.

Proposition 19 ([Al 08]) *La relation de dérivation de tout système bifixe marqué reconnaissable est rationnelle.*

Démonstration. On redonne la construction de [Al 08].
Soit R un système bifixe marqué reconnaissable.
Pour tout $\#, \& \in M$, nous considérons les relations binaires ${}_\#P$, $S_\&$, ${}_\#B_\&$ sur N^* définies comme suit :

$$\begin{array}{rcccl} \#.({}_\#P) & = & \xrightarrow[R_p]{*} \cap (\#N^* \times \#N^*) & = & \xrightarrow[R]{*} \cap (\#N^* \times \#N^*) \\ (S_\&).\& & = & \xrightarrow[R_s]{*} \cap (N^*\& \times N^*\&) & = & \xrightarrow[R]{*} \cap (N^*\& \times N^*\&) \\ \#.({}_\#B_\&).\& & = & \xrightarrow[R_p \cup R_s \cup R_b]{*} \cap (\#N^*\& \times \#N^*\&) & = & \xrightarrow[R]{*} \cap (\#N^*\& \times \#N^*\&). \end{array}$$

Par les propositions 16 et 17, ces relations sont rationnelles.
A chaque $\# \in M$ est associé deux symboles $0_\#$, $1_\#$ et nous construisons l'automate suivant :

$$\begin{array}{lcllcl} A & = & \{\ 0_\# \xrightarrow{(\#,\#)} 1_\# \mid \# \in M\ \} & \cup & \{\iota\, 0\,,\, o\, 1\} \cup \{0 \xrightarrow{\mathrm{Id}} 1\} \\ & \cup & \{\ 0 \xrightarrow{S_\#} 0_\# \mid \# \in M\ \} & \cup & \{\ 1_\# \xrightarrow{\#^P} 1 \mid \# \in M\ \} \\ & \cup & \{\ 1_\# \xrightarrow{\#B_\&} 0_\& \mid \#, \& \in M\ \}. & & \end{array}$$

Comme l'identité Id sur N^* est une relation reconnaissable (donc rationnelle), nous pouvons transformer A en un transducteur fini reconnaissant $\xrightarrow[R]{*}$.
□

Avant d'étendre cette classe de systèmes (bifixes marqués) en permettant l'ajout de marques et de règles infixes, nous considérons des systèmes dont l'ensemble des marques M est divisé en un sous-ensemble M_p de *marques préfixes* et un sous-ensemble M_s de *marques suffixes* : $M_p \cup M_s = M$ et $M_p \cap M_s = \emptyset$.

Définition. Un *système préfixe/suffixe avec ajout de marques* est un système

$$R \subseteq (M_p \cup \{\varepsilon\})N^*(M_s \cup \{\varepsilon\}) \times (M \cup N)^*$$

tel que pour toute règle $(u, v) \in R$,

$$\big(u(1) \in M_p \implies v(1) \in M_p\big) \text{ et } \big(u(|u|) \in M_s \implies v(|v|) \in M_s\big).$$

En prenant les marques $\#, \& \in M$ et une lettre $a \in N$, le système à deux règles

$$R = \{(\#a, \&), (\&, a\#)\}$$

ne peut pas être un système préfixe/suffixe avec ajout de marques puisque la marque $\#$ serait simultanément préfixe et suffixe. Pour ce système, les dérivations directe et inverse ne préservent pas la régularité :

$$\xrightarrow[R]{*}\big((\#a)^*\big) \cap a^*\#^* = \{\, a^n\#^n \mid n \geq 0\,\} \text{ et } \xrightarrow[R^{-1}]{*}\big((\#\mathrm{a})^*\big) \cap \#^*\mathrm{a}^* = \{\, \#^{\mathrm{n}}\mathrm{a}^{\mathrm{n}} \mid \mathrm{n} \geq 0\,\}.$$

On dit qu'un système R préfixe/suffixe avec ajout de marques est *algébrique*, si

$$R \cap N^* \times N^* \text{ est un systme algbriqueet } R - N^* \times N^* \text{ est reconnaissable.}$$

Appliquons le théorème 14.

Proposition 20 *Tout système algébrique préfixe/suffixe avec ajout de marques est alg/reg-préservant.*

Démonstration. Soit R un système algébrique préfixe/suffixe avec ajout de marques.

Les ensembles M_p et M_s sont respectivement des sous-alphabets préfixe et suffixe de R^{-1}.

Ainsi R^{-1} est M_pM_s-décomposable en $R^{-1} \cap N^*{\times}N^*$ et par la proposition 15, R est alg/reg-préservant.
□

Un cas particulier d'un système préfixe avec ajout de marques ($M_s = \emptyset$) est donné par un système reconnaissable $R \subseteq MN^*{\times}(MN^*)^+$ qui est alg/reg-préservant par la proposition 20, et également par la proposition 18 : la dérivation est égale à sa dérivation gauche-droite. Ces systèmes particuliers généralisent le premier modèle de réseaux dynamiques de systèmes à pile de [BMT 05].

Nous allons maintenant utiliser la proposition 20 pour obtenir les mêmes propriétés de fermeture pour l'extension suivante de systèmes bifixes marqués.

Définition. Un *système bifixe avec ajout de marques* est un système

$$R \subseteq (M \cup \{\varepsilon\})N^*(M \cup \{\varepsilon\}){\times}(M \cup N)^*$$

tel que pour toute règle $(u, v) \in R$,

$$\big(u(1) \in M \implies u(1) = v(1)\big) \text{ and } \big(u(|u|) \in M \implies u(|u|) = v(|v|)\big).$$

Notons que le système précédent $\{(\#a, \&), (\&, a\#)\}$ n'est pas un système bifixe avec ajout de marques. La proposition 20 reste valable pour de tels systèmes quand ils sont *algébriques*, à savoir si $R \cap N^*{\times}N^*$ est un système algébrique et $R - N^*{\times}N^*$ est reconnaissable.

Tout système bifixe avec ajout de marques R peut être décomposé en

$$R = R_p \cup R_s \cup R_b \cup R_i \cup R_d \cup R_0 \quad \text{avec}$$

$$
\begin{array}{lrcll}
R_p & \subseteq & MN^*\mathrm{x} & \times & M(M\cup N)^* & \textit{règles préfixes avec ajout de marques} \\
 & & & & & \#u \longrightarrow \#v \\
R_s & \subseteq & N^*M & \times & (M\cup N)^*M & \textit{règles suffixes avec ajout de marques} \\
 & & & & & u\& \longrightarrow v\& \\
R_b & \subseteq & MN^*M & \times & M(M\cup N)^*M & \textit{règles bifixes avec ajout de marques} \\
 & & & & & \#u\& \longrightarrow \#v\& \\
R_i & \subseteq & N^* & \times & (M\cup N)^*M(M\cup N)^* & \textit{règles infixes avec ajout de marques} \\
 & & & & & u \longrightarrow v\&w \\
R_d & \subseteq & MN^*M & \times & M & \textit{règles de suppression de marques} \\
 & & & & & \#u\# \longrightarrow \# \\
R_0 & \subseteq & N^* & \times & N^* & \textit{règles sans marque} \\
 & & & & & u \longrightarrow v
\end{array}
$$

où R_p et R_s peuvent avoir des règles communes : $R_p \cap R_s \;=\; R \cap M{\times}M(N^*M)^*$.

Théorème 21 *Tout système bifixe avec ajout de marques est alg/rég-préservant.*

Démonstration.
i) Tout d'abord, nous supposons que R n'a pas de règle de suppression de marques :

$$R_d \;=\; \emptyset.$$

Nous prenons une copie disjointe $\overline{M} \;=\; \{\, \overline{x} \mid x \in M \,\}$ de M, et nous définissons le morphisme $k : (M\cup N)^* \longrightarrow (M\cup\overline{M}\cup N)^*$ par

$$k(x) \;=\; \overline{x}x \;\text{ pour tout }\; x\in M \quad\text{et}\quad k(x)=x \;\text{ pour tout } x\in N.$$

Nous transformons R en le système préfixe/suffixe avec ajout de marques $\overline{R}$ sur $M_p = M$ et $M_s = \overline{M}$:

$$
\begin{array}{lrcl}
\overline{R} \;=\; & R_0 & \cup & \{\; (\#u \;,\; \#k(v)) \;\mid\; \#u\; R_p\; \#v \;\wedge\; \# \in M \;\} \\
 & & \cup & \{\; (u\overline{\&} \;,\; k(v)\overline{\&}) \;\mid\; u\&\; R_s\; v\& \;\wedge\; \& \in M \;\} \\
 & & \cup & \{\; (\#u\overline{\&} \;,\; \#k(v)\overline{\&}) \;\mid\; \#u\&\; R_b\; \#v\& \;\wedge\; \#,\& \in M \;\} \\
 & & \cup & \{\; (u \;,\; k(v)) \;\mid\; u\; R_i\; v \;\}.
\end{array}
$$

Pour tout $L \subseteq (M\cup N)^*$, nous avons

$$\xrightarrow[R]{*}(L) \;=\; \pi\big(\xrightarrow[\overline{R}]{*}(k(L))\big) \quad\text{et}\quad \xrightarrow[R^{-1}]{*}(L) \;=\; \pi\big(\xrightarrow[\overline{R}^{-1}]{*}(k(L))\big)$$

pour le morphisme $\pi : (M \cup \overline{M} \cup N)^* \longrightarrow (M \cup N)^*$ défini par

$$\pi(x) \;=\; \varepsilon \;\text{ pour tout }\; x \in \overline{M} \quad \text{et} \quad \pi(x) \;=\; x \;\text{ pour tout }\; x \in M \cup N.$$

Nous concluons en appliquant la proposition 20 à $\overline{R}$.

ii) Nous allons maintenant permettre des règles de suppression de marques :

$$R \;=\; R' \cup R_d \quad \text{avec} \quad R' \;=\; R_p \cup R_s \cup R_b \cup R_i \cup R_0\,.$$

Nous prenons la substitution h suivante :

$$\begin{array}{lcll} h(x) & = & x\,\{\, ux \,|\, xux \in \mathrm{Dom}(R_d)\,\}^* & \text{pour tout } \; x \in M \\ h(x) & = & x & \text{pour tout } \; x \in N. \end{array}$$

Comme R est reconnaissable, R_d est reconnaissable, donc h est une substitution régulière.

Pour tout $L \subseteq (M \cup N)^*$, nous avons

$$\xrightarrow[R_d]{*}(L) \;=\; h^{-1}(L) \quad \text{et} \quad \xrightarrow[R_d^{-1}]{*}(L) \;=\; h(L).$$

Donc $\xrightarrow[R_d]{*}$ et $\xrightarrow[R_d^{-1}]{*}$ préservent la régularité et l'algébricité.

Nous vérifions que

$$\xrightarrow[R_d]{} \circ \xrightarrow[R']{} \;\subseteq\; \xrightarrow[R']{} \circ \xrightarrow[R_d]{}.$$

Soit $v\#u\#w \xrightarrow[R_d]{} v\#w \xrightarrow[R']{} z$. Nous n'avons que les deux cas ci-dessous.

Cas 1 : $z = v'\#w$ pour $v\# \xrightarrow[R']{} v'\#$.

Alors $v\#u\#w \xrightarrow[R']{} v'\#u\#w \xrightarrow[R_d]{} v'\#w = z$.

Cas 2 : $z = v\#w'$ pour $\#w \xrightarrow[R']{} \#w'$.

Alors $v\#u\#w \xrightarrow[R']{} v\#u\#w' \xrightarrow[R_d]{} v\#w' = z$.

Ainsi

$$\xrightarrow[R]{*}(L) \;=\; \xrightarrow[R' \cup R_d]{*}(L) \;=\; \xrightarrow[R_d]{*}(\xrightarrow[R']{*}(L)) \quad \text{pour tout} \;\; L \subseteq (M \cup N)^*.$$

Par (i), $\xrightarrow[R]{*}$ préserve l'algébricité.

De plus $\xrightarrow[R^{-1}]{*}(L) \;=\; \xrightarrow[R'^{-1}]{*}(\xrightarrow[R_d^{-1}]{*}(L))$ et par (i), $\xrightarrow[R^{-1}]{*}$ préserve la régularité.

□

Notons que la proposition 20 et le théorème 21 peuvent être généralisés à la fois aux systèmes R tels que

$$R - N^* \times N^* \;\text{ est reconnaissable } \quad \text{et} \quad (R \cap N^* \times N^*) \cup D^{-1} \;\text{ est alg/reg} - \text{preservant.}$$

Le théorème 21 généralise le théorème 7 de [Al 08] : pour tout système R « marqué infixe avec des règles de suppression de marques », son inverse R^{-1} est un système particulier bifixe avec ajout de marques, donc $\xrightarrow[R]{*}$ préserve la régularité. Comme corollaire, le théorème 21 peut également répondre à une conjecture énoncée dans [Al 09] (page 99).

Dans ce chapitre, nous avons présenté un mécanisme de décomposition pour les systèmes de récriture de mots, ce qui nous a permis de transférer à plusieurs classes de systèmes de récriture de mots la préservation de la régularité et de l'algébricité inverse du système de Dyck.

6

Conclusion

Dans ce document de thèse, on a présenté plusieurs contributions dans le domaine des langages formels. Notre premier travail a été d'illustrer la pertinence des grammaires de graphes comme outil de démonstration pour la théorie des langages algébriques. Nous avons ainsi reformulé avec un point de vue géométriques les démonstrations du lemme des paires itérantes et du lemme de Parikh pour les langages algébriques. Ces démonstrations sont simples et ne font que traduire en terme de grammaires de graphes les démonstrations d'origine qui travaillent avec les arbres de dérivation. Cependant, elles constituent un premier pas vers une approche similaire d'autres résultats pour lesquels les arbres de dérivation ne suffisent plus. C'est le cas notamment pour l'extraction d'une forêt régulière couvrante qui nécessite de gérer les allers-retours sur les entrées des règles des grammaires de graphes. De même, le théorème de Chomsky-Schützenberger se traduit géométriquement comme la caractérisation des automates réguliers de degré fini (reconnaissant les langages algébriques) par les substitutions finies inverses de l'arbre binaire avec retours (reconnaissant, de la racine à elle-même, le langage de Dyck sur deux paires de parenthèses). Enfin, la fermeture des langages algébriques par intersection avec les langages réguliers résulte immédiatement de la fermeture des automates réguliers par produit de synchronisation avec les automates finis. On pourra aussi vérifier géométriquement la fermeture des langages algébriques par substitutions régulières inverses. Par conséquent, il me semble qu'il faudrait développer cette approche géométrique d'autant plus qu'elle pourrait s'étendre pour les familles des langages indexés d'ordre supérieur [Ma 74].

Un deuxième travail effectué dans cette thèse a consisté à étendre aux graphes réguliers des algorithmes de base sur les graphes finis, notamment pour calculer des problèmes de plus court chemin. Ces extensions ont été faites par calcul de plus petits points fixes sur les grammaires de graphes. Même si notre approche est générale et

s'appuie sur un étiquetage des graphes par les éléments d'un demi-anneau continu et idempotent quelconque, il me semble important de poursuivre l'extension de la théorie des graphes finis aux graphes réguliers. Cela a été fait (par Arnaud Carayol et Didier Caucal, mais non publié) pour l'extraction d'une forêt couvrante régulière d'un graphe régulier. Il faudrait continuer en étendant aux graphes réguliers par exemple les algorithmes de calcul de flots. De même, le coloriage des sommets des graphes finis avec quatre couleurs s'étend aisément aux graphes réguliers de degré borné. Une autre question très naturelle est celle de l'extension de ces résultats aux graphes de degré borné de la hiérarchie à pile [CW 03] ?

Enfin, un troisième travail porte sur l'étude de systèmes de récriture de mots dont la relation de dérivation préserve la régularité ou l'algébricité. Pour obtenir des familles générales de tels systèmes de récriture de mots, on a étendu le travail de [HW 04] pour décomposer la dérivation d'un système en une substitution régulière suivie de la dérivation d'un sous-système (auquel on adjoint le système de Dyck). L'itération de cette décomposition a permis d'étendre les travaux de [Al 08] aux systèmes bifixes marqués avec ajout de marques. Il serait opportun d'étendre cette décomposition de la dérivation aux systèmes de récriture sur les termes clos (sans variable libre) [DT 90].

Bibliographie

[AEI 01] L. Aceto, Z. Esik, A. Ingolfsdottir : *A fully equational proof of Parikh's theorem*, BRICS Report Series, 2001.

[Al 08] J. Altenbernd : *On bifix systems and generalizations*, 2^{nd} LATA (Martin-Vide, C., Otto, F., Fernau, H. eds.), LNCS 5196, pp. 40–51, Springer, 2008.

[Al 09] J. Altenbernd : *Reachability over word rewriting systems*, Ph.D. thesis, RWTH Aachen, Germany, 2009.

[Au 94] J.M. Autebert : *Théorie des langages et des automates*, Masson, 1994.

[Be 69] M. Benois : *Parties rationnelles du groupe libre*, C.R. Académie des Sciences, Série A 269, pp. 1188–1190, 1969.

[Be 79] J. Berstel : *Transductions and context-free languages*, Teubner-Verlag, 1979.

[BJW 82] R. Book, M. Jantzen, C. Wrathall : *Monadic thue systems*, Theoretical Computer Science 19, pp. 231–251, 1982.

[BO 93] R. Book, F. Otto : *String-rewriting systems*, Texts and Monographs in Computer Science, Springer-Verlag, 1993.

[BMT 05] A. Bouajjani, M. Müller-Olm, T. Touili : *Regular symbolic analysis of dynamic networks of pushdown systems*, 16^{th} CONCUR, LNCS 3653, pp. 473–487, Springer, 2005.

[Bu 64] R. Büchi : *Regular canonical systems*, Archiv für Mathematische Logik und Grundlagenforschung 6, pp. 91–111, 1964.

[BDLP 01] C.C. Bui, T.H. Dinh, B.L. Le, V.L. Pham : *Some properties of t-norms with threshold*, VJFUZZY'2001, pp. 28–33, 2001.

[BDTL 00] C.C. Bui, T.H. Dinh, D.H. Tran, T.Q. Le : *T-norms with threshold in fuzzy logic*, VietIT 2000, pp. 22–33, 2000.

[CW 03] A. Carayol, S. Wöhrle : *The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata*, in FSTTCS, LNCS 2914, pp. 112–123, 2003.

[Ca 90] D. Caucal : *On the regular structure of prefix rewriting*, 15^{th} CAAP, LNCS 431, pp. 87–102, 1990, [une version complète est dans Theoretical Computer Science 106, pp. 61–86, 1992].

[Ca 08] D. Caucal : *Deterministic graph grammars*, in Logic and Automata, Amsterdam University Press, J. Flum, E. Grädel, T. Wilke (Eds.), pp. 169–250, 2008.

[CD 07] D. Caucal, T.H. Dinh : *Path algorithms on regular graphs*, 16th FCT, LNCS 4639, pp. 199–212, 2007.

[CD 11] D. Caucal, T.H. Dinh : *Regularity and Context-Freeness over Word Rewriting Systems*, FOSSACS 2011, LNCS 6604, pp. 214–228, 2011.

[Ch 56] N. Chomsky : *Three models for the description of language*, IRE Transactions on Information Theory (2), pp. 113–124, 1956.

[Ch 59] N. Chomsky : *On certain formal properties of grammars*, Informations and Control, 2, pp. 137–167, 1959.

[Co 90] B. Courcelle : *Graph rewriting : An algebraic and logic approach*, in "Handbook of Theoretical Computer Science , Volume B", J. Van Leeuwen ed., pp. 193–242, Elsevier, 1990.

[DT 90] M. Dauchet, S. Tison : *The theory of ground rewrite systems is decidable*, Fifth Annual IEEE Symposium on Logic in Computer Science, pp. 242-248, 1990.

[De 11] M. Dehn *Uber unendliche diskontinuierliche gruppen*, Mathematische Annalen 71(1), pp. 116–144, 1911.

[DJ 90] N. Dershowitz, J.P. Jouannaud : *Rewrite systems*, Handbook of Theoretical Computer Science, Vol B, pp. 243–320, edited by J.Van. Leeuwen, Elsevier, 1990.

[Di 06] T.H. Dinh : *Le lemme de Parikh via les grammaires de graphes*, Addendum Contributions of the 4th IEEE International Conference on Computer Sciences - RIVF'06, pp. 13–18, Studia Informatica Universalis, 2006.

[EHW 06] J. Endrullis, D. Hofbauer, J. Waldmann : *Decomposing terminating rewrite relations*, 8^{th} WST, 39–43, Computing Research Repository, *http ://www.acm.org/corr/*, 2006.

[EKL 07] J. Esparza, S. Kiefer, M. Luttenberger : *On fixed point equations over commutative semirings*, 24^{th} STACS, LNCS 4393, pp. 296–307, 2007.

[GHW 04] A. Geser, D. Hofbauer, J. Waldmann : *Match-bounded string rewriting systems*, Applicable Algebra in Engineering, Communication and Computing 15, pp. 149–171, 2004.

[GS 64] S. Ginsburg, E. Spanier : *Bounded Algol like languages*, Trans. Amer. Math. Soc. 113, pp. 333–365, 1964.

[Ha 78] M. Harrison : *Introduction to formal language theory*, Addison - Wesley, 1978.

[Hi 74] T. Hibbard : *Context-limited grammars*, JACM 21(3), 446–453, 1974.

[HW 04] D. Hofbauer, J. Waldmann : *Deleting string rewriting systems preserve regularity*, Theoretical Computer Science 327, pp. 301–317, 2004 [originally published in DLT'03].

[HU 79] J. Hopcroft, J. Ullman : *Introduction to automata theory, languages, and compilation*, Addison - Wesley, 1979.

[HK 99] M. Hopkins, D. Kozen : *Parikh's theorem in commutative Kleene algebra*, Logic in Computer Science (LICS'99), IEEE Press, pp. 394–401, 1999.

[Im 88] N. Immerman, Nondeterministic space is closed under complementation. SIAM Journal on computing, 17(5), pp. 935–938, 1988.

[JKLP 87] M. Jantzen, M. Kudlek, K.J. Lange, H. Petersen : *$Dyck_1$-reductions of context-free languages*, 6^{th} FCT, LNCS 278, pp. 218–227, Springer, 1987.

[KKO 06] J. Karhumäki, M. Kunc, A. Okhotin : *Computing by commuting*, Theoretical Computer Science 356, pp. 200–211, 2006.

[Kl 56] S. Kleene : *Representation of Events in Nerve Nets and Finite Automata*, Automata Studies, pp. 3–42, Princeton, 1956.

[Kn 28] B. Knaster : *Un théorème sur les fonctions d'ensembles*, Annales de la Société Polonaise de Mathématique, v6, pp. 133–134, 1928.

[Ma 74] A. Maslov : *The hierarchy of indexed languages of arbitrary level*, Doklady Akademii Nauk SSSR 217, pp. 1013–1016, 1974.

[MS 85] D. Muller, P. Schupp : *The theory of ends, pushdown automata, and second-order logic*, TCS 37, pp. 51–75, 1985.

[Pa 66] R. Parikh : *On context-free languages*, J. Assoc. Comput. Mach., Vol. 13(4), pp. 570–581, 1966.

[Pi 73] D. Pilling : *Commutative regular equations and Parikh's theorem*, J. London. Math. Soc, 6(2), pp. 663–666, 1973.

[Re 88] C. Reutenauer : *Aspects mathématiques des réseaux de Pétri*, Masson, 1988.

[Ri 53] H. Rice *Classes of recursively enumerable sets and their decision problems*, Transactions of the American Mathematical Society 74, pp. 358–366, 1953.

[Sz 88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata, Acta Informatica, 26(3), pp. 279–284, 1988.

[Ta 55] A. Tarski : *A lattice-theoretical fixpoint theorem and its applications*, Pac. J. Math. 5, pp. 285–309, 1955.

[Tho 09] W. Thomas : *The reachability problem over infinite graphs*, 4th CSR 2009, LNCS 5675, pp. 12–18. Springer, 2009.

[Tho 06] W. Thomas : *Automata theory and infinite transition systems*. In Preproceedings of EMS Summer School CANT 2006, Prépublication 06.002, Institut de Mathématique, Université de Liège, 2006.

[Thu 14] A. Thue *Probleme über Veränderungen von Zeichenreihen nach gegeben Regeln*, Skr. Vid. Kristianaia. I Mat. Naturv. Klasse 1914.

[Tu 36] A. Turing : *On computable numbers, with an application to the Entscheidungs* Proceedings of the London Mathematical Society, Ser. 2, Vol. 42, pp. 230–265, 1936. [A correction, vol. 43, pp. 544–546, 1937.]

Table des figures