$\mathcal{A}$ ma grand-mère, $\mathcal{I}$sabelle

Remerciements

Je remercie chaleureusement Jacques Sakarovitch pour avoir suivi mes travaux depuis le DEA. Il a su m'aiguiller et m'aiguillonner à tout moment, sans compter le temps qu'il me consacrait.

Je remercie Volker Diekert, Stéphane Gaubert et Pacal Weil d'avoir accepté d'être rapporteurs pour mon mémoire de thèse, ainsi que Jean Berstel, Gérard Cohen, Jean-Eric Pin et Wolfgang Thomas qui ont accepté de faire partie de mon jury de thèse.

Je remercie le personnel de l'ENST pour avoir permis que cette thèse se déroule dans les meilleures conditions, plus particulièrement le département INFRES et les « habitants » du couloir E7. Un grand merci à Chan, Raphaël et Yann qui furent des « co-bureaux » en or.

Je remercie les chercheurs du LIAFA dont l'accueil fut chaleureux chaque fois que je vins hanter leurs locaux, en particulier les ex-doctorants, Christophe, Cyril, Ines et Pierre-Cyrille pour les échanges fructueux (au moins pour moi) que nous avons pu avoir, sur la théorie des automates et d'autres sujets (presque) aussi passionnants.

Je remercie ceux qui m'ont permis de franchir les étapes qui me conduisent aujourd'hui à rédiger ce mémoire, mes professeurs, de l'école maternelle à la préparation de cette thèse, tout particulièrement Jacques Noailles, qui sait communiquer son amour de la belle mathématique à ses élèves.

Je tiens à remercier tous ceux dont la présence a rendu ces années à Paris plus qu'agréables, Alain, Alexandre, Axel, Dominique, Emmanuel, Fabienne, Isabelle, Lætitia, Nicolas, Octave, Thomas et les nombreux autres que j'oublie.

Enfin, tout ceci n'aurait pas été possible sans une arrière-garde solide, au premier rang de laquelle se trouve ma famille, ainsi que Franz, Jean-Baptiste, Jean-François, Paul, Sophie, Stéphane, … Je les remercie du fond du cœur.

— ∘ —

Je remercie aussi Timothy Van Zandt pour PSTricks, et surtout Jacques Sakarovitch qui, à partir de celui-ci, a élaboré VAUCANSON avec lequel la totalité des automates représentés dans ce mémoire a été dessinée.

Table des matières

— ∘ —

Introduction

Cette thèse est consacrée à l'étude de quelques propriétés « structurelles » des langages et séries rationnelles. La structure des objets que nous traiterons intervient en fait à deux niveaux. D'une part, les différents résultats que nous présentons ici sont mis en relation avec la structure des langages. Par exemple, nous ne nous contenterons pas de dire que la hauteur d'étoile d'un langage réversible est calculable, mais qu'on peut trouver une expression rationnelle issue d'un automate réversible qui réalise cette hauteur d'étoile. De même, lorsque nous parlerons du passage des expressions rationnelles aux automates, nous expliquerons en quoi les objets syntaxiques manipulés reflètent la structure des séries rationnelles sous-jacentes. D'autre part, les méthodes employées dans les preuves se basent sur la structure des automates qui représentent les langages ou séries dont nous montrons les propriétés. Ainsi, nous recourrons aux morphismes entre automates ou à l'étude des circuits du graphe sous-jacent. Ces outils permettent d'apporter un éclairage différent et parfois fructueux aux résultats issus de l'étude des monoïdes syntaxiques ou de la théorie spectrale appliquée aux matrices de transition des automates.

La quasi-totalité des résultats de la théorie des automates correspondent à des algorithmes. Nous avons essayé de mieux tirer partie des propriétés des objets auxquels s'appliquent ces résultats, pour obtenir des énoncés plus précis, et qui donnent des algorithmes qui correspondent mieux à la difficulté « intrinsèque » du problème.

On trouvera dans ce mémoire, une méthode de construction d'un automate réversible pour un langage réversible donné, ainsi que le calcul effectif de la hauteur d'étoile d'un langage réversible. Ces deux résultats sont obtenus *via* l'étude de l'automate universel dont nous donnerons une construction.

Par ailleurs, on donne pour les séries $(\mathsf{max},+)$ sur un alphabet unaire, un algorithme qui permet de décider de la séquentialité ainsi qu'un algorithme de déterminisation. Ce mémoire se conclut par l'exposé d'une méthode permettant le calcul d'un automate réalisant la série rationnelle donnée par une expression avec multiplicités dans un semi-anneau quelconque.

— ◦ —

Le premier chapitre d'une thèse consacrée aux langages et séries rationnels se doit de contenir un certain nombre de résultats maintes fois énoncés, même si une compréhension différente ou, en étant plus pragmatique, un objectif précis à atteindre conduisent à les formuler de façon plus ou moins originale. Celui-ci ne dérogera pas à la règle.

Nous y présentons tour à tour les graphes orientés, les structures algébriques que nous utilisons puis les différentes formes d'automates que nous rencontrerons. Les objets sont présentés de la façon la plus générale possible lorsque cela ne soulève pas de problème particulier. Par exemple, nous ne parlons pas des séries sur un monoïde quelconque pour éviter de traiter le cas des séries non sommables dont le produit peut ne pas être défini et que nous ne rencontrerons pas dans ce mémoire.

Les principaux points abordés dans ce chapitre sont, au regard de l'importance qu'ils prennent dans la suite de ce mémoire, et dans l'ordre où ils apparaissent, la notion de morphisme de graphes (et plus loin, d'automates), d'action d'un monoïde sur un ensemble, les problèmes liés à la factorisation ou à la définition de l'étoile dans les semi-anneaux, la notion de langage reconnaissable et le lien qu'on peut établir entre les *futurs* des états d'un automate (avec multiplicité ou non) et les quotients du langage ou de la série reconnus par cet automate. Le dernier point, qui sera mis à profit dans le dernier chapitre, rappelle que les quotients d'une série rationnelle appartiennent à un semi-module finiment engendré et que, plus particulièrement, les quotients d'une série séquentielle appartiennent à un faisceau de droites finiment engendré.

Le second chapitre est dévolu à la présentation de l'automate *universel*. Cet objet a été introduit par J.H. Conway [19] sous le nom de *matrice des facteurs*. O. Carton [12] a montré qu'on pouvait définir cette matrice (qui apparaît naturellement comme un automate) pour n'importe quel monoïde. Là encore, nous présentons cet objet dans le cadre le plus général possible. La première partie de ce chapitre est une présentation de définitions et de résultats qui, même s'ils sont déjà apparus dans d'autres travaux dont certains ont presque trente ans, ne peuvent pas être considérés comme faisant partie du corpus commun de la théorie des automates.

La notion de base de ce chapitre est celle de *factorisations* d'un sous-ensemble d'un monoïde. Celles-ci permettent de construire l'automate universel de ce sous-ensemble. Il s'agit du plus petit automate qui reconnaît le langage dans lequel tout automate équivalent a une image par morphisme. Nous verrons que l'on peut définir cet automate pour n'importe quel sous-ensemble de n'importe quel monoïde, mais qu'il ne s'agit d'un automate fini que si l'ensemble est reconnaissable; de plus, nous ne l'utiliserons que pour les langages rationnels du monoïde libre.

Deux éléments inédits apparaissent dans cette partie. D'une part, on définit l'automate universel *écorché* dans lequel les propriétés du langage sont plus visibles, au sens propre comme au figuré. L'écorché est obtenu à partir de l'automate universel en supprimant un grand nombre de transitions et en ajoutant des transitions spontanées. On montre que l'automate universel correspond exactement à la clôture de son écorché selon les transitions spontanées. D'autre part, on donne un moyen effectif pour construire, à partir de l'automate minimal d'un langage, l'automate universel correspondant sans calculer le monoïde syntaxique. La dernière partie du chapitre, consacrée à ce que j'appelle le *développé* d'un automate, est plus une façon de voir l'automate universel, comme issu d'un automate particulier qui reconnaît le langage, qu'une méthode de construction. Il se peut en effet que ce ne soit pas l'automate minimal qui reflète le mieux les propriétés du langage. Il est alors plus pertinent de voir l'automate universel comme issu de l'automate *ad hoc*.

Cette construction nous servira particulièrement dans le second chapitre où, pour étudier les langages réversibles, nous considérerons l'automate universel d'un tel langage comme issu d'un automate réversible. Même si on ne connaît pas ce dernier, il nous permettra de mettre à jour des propriétés du premier.

Le second chapitre est en fait le socle sur lequel sont bâtis les troisième et quatrième chapitres qui sont consacrés à l'étude de certains problèmes liés aux langages *réversibles*.

Le troisième chapitre débute donc par une présentation de ces langages qui apparaissent naturellement dans l'étude de problèmes aussi variés que l'intelligence artificielle, les codes bipréfixes, les semi-groupes inversifs ou les automates quantiques ... Après avoir rappelé la définition d'un langage réversible, nous rappelons que la propriété de réversibilité d'un automate peut se faire aux dépens du nombre d'état (résultat de P.-C. Héam [35]), du déterminisme ([36]) et même de la non-ambiguïté. Un automate minimal peut donc ne pas refléter la réversibilité du langage qu'il reconnaît. Ce fait rend la manipulation de cette classe de langages délicate. Bien que J.-E. Pin [51] ait montré qu'on peut décider de l'appartenance d'un langage à cette classe, l'algorithme induit par sa preuve pour construire un automate réversible pour un langage réversible donné conduirait à un automate beaucoup plus gros que ce qu'on peut raisonnablement espérer. Nous nous proposons ici de donner une construction d'un automate réversible.

Le principal résultat de ce troisième chapitre est en effet l'existence, dans l'automate universel, d'un sous-automate *quasi-réversible* équivalent. Cette classe d'automates apparaît comme plus compacte et semble être la bonne notion pour une représentation canonique des langages réversibles. Le passage d'un automate quasi-réversible à un automate réversible est de plus très simple (même s'il peut s'avérer coûteux).

Avant d'en arriver là, on étudie une catégorie particulière de langages réversibles, la classe des langages à *groupe*. Les problèmes qu'on rencontre pour ces langages sont beaucoup plus simples, car, en particulier, l'automate minimal d'un langage à groupe est un automate à groupe. Entamer notre étude par ces langages-ci permet donc de mieux échelonner les difficultés et d'introduire les outils et méthodes que nous emploierons pour l'étude des langages réversibles.

Nous montrons que l'automate universel d'un langage à groupe peut être décomposé en *étages* et que chaque composante fortement connexe est un *automate à groupe*. Nous abordons ensuite les langages réversibles et montrons que ces propriétés sont conservées : on peut décomposer l'automate universel en étages et chaque composante fortement connexe est réversible. Ceci nous permet de montrer ensuite que l'image d'un certain automate réversible dans l'automate universel est un automate quasi-réversible.

Le quatrième chapitre est consacré au problème de la *hauteur d'étoile*. Ce problème, posé par L. C. Eggan [22] en 1963 a été résolu par K. Hashiguchi [33] en 1988. Il demeure que le résultat d'Hashiguchi est avant tout un résultat d'« existence », tant par la complexité de l'algorithme mis en œuvre que par le manque d'informations obtenues sur la forme du résultat.

Le principal résultat de ce chapitre comporte en effet deux volets : d'une part la hauteur d'étoile d'un langage réversible est effectivement calculable, d'autre part, on peut calculer un automate *réversible* qui reconnaît le langage avec un *enlacement* qui correspond à la hauteur de celui-ci. Ce résultat est le fruit d'une collaboration avec J. Sakarovitch [43].

Jusqu'à présent, les études similaires étaient basées sur la connaissance de l'automate minimal qui se devait de refléter les propriétés désirées du langage. C'est pourquoi les résultats précédents se limitent aux langages à groupe (R. McNaughton [47]) ou aux langages dont l'automate minimal est réversible (R. Cohen [17] ou K. Hashiguchi et N. Honda [31]).

Le chapitre débute par un rappel du théorème d'Eggan qui lie la hauteur d'étoile d'un langage aux *enlacements* des automates qui le reconnaissent. Ce sera en effet sur cette notion que nous travaillerons, en établissant l'existence de morphismes qui préservent l'enlacement entre des automates d'enlacement minimum et l'automate universel du langage.

Les langages à groupe, pour lesquels les résultats sont en fait une réexpression de résultats anciens, sont utilisés comme introduction au cas plus général des langages réversibles. La preuve pour les langages à groupe reprend celle donnée par R. McNaughton [47], réexprimée dans le cadre des automates universels, et présentée de manière à ce que la preuve pour les langages réversibles suive le même schéma.

Cette preuve reflète le changement de méthode dans l'approche du problème. R. McNaughton montre, qu'à partir de l'automate minimal, on peut construire un ensemble d'automates parmi lesquels, on peut en trouver dont l'union reconnaît le langage avec un enlacement minimum. A l'opposé, on montre que dans l'automate universel, on peut trouver des sous-automates dont l'union reconnaît le langage avec un enlacement minimum. Même si, au final, les automates construits par R. McNaughton et ceux qu'on trouve dans l'automate universel sont les mêmes, c'est ce renversement de point de vue qui nous permet ensuite de traiter des langages réversibles, pour lesquels l'automate minimal ne peut plus servir de référence.

Les deux derniers chapitres ne dépendent pas des précédents.

Le cinquième chapitre est centré sur l'étude des automates « $(\mathsf{max},+)$ ». L'algèbre $(\mathsf{max},+)$ permet de résoudre des problèmes de contrôle optimal ou d'études de comportement asymptotiques de systèmes physiques ou informatiques. Les automates $(\mathsf{max},+)$ interviennent dans de nombreux domaines. Ils ont été, entre autre, utilisés pour décider le problème de limitation d'une série [32, 58] qui est en soit un résultat important, puisqu'il donne la décision du problème de la puissance finie (savoir si l'étoile d'un langage rationnelle est une union finie de puissances de ce langage), et qui, de plus, est utilisé dans la preuve générale de la décision de la hauteur d'étoile. Nous les présentons de manière à ce que les résultats énoncés restent valables pour tout semi-anneau $(\mathsf{max},+)$ ou $(\mathsf{min},+)$ raisonnable.

Nous montrons dans ce chapitre qu'on peut décider de la séquentialité d'une série sur un alphabet unaire et nous donnons un algorithme de déterminisation. L'algorithme de décision peut être vu comme une variation sur certains résultats de la théorie spectrale des matrices à coefficients dans un semi-anneau $(\mathsf{max},+)$.

Ce résultat se trouve à la croisée de deux problèmes déjà résolus. D'une part, sur un alphabet à plusieurs lettres, si l'automate $(\mathsf{max},+)$ qui réalise la série est *univoque*, c'est-à-dire que tout calcul étiqueté par un mot donné a la même valeur, on peut appliquer les techniques de décisions développées par C. Choffrut [14] pour les transducteurs. Toutefois, il faut souligner que toute série rationnelle n'est pas réalisable par un automate univoque et que, dans le cas général, le problème de la déterminisation d'un automate $(\mathsf{max},+)$ est fondamentalement différent de celui des transducteurs fonctionnels.

D'autre part, S. Gaubert [24] a montré que toute série $(\mathsf{max},+)$-rationnelle sur un alphabet à une seule lettre peut être réalisée par un automate non ambigu. La preuve peut être transformée en un algorithme qui produit un automate non ambigu à partir d'une expression ou d'un automate quelconque. Nous montrons toutefois qu'on n'a pas besoin de passer par l'intermédiaire d'une représentation non ambiguë, dont la taille peut être arbitrairement grande devant celle de l'automate de départ, pour décider de la séquentialité dans le cas unaire et que notre algorithme de détermination donne en outre, une autre méthode pour obtenir un automate non ambigu réalisant n'importe quelle série rationnelle sur un alphabet unaire donnée.

L'étude des liens entre la hauteur d'étoile et l'enlacement fait apparaître, dans le quatrième chapitre, des rapports étroits entre un automate et l'expression rationnelle qu'il peut donner. Le sixième chapitre traite du passage effectif inverse d'une expression avec multiplicités à un automate. Ce problème a été résolu de diverses façons pour les automates sans multiplicités et c'est une de ces méthodes, qui rend compte de la propriété de rationalité des séries représentées, que nous nous sommes proposé de généraliser.

Nous présentons un algorithme qui permet, à partir d'une expression rationnelle avec multiplicités dans un semi-anneau quelconque, de construire un automate équivalent dont le nombre d'états, hormis l'état initial, est inférieur ou égal au nombre de lettres qui apparaissent dans cette expression. Ce résultat est l'exacte généralisation du résultat d'Antimirov [3] sur les expressions rationnelles sans multiplicités. Nous examinons aussi quels sont les amendements qui peuvent être apportés à cet algorithme selon la forme souhaitée du résultat ou l'hypothèse de commutativité du semi-anneau.

La construction de l'automate correspond au calcul d'un ensemble d'expressions qui représentent des séries qui engendrent le semi-module auquel appartiennent les quotients de la série représentée par l'expression de départ. Il ne s'agit pas ici de redémontrer le fait que les quotients d'une série rationnelle appartiennent à un semi-module fini, mais plutôt d'obtenir explicitement une représentation d'un ensemble (fini) de générateurs de ce semi-module.

— ∘ —

Notice

Les notations et conventions adoptées dans ce mémoire sont précisées dans le premier chapitre. Il paraît difficile de lire les chapitres 3 ou 4 sans connaître ce qui est exposé dans le chapitre 2. En revanche, même s'il est fait appel à certains résultats du chapitre 3, les notions développées dans le chapitre 4 sont indépendantes, et on peut envisager de lire l'un sans avoir pris connaissance de l'autre. Les deux derniers chapitres sont totalement indépendants, sauf, évidemment, du premier chapitre.

La numérotation des différents énoncés est classique. Ils sont indexés par le numéro du chapitre courant, suivi de celui de l'énoncé dans le chapitre. Les propositions, corollaires et lemmes sont numérotés ensemble (le lemme 4.6 est par exemple suivi du corollaire 4.7 et de la proposition 4.8). Les définitions et théorèmes sont numérotés à part.

Les exemples sont indexés indépendamment des chapitres. Quelques-uns traversent en effet presque tous les chapitres de ce mémoire. Afin d'éviter de répéter à chaque occurence le langage ou son automate minimal, j'ai choisi de donner un numéro à chacun d'entre eux. On trouvera dans la table des exemples le lieu de leurs différentes apparitions. On pourra ainsi repérer quels sont mes préférés.

— ◦ —

Chapitre 1

Notions fondamentales

Une tourniquette
Pour fair' la vinaigrette
Un bel aérateur
Pour bouffer les odeurs
Des draps qui chauffent
Un pistolet à gaufres
Un avion pour deux
Et nous serons heureux

B. Vian, *La complainte du progrès*

1 Notations et généralités

Les notions introduites dans ce chapitre ne sont que des rappels. On peut donc sans dommage passer au chapitre suivant. Toutefois, avant ces diverses définitions, voici un certain nombre de conventions typographiques que j'ai essayé de respecter dans ce rapport.

Un ensemble est désigné par une lettre majuscule, ainsi qu'un monoïde, dont la multiplication (interne) est notée par un point. Un semi-anneau est noté par une majuscule ajourée ($\mathbb{K}$), son addition et sa multiplication sont notées respectivement $\oplus$ et $\otimes$, sauf si l'instance de semi-anneau qu'on considère a des lois usuellement désignées par d'autres symboles. Les éléments de ces structures sont notés par des minuscules.

L'action (à droite) d'un élément x d'un monoïde sur un élément p d'un ensemble est notée $p \cdot x$. De même, l'action à gauche est notée $x \cdot p$; le contexte permet de différencier ces deux opérations.

Les fonctions sont désignées par des minuscules grecques.

Les automates et les graphes sont notés par des majuscules « calligraphiées ». La majuscule $\mathcal{L}$ désigne un langage. Les expressions rationnelles sont désignées par des majuscules « droites ».

Si X est un ensemble, on note $\mathcal{P}(X)$ l'ensemble des parties de X :

$$\mathcal{P}(X) = \{Y \mid Y \subseteq X\}.$$

Une **relation** α sur $X \times Y$ est un sous-ensemble de $X \times Y$. On note $x\alpha = \{y \mid (x,y) \in \alpha\}$. Une relation est une **fonction** si pour tout x dans X, il existe au plus un élément y dans Y tel que (x,y) appartient à α; l'élément y est alors appelé image de x par α et on note $y = x\alpha$[1]. Toute relation de X dans Y peut être vue comme une fonction de X dans $\mathcal{P}(Y)$.

On note Y^X l'ensemble des fonctions de X dans Y. Si X est un ensemble fini, une telle fonction peut être vue comme un vecteur. Le **support** d'une fonction α de Y^X, noté $\mathsf{Supp}(\alpha)$, est l'ensemble des éléments de X qui ont une image par α. Une **application** est une fonction dont le support est X.

Dans certains cas, il est plus commode de travailler avec des applications. On adjoint alors à Y un zéro (0_Y) et, à toute fonction α de Y^X, on associe l'application α' de $(Y \cup \{0_Y\})^X$ qui est égale à α sur le support de α et dont l'image est 0_Y ailleurs. Dans ce contexte, le support de α' est l'ensemble des éléments de X dont l'image est différente de 0_Y par α'.

— ◦ —

2 Graphes orientés

Un automate est avant tout un graphe orienté. Nous allons donc présenter ces objets.

DÉFINITION 1.1 *Un **graphe orienté** est un ensemble de sommets (Q) reliés par un ensemble de flèches ou arcs (orientés). Chaque arc est désigné par un couple formé de son sommet de départ et de celui d'arrivée. Formellement un graphe orienté $\mathcal{G}$ est donc un couple $\langle Q,E \rangle$ tel que E est un sous-ensemble de $Q \times Q$. Le graphe $\mathcal{G}$ est fini si Q est un ensemble fini.*

DÉFINITION 1.2 *Soit $\mathcal{G} = \langle Q,E \rangle$ un graphe orienté. Un **sous-graphe** de $\mathcal{G}$ est un graphe $\mathcal{G}' = \langle Q',E' \rangle$ tel que $Q' \subseteq Q$ et $E' \subseteq E \cap Q' \times Q'$.*

DÉFINITION 1.3 *Soit $\mathcal{G} = \langle Q,E \rangle$ un graphe orienté. Soit n un entier positif. Un **chemin** de $\mathcal{G}$ de longueur n est un (n+1)-uplet de sommets $(p_0,p_1,\ldots,p_n)$ tel que, pour tout i dans $[1;n]$, (p_{i-1},p_i) est un arc de $\mathcal{G}$. Le sommet p_0 est le **sommet de départ** du chemin, p_n est le **sommet d'arrivée**. Un chemin est un circuit (ou une **boucle**) s'il est de longueur non nulle et que $p_0 = p_n$. C'est un **circuit élémentaire** si, pour tout $i < j$ dans $[0;n]$, $p_i = p_j$ entraîne $i = 0$ et $j = n$.*

1. On note la fonction *à droite* de l'élément sur lequel elle s'applique (par exemple $x\alpha$ est l'image de x par la fonction α). Ceci permet (entre autre) de noter $\alpha\beta$ la fonction qui consiste à appliquer d'abord α puis β.

DÉFINITION 1.4 *Un graphe orienté est un* ***graphe acyclique*** *s'il ne contient aucune boucle. Un graphe orienté est un* ***graphe fortement connexe*** *si, pour tout couple de sommets (p,q), il existe un chemin de p à q. Une* ***composante fortement connexe*** *d'un graphe orienté est un sous-graphe fortement connexe maximal. Une* ***pelote*** *est une composante fortement connexe non triviale, c'est-à-dire contenant au moins un arc.*

DÉFINITION 1.5 *Soit $\mathcal{G} = \langle\, Q,E \,\rangle$ un graphe orienté. Soit $\overline{E} = \{(p,q) \mid (q,p) \in E\}$. Le graphe $\mathcal{G}$ est* ***connexe*** *si le graphe $\langle\, Q,E \cup \overline{E} \,\rangle$ est fortement connexe. Une* ***composante connexe*** *d'un graphe orienté est un sous-graphe connexe maximal.*

EXEMPLE 1.1 Le graphe présenté figure 1 contient trois composantes fortement connexes : $\langle\, \{p\},\{(p,p)\} \,\rangle$, $\langle\, \{q\},\emptyset \,\rangle$ et $\langle\, \{r\},\{(r,r)\} \,\rangle$; la première et la troisième sont des pelotes. D'autre part, le graphe, bien qu'il ne soit pas fortement connexe, est connexe.

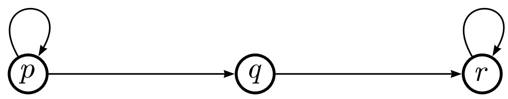


FIG. 1 – *Un graphe orienté à trois sommets*

DÉFINITION 1.6 *Un graphe $\mathcal{G} = \langle\, Q,E \,\rangle$ est* ***étiqueté*** *par un ensemble X s'il existe une application de E dans X. L'image d'un arc e par cette application est appelée* ***étiquette*** *de E. On notera alors $\mathcal{G} = \langle\, Q,E' \,\rangle$, avec $E' = \{(p,x,q) \mid x \in X$ étiquette de $(p,q) \in E\}$.*

DÉFINITION 1.7 *Soit $\mathcal{G} = \langle\, Q,E \,\rangle$ et $\mathcal{H} = \langle\, R,F \,\rangle$ deux graphes. Une application μ de Q dans R est un* ***morphisme de graphes*** *de $\mathcal{G}$ dans $\mathcal{H}$ si, quel que soit (p,q) dans E, $(p\mu,q\mu)$ appartient à F. Par extension, on dit que $(p\mu,q\mu)$ est l'image de (p,q) par μ. Si $\mathcal{G}$ et $\mathcal{H}$ sont des graphes étiquetés, l'application μ est un morphisme de $\mathcal{G}$ dans $\mathcal{H}$ si, quel que soit (p,x,q) dans E, $(p\mu,x,q\mu)$ appartient à F.*

— ∘ —

3 Structures algébriques

3.1 Monoïdes

DÉFINITION 1.8 *Un ensemble X, muni d'une loi associative et qui contient un élément neutre noté 1_X est un* ***monoïde***.[2] *On appelle généralement cette loi la multiplication du monoïde. Lorsqu'on veut préciser que X est muni de la loi « . », on désigne le monoïde par $(X,.)$.*

2. Une telle structure sans élément neutre est appelée semi-groupe.

Évidemment, les axiomes de monoïde sont beaucoup plus faibles que ceux de groupe. Toutefois, l'associativité est une contrainte forte qui conduit les monoïdes à avoir des structures très particulières (voir [40, 52]).

Un monoïde peut contenir un élément absorbant. On appelle alors celui-ci le zéro du monoïde.

Dans la plupart des cas, les ensembles classiques munis d'une loi vérifient les axiomes de monoïde. Ainsi, $(\mathbb{N},+)$, $(\mathbb{N},*)$, $(\mathbb{Z},*)$, $(\mathcal{M}_n(\mathbb{R}),*)$, *etc.* sont des monoïdes. Les groupes sont eux aussi des monoïdes.

Définition 1.9 *Soit $(M,.)$ un monoïde. Pour tout couple (x,y) d'éléments de M, le* ***quotient à gauche*** *de x par y noté $y^{-1}x$ est l'ensemble $\{z \in M \mid y.z = x\}$. Le quotient à gauche d'un sous-ensemble de M par y est l'union des quotients des éléments du sous-ensemble par y.*

Définition 1.10 *Soit M un monoïde et A un ensemble de générateurs de M. Le* ***graphe de Cayley*** *(droit) de M par rapport à A est le graphe étiqueté $\langle M,E \rangle$, où*

$$E = \{(x,a,y) \in M \times A \times M \mid y = x.a\}.$$

Exemple 1.2 Dans la suite, on utilisera beaucoup les monoïdes finis. Soit M_1 le monoïde donné par la table ci-dessous. On note 1 pour 1_{M_1} et 0 pour 0_{M_1}.

.	1	x	y	t	0
1	1	x	y	t	0
x	x	x	0	0	0
y	y	t	y	t	0
t	t	t	0	0	0
0	0	0	0	0	0

Quelques quotients à gauche de ce monoïde :

$$y^{-1}t = \{x,t\}, \qquad x^{-1}0_{M_1} = \{y,t,0_{M_1}\}, \qquad \text{et} \qquad x^{-1}t = \emptyset.$$

Les éléments x et y engendrent le monoïde M_1. Le graphe de Cayley de M_1 par rapport à ces éléments est représenté figure 2.

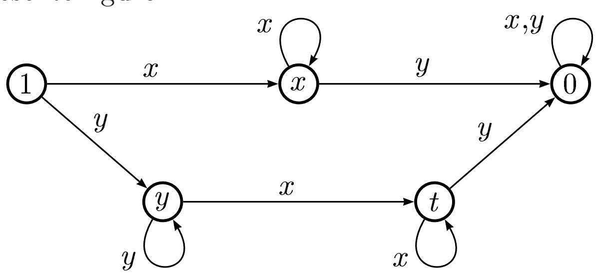


Fig. 2 – *Graphe de Cayley de M_1.*

Définition 1.11 *Soit $\sim$ une équivalence sur un monoïde M. Cette équivalence est **régulière à droite** (on dit aussi que c'est une congruence droite) si*

$$\forall x,y \in M \ x \sim y \Rightarrow \forall z \ x.z \sim y.z.$$

De même, c'est une congruence gauche, ou une relation d'équivalence régulière à gauche si

$$\forall x,y \in M \ x \sim y \Rightarrow \forall z \ z.x \sim z.y.$$

Si une relation d'équivalence est régulière à gauche et à droite, c'est une congruence. Les classes de M modulo cette équivalence forment alors un monoïde.

Définition 1.12 *Soit X un ensemble et M un monoïde. Une application μ de M dans X^X est une **action** à droite de M sur X si :*

$$\begin{aligned} \forall p \in X, \forall x,y \in M, \qquad p(1_M\mu) &= p, \\ (p(x\mu))(y\mu) &= p((x.y)\mu). \end{aligned}$$

On note alors $p \cdot x = p(x\mu)$. Symétriquement, une telle application est une action à gauche de M sur X si :

$$\begin{aligned} \forall p \in X, \forall x,y \in M, \qquad p(1_M\mu) &= p, \\ (p(x\mu))(y\mu) &= p((y.x)\mu). \end{aligned}$$

On note alors $x \cdot p = p(x\mu)$.

— ∘ —

Une famille de monoïdes tient une place particulière parmi les monoïdes; ce sont les monoïdes libres, et, en ce qui nous concerne, les monoïdes libres finiment engendrés.

Définition 1.13 *On désigne par **alphabet** un ensemble fini A non vide de symboles appelées **lettres**. On peut former des **mots** par concaténation de ces lettres. L'opération de concaténation, associative, fait de l'ensemble des mots sur A, noté A^*, un monoïde : le **monoïde libre engendré par** A. Le mot vide, élément neutre de ce monoïde est noté 1_{A^*}. On notera A^+ l'ensemble des mots de A^* différents du mot vide. Un sous-ensemble de A^* est appelé un **langage**. La **longueur d'un mot** est le nombre de lettres qu'il comporte; on note la longueur d'un mot u par $|u|$, de même, pour toute lettre a de A, on note le nombre d'occurrences de a dans u, $|u|_a$. On notera par ailleurs u_i la i-ème lettre du mot u.*

La métaphore linguistique qui guide cette définition ne doit pas induire en erreur. Il n'est *a priori* nullement question de sémantique dans la définition des mots. Ainsi, sur l'alphabet latin des majuscules (non accentuées) qui compte vingt-six éléments, $JRASKDFW$ est un mot, bien qu'on puisse douter qu'il ait un sens dans une des langues utilisant cet alphabet.

La taille des alphabets qu'on considère peut varier selon les emplois. Ainsi, traditionnellement, en informatique, l'alphabet a deux lettres, 0 et 1 (nous utiliserons plutôt a et b

pour ne pas confondre avec d'autres usages de 0 et 1). En bio-informatique, il compte généralement 4 lettres (les bases du code génétique) ou 20 (les acides aminés). En linguistique, il peut en compter des centaines (alphabets, sons, nature des mots, *etc.*).

On peut définir sur le monoïde libre, à cause de l'unicité de la décomposition de chaque élément, certaines notions qui n'auraient pas de sens dans d'autres structures.

DÉFINITION 1.14 *Soit $u = u_1 u_2 \dots u_k$ un mot de A^*. L'**image miroir** de u est le mot $u_k u_{k-1} \dots u_1$. L'image miroir d'un langage est l'ensemble des images miroir de ses éléments.*

Ordres et distances sur le monoïde libre

DÉFINITION 1.15 *Soit u et v deux mots de A^*.*

- *u est un **préfixe** de v s'il existe w dans A^* tel que $v = u.w$.*
- *u est un **suffixe** de v s'il existe w dans A^* tel que $v = w.u$.*
- *u est un **facteur** de v s'il existe w et t dans A^* tels que $v = w.u.t$.*

On peut, à partir de ces notions, définir des relations d'ordre partiel sur le monoïde libre. Ainsi, un mot u est plus petit qu'un mot v selon la relation « préfixe » si et seulement si u est un préfixe de v, et on peut définir de la même manière des relations à l'aide du suffixe ou du facteur. On préfère parfois utiliser des relations d'ordre total sur A^*; nous allons ici en définir deux.

DÉFINITION 1.16 *On suppose l'alphabet A totalement ordonné. L'**ordre lexicographique** sur A^* est défini comme suit :*

- *Le mot vide est inférieur à tout mot de A^+.*
- *Quels que soient $u = a.u'$ et $v = b.v'$,*

$$u <_{\mathsf{lex}} v \Leftrightarrow \mathit{ou} \begin{cases} a < b, \\ a = b \text{ et } u' <_{\mathsf{lex}} v'. \end{cases}$$

*L'**ordre radiciel** (ou ordre militaire) est défini sur A^* par :*

$$u <_{\mathsf{rad}} v \Leftrightarrow \mathit{ou} \begin{cases} |u| < |v|, \\ |u| = |v| \text{ et } u <_{\mathsf{lex}} v. \end{cases}$$

REMARQUE 1.1 Pour l'ordre radiciel, un mot n'a qu'un nombre fini de mots qui lui sont inférieurs, ce qui n'est pas le cas pour l'ordre lexicographique.

Le monoïde A^* peut être muni d'une distance;

DÉFINITION 1.17 *Soit u et v deux mots de A^*. On note $u \wedge v$ le plus grand préfixe commun à u et v.*

DÉFINITION 1.18 *La **distance préfixe** est définie sur A^* par :*

$$d_{\mathsf{pref}}(u,v) = |u| + |v| - 2|u \wedge v|.$$

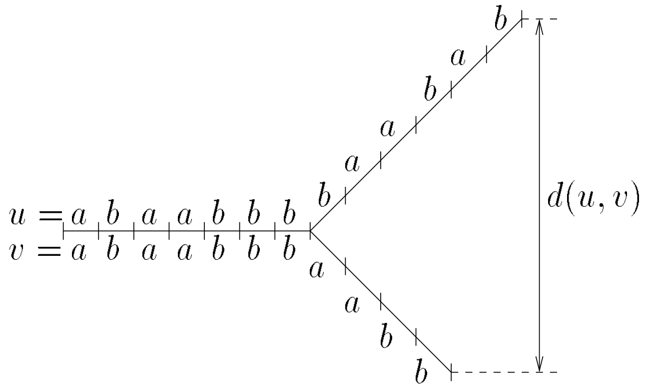


FIG. 3 – *Distance préfixe.*

EXEMPLE 2 La distance préfixe peut être vue comme l'indique la figure 3.

— ∘ —

3.2 Semi-anneaux

DÉFINITION 1.19 *Un **semi-anneau** est un ensemble $\mathbb{K}$, muni de deux lois associatives et d'un élément neutre pour chaque loi. La première loi ($\oplus$) est commutative, la seconde ($\otimes$) est distributive sur la première. On les appellera respectivement addition et multiplication. L'élément neutre de l'addition ($0_{\mathbb{K}}$) doit de plus être absorbant pour la multiplication. Dans le cas où l'on souhaite préciser quelles sont les lois du semi-anneau, on note* $(\mathbb{K}, \oplus, \otimes)$. *On note $\mathbb{K}_*$ l'ensemble des éléments de $\mathbb{K}$ différents de $0_{\mathbb{K}}$.*

EXEMPLE 3 Les nombres entiers positifs, ou naturels, $(\mathbb{N}, +, *)$ forment un semi-anneau : les deux lois sont associatives, l'addition est commutative et la multiplication est distributive sur l'addition. Comme la multiplication est elle aussi commutative, on dit que $\mathbb{N}$ est un semi-anneau commutatif. L'ensemble $\mathbb{N}$ n'est un groupe ni par rapport à son addition ni par rapport à sa multiplication. $\mathbb{N}$ peut cependant être plongé dans un anneau (les entiers relatifs $\mathbb{Z}$) et même dans un corps (les rationnels $\mathbb{Q}$). On verra toutefois qu'on ne peut pas faire de même avec n'importe quel semi-anneau.

En revanche, $(\mathbb{N}, \mathsf{max}, +)$ n'est pas un semi-anneau, bien que les deux lois soient associatives et que la seconde soit distributive sur la première; l'élément neutre de max, 0, n'est pas absorbant pour $+$, puisque c'est aussi son élément neutre.

DÉFINITION 1.20 *Soit $(M,.)$ un monoïde. Un élément x de M tel que $x.x = x$ est appelé **idempotent**. Si chaque élément de M est idempotent, M est un monoïde idempotent. Un semi-anneau $(\mathbb{K}, \oplus, \otimes)$ est idempotent si $(\mathbb{K}, \oplus)$ est un monoïde idempotent.*

REMARQUE 1.2 Un élément idempotent d'un monoïde M différent de 1_M n'est pas inversible. En effet, soit x un élément idempotent et y inverse de x, alors $x = x.x.y = x.y = 1_M$.

Un monoïde idempotent non trivial ne peut donc pas être plongé dans un groupe. Les semi-anneaux idempotents ne peuvent donc en particulier pas être plongés dans des anneaux et encore moins dans des corps.

Exemple 4 Un autre exemple connu est le semi-anneau de Boole, c'est-à-dire l'ensemble $\{0,1\}$ muni des lois :

$$\begin{array}{c|cc} \oplus & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \qquad \begin{array}{c|cc} \otimes & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

On peut voir ce semi-anneau comme l'ensemble $\{\text{VRAI,FAUX}\}$ muni des lois « OU » et « ET ». Pour définir un semi-anneau à deux éléments en respectant les axiomes, le seul choix possible porte sur la valeur de $1 \oplus 1$. Si on fixe $1 \oplus 1 = 1$, on obtient le semi-anneau de Boole, sinon $1 \oplus 1 = 0$ et le semi-anneau construit est en fait le corps $\mathbb{Z}/2\mathbb{Z}$. Ce sont les représentants de deux types de semi-anneaux. D'une part, $\mathbb{Z}/2\mathbb{Z}$ fait partie des semi-anneaux qu'on peut plonger dans un corps (c'est un corps lui-même), d'autre part, le semi-anneau de Boole est un semi-anneau non-plongeable dans un corps (c'est plus particulièrement un semi-anneau idempotent).

Remarque 1.3 Soit $(M,.)$ un monoïde. On définit la multiplication de deux sous-ensembles X et Y de M par $X.Y = \{x.y \mid x \in X \text{ et } y \in Y\}$. Alors $(\mathcal{P}(M), \cup, .)$ est un semi-anneau idempotent. Dans ce qui suit, il arrivera que l'identification soit faite entre un monoïde et le semi-anneau des parties qui lui correspond.

Nous aurons besoin de factoriser des éléments de semi-anneaux. Cette factorisation n'est pas toujours unique, sauf dans un semi-anneau **factoriel**. Ce n'est pas exactement de cette propriété dont nous aurons besoin.

Définition 1.21 *Soit $\mathbb{K}$ un semi-anneau. Un **idéal** (droit) de $\mathbb{K}$ est un sous-ensemble $\mathcal{I}$ tel que $\mathcal{I} \oplus \mathcal{I} \subseteq \mathcal{I}$ et $\mathcal{I} \otimes \mathbb{K} \subseteq \mathcal{I}$. Un idéal est finiment engendré s'il existe un ensemble fini X dans $\mathbb{K}$ tel que $\mathcal{I} = X \otimes \mathbb{K}$. Un idéal $\mathcal{I}$ est **principal** s'il existe un élément x de $\mathbb{K}$ tel que $\mathcal{I} = x \otimes \mathbb{K}$. On dira qu'un semi-anneau est un **semi-anneau principal** si, pour tout idéal $\mathcal{I}$, il existe un unique idéal principal minimum qui contient $\mathcal{I}$.*

Soit $\mathbb{K}$ un semi-anneau principal, X un ensemble d'éléments de $\mathbb{K}$ et $\mathcal{I}$ le plus petit idéal principal qui contient X. Tout élément z qui engendre l'idéal $\mathcal{I}$ est un pgcd *de X.*

Remarque 1.4 D'un point de vue effectif, la non unicité du pgcd peut poser un problème. Nous verrons le moment voulu comment le résoudre. On peut dès à présent remarquer qu'il est possible de passer d'un pgcd à un autre par multiplication à droite, puisque chacun appartient à l'idéal engendré par l'autre.

Exemple 5 Le semi-anneau $(\mathbb{R}_+ \cup \{-\infty\}, \mathsf{max}, +)$ est principal. En effet, pour tout ensemble X, l'adhérence (pour la topologie usuelle) de l'idéal engendré par X est l'idéal principal engendré par $\mathsf{inf}\{x \in X \mid x \neq -\infty\}$.

Le semi-anneau $(\mathbb{R}\cup\{-\infty\}, \mathsf{max}, +)$ est lui aussi principal. Les deux seuls idéaux de ce semi-anneau sont le semi-anneau lui-même, engendré par n'importe quel élément différent de $-\infty$ et $\{-\infty\}$.

En revanche, le semi-anneau $(\mathcal{P}(B^*), \cup, .)$, avec $B = \{a,b\}$ n'est pas principal. Par exemple, soit $x = \{a,b,a^2,ab,ba,aba\}$ et $y = \{a,b,ab,a^2,ba,b^2,aba,ab^2\}$. Ces deux ensembles admettent $\{1_{B^*},a\}$ et $\{a,b,ab\}$ pour facteurs maximaux gauches communs :

$$
\begin{aligned}
x &= \{1_{B^*},a\}.\{a,b,ba\} & \qquad x &= \{a,b,ab\}.\{1_{B^*},a\}\\
y &= \{1_{B^*},a\}.\{a,b,ba,bb\} & \qquad y &= \{a,b,ab\}.\{1_{B^*},a,b\}
\end{aligned}
$$

Aucun de ces facteurs ne divise l'autre, il n'y a donc pas unicité de l'idéal principal minimal qui contient x et y.

Définition 1.22 *Soit $(\mathbb{K},\oplus,\otimes)$ un semi-anneau. Pour tout x dans $\mathbb{K}$, pour tout n dans $\mathbb{N}$, x^n est défini inductivement par $x^0 = 1_{\mathbb{K}}$ et $x^n = x \otimes x^{n-1}$. On pose la quantité suivante, pour n dans $\mathbb{N}$:*

$$X_n = \bigoplus_{k=0}^{n} x^k$$

*Si X_n admet une limite dans $\mathbb{K}$ lorsque n tend vers l'infini, cette limite est appelée **étoile** de x et notée x^*. Dans ce cas, on note*

$$x^* = \bigoplus_{k=0}^{\infty} x^k, \quad x^+ = \bigoplus_{k=1}^{\infty} x^k.$$

Remarque 1.5 Puisqu'on parle de limite dans $\mathbb{K}$, ce semi-anneau est implicitement muni d'une topologie. Il peut, par défaut, être muni de la topologie discrète, mais il est plus naturel de munir certains semi-anneaux ($\mathbb{R}$,$\mathbb{Q}$, ...) de leur topologie habituelle. Le choix de celle-ci n'est pas sans conséquence sur la définition de l'étoile.

Exemple 6 Dans $\mathbb{Q}$, l'étoile de $1/2$ est la limite, lorsque n tend vers l'infini, de la somme :

$$X_n = \sum_{i=0}^{n} \left(\frac{1}{2}\right)^i = 2 - \left(\frac{1}{2}\right)^n.$$

Cette suite converge pour la topologie habituelle mais pas pour la topologie discrète.

Remarque 1.6 L'étoile d'un élément x d'un monoïde M est toujours définie. En effet les sous-ensembles X_n forment dans ce cas une suite croissante dont la limite est :

$$X = \{y \in M \mid \exists n, y = x^n\}.$$

Plus généralement, l'étoile de n'importe quel sous-ensemble de M est définie.

Définition 1.23 *Soit $\mathbb{K}$ un semi-anneau et P un sous-ensemble de $\mathbb{K}$. Un élément x de $\mathbb{K}$ est **rationnel** par rapport à P s'il peut être obtenu à partir d'éléments de P et des opérations d'addition, de multiplication et d'étoile.*

Remarque 1.7 L'ensemble des éléments de $\mathbb{R}$ rationnels par rapport à $\mathbb{Z}$ n'est pas $\mathbb{Q}$! Avec la topologie usuelle, le seul élément de $\mathbb{Z}$ auquel on peut appliquer l'opération étoile est 0, donc l'ensemble des éléments de $\mathbb{R}$ rationnels par rapport à $\mathbb{Z}$ est $\mathbb{Z}$.

En revanche, munissons $\mathbb{Z}$ de la norme 2-adique : pour tout entier n qui s'écrit $n = 2^k m$, avec $\mathsf{pgcd}(2,m) = 1$, $|n|_2 = 2^{-k}$. Alors l'ensemble des éléments de $\mathbb{Q}_2$ (clôture 2-adique de $\mathbb{Q}$) rationnels par rapport à $\mathbb{Z}$ contient par exemple $-1/3$, car la suite $\sum_{k=0}^{n} 4^k = \frac{4^{(n+1)}-1}{3}$ est une suite de Cauchy, donc converge dans $\mathbb{Q}_2$ et 4^* vérifie $1 + 4.4^* = 4^*$, donc $4^* = -1/3$.

Nous emploierons parfois le terme de semi-module. Il s'agit de la généralisation du concept de module.

Définition 1.24 *Soit $\mathbb{K}$ un semi-anneau, V est un $\mathbb{K}$-**semi-module** (à gauche) si V est muni d'une loi $\oplus$ telle que $(V,\oplus)$ est un monoïde d'élément neutre 0_V et s'il existe une action (à gauche) de $(\mathbb{K},\otimes)$ sur V (notée par simple concaténation d'un élément de $\mathbb{K}$ avec un élément de V) telle que :*

$$\begin{aligned} &\forall x \in V,\ 0_{\mathbb{K}}\, x = 0_V \\ &\forall k,k' \in \mathbb{K}, \forall x,y \in V,\ (k \oplus k')\, x = k\, x \oplus k'\, x \\ &\qquad k\,(x \oplus y) = k\, x \oplus k\, y. \end{aligned}$$

*Un semi-module est dit finiment engendré, ou de **type fini**, s'il existe un ensemble fini $\{x_1,x_2,\dots,x_n\}$ tel que, pour tout x, il existe $(k_1,k_2,\dots,k_n)$ dans $\mathbb{K}^n$ tel que*

$$x = \bigoplus_{r=1}^{n} k_r x_r.$$

Nous nous intéresserons aussi à des sous-ensembles particuliers de semi-modules : les faisceaux finis de droites, que nous appellerons aussi $\mathbb{K}$-cônes finiment engendrés.

Définition 1.25 *Soit V un semi-module et X un sous-ensemble de V. L'ensemble X est un $\mathbb{K}$-**cône** finiment engendré s'il existe un ensemble fini d'éléments de V, $\{x_i \mid i \in I\}$, tel que :*

$$X = \bigcup_{i \in I} \mathbb{K} x_i.$$

— ◦ —

3.3 Polynômes et séries formelles

Par souci de simplification, on définit ici uniquement les polynômes et séries formelles sur A^*. On trouvera une étude plus complète des séries sur un monoïde dans [55].

Définition 1.26 *Soit A un alphabet et $(\mathbb{K},\oplus,\otimes)$ un semi-anneau. On appelle semi-anneau des **series** (formelles) sur A^* à coefficients (ou à multiplicité) dans $\mathbb{K}$, qu'on note $\mathbb{K}\langle\langle A^*\rangle\rangle$,* *l'ensemble des applications de A^* dans $\mathbb{K}$ muni des opérations :*

$$\textit{d'addition}: \alpha \oplus \beta : x \mapsto x\alpha \oplus x\beta,$$
$$\textit{et de multiplication}: \alpha \otimes \beta : x \mapsto \bigoplus_{y.z=x} y\alpha \otimes z\beta.$$

Pour une série α, on utilise de préférence la notation suivante : pour tout mot u de A^*, $\langle \alpha,u\rangle = u\alpha$. On note formellement la série comme une somme infinie :

$$\alpha = \bigoplus_{u\in A^*} \langle \alpha,u\rangle u.$$

Si on identifie un scalaire k de $\mathbb{K}$ à la série $k\,1_{A^*}$, on définit du même coup la multiplication d'une série par un scalaire. Les séries sont donc un $\mathbb{K}$-semi-module.

Remarque 1.8 Comme on se place dans le monoïde libre, la somme effectuée dans la définition de la multiplication est une somme finie. On peut définir les séries sur d'autres monoïdes que le monoïde libre. Toutefois, si celui-ci n'est pas gradué, c'est-à-dire s'il existe des éléments admettant un nombre infini de factorisations, la multiplication peut ne pas être définie partout. Les séries ne forment alors pas un semi-anneau.
La multiplication des séries faisant intervenir la multiplication du monoïde libre, elle n'est pas commutative, même si le semi-anneau l'est (sauf si l'alphabet n'a qu'une lettre).

Définition 1.27 *Le **support d'une série** s de $\mathbb{K}\langle\langle A^*\rangle\rangle$, noté $\mathsf{Supp}(s)$ est l'ensemble des mots dont les coefficients sont non nuls :*

$$\mathsf{Supp}(s) = \{u \in A^* \mid \langle s,u\rangle \neq 0_{\mathbb{K}}\}.$$

*Une série dont le support est fini est un **polynôme**. L'ensemble des polynômes forme un sous-semi-anneau de $\mathbb{K}\langle\langle A^*\rangle\rangle$ noté $\mathbb{K}\langle A^*\rangle$.*

Remarque 1.9 L'ensemble des polynômes en variables commutatives est généralement noté $\mathbb{K}[A]$, où A est l'alphabet de variables. On préfère utiliser la notation $\mathbb{K}\langle M\rangle$, où M est le monoïde des variables. Dans le cas des variables commutatives, on noterait le semi-anneau des polynômes $\mathbb{K}\langle A^{\oplus}\rangle$, où $A^{\oplus}$ est le monoïde *commutatif* libre engendré par A. Cette notation inclut non seulement les variables, mais les relations éventuelles qui existent entre elles.

Définition 1.28 *On appelle **terme constant de la série** s et on note $\mathsf{c}(s)$ le coefficient du mot vide dans s : $\langle s,1_{A^*}\rangle$. La **partie propre d'une série** s est la série s_p définie par :*

$$\langle s_p,1_{A^*}\rangle = 0_{\mathbb{K}}\ , \quad \forall u \in A^+,\ \langle s_p,u\rangle = \langle s,u\rangle.$$

On peut donc écrire $s = \mathsf{c}(s)1_{A^*} \oplus s_p$.

Comme dans tout semi-anneau, on peut vouloir calculer l'étoile d'une série. La définition d'une telle quantité est fortement liée à la définition de l'étoile dans le semi-anneau $\mathbb{K}$:

Proposition 1.1 *[8] L'étoile d'une série s de $\mathbb{K}\langle\langle A^*\rangle\rangle$ est définie si et seulement si l'étoile du terme constant de s est définie dans $\mathbb{K}$. On a alors l'égalité suivante :*

$$s^* = (\mathsf{c}(s)^*\, s_p)^*\, \mathsf{c}(s)^*.$$

Remarque 1.10 L'étoile d'une série propre est toujours définie, en effet :

$$\begin{aligned}\langle s_p^*,u\rangle &= \bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\ldots,v_n\in A^*}} \langle s_p^*,v_1\rangle \otimes \ldots \otimes \langle s_p^*,v_n\rangle\\ &= \bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\ldots,v_n\in A^+}} \langle s_p^*,v_1\rangle \otimes \ldots \otimes \langle s_p^*,v_n\rangle.\end{aligned}$$

Le membre droit de cette équation est une somme finie.

Remarque 1.11 Si l'étoile est toujours définie sur le semi-anneau $\mathbb{K}$, elle est d'après la proposition, toujours définie sur $\mathbb{K}\langle\langle A^*\rangle\rangle$. Ainsi, si $\mathbb{K}$ est le semi-anneau de Boole, on retrouve le cas particulier de la remarque 1.3, puisque $\mathbb{B}\langle\langle A^*\rangle\rangle$ est isomorphe à $\mathcal{P}(A^*)$.

Définition 1.29 *Le **quotient à gauche d'une série** s par un mot u de A^* est la série*

$$u^{-1}s = \bigoplus_{v\in A^*} \langle s,uv\rangle v.$$

Il n'est pas difficile de voir que le quotient ainsi défini est une action (à droite) des mots sur les séries.

— ∘ —

3.4 Ensembles et séries rationnels

Définition 1.30 *Soit M un monoïde. L'ensemble* Rat M *des **ensembles rationnels** de M est la clôture des ensembles finis de M par les opérations de produit, d'union et d'étoile. Ce qui revient à dire que les ensembles rationnels de M sont les éléments de $\mathcal{P}(M)$ rationnels par rapport aux ensembles finis. Si M est le monoïde libre, on parle de **langage rationnel**.*

Définition 1.31 *Le semi-anneau des **séries rationnelles** $\mathbb{K}\mathrm{Rat}\, A^*$ de $\mathbb{K}\langle\langle A^*\rangle\rangle$ est la clôture du semi-anneau des polynômes par les opérations d'addition, de multiplication et d'étoile, si cette dernière est définie.*

Les ensembles rationnels d'un monoïde M et les séries rationnelles forment des sous-semi-anneaux de $\mathcal{P}(M)$ et des séries formelles respectivement.

Exemple 1.3 Soit $A = \{a,b\}$ un alphabet. Formons un langage rationnel dans le monoïde libre A^*. Le langage ab (en fait le singleton $\{ab\}$) est un langage fini, le langage $a+b$ aussi. Le langage $(a+b)^*$ est l'étoile d'un langage fini, donc il est rationnel. En fait, il s'agit de A^* tout entier. Le langage $\mathcal{L}_1 = (a+b)^*ab(a+b)^*$ est un produit de langages rationnels ; il est donc aussi rationnel. Intuitivement ce langage contient tous les mots qui contiennent le facteur ab. Pour former un mot qui appartient à ce langage, on peut en effet mettre ce que l'on veut au début ou à la fin ($(a+b)^*$) à condition d'insérer ab au milieu.

On peut analyser un peu plus finement ce que représente cette description. Plaçons nous dans le cadre des séries à multiplicité dans $\mathbb{N}$. Le mot ab est alors vu comme un polynôme formé d'un seul monôme dont le coefficient est 1. La série $\chi_{A^*} = (a+b)^*$ est la série caractéristique de A^*, c'est-à-dire que pour tout mot u de A^*, $\langle \chi_{A^*}, u \rangle = 1$. La série $s_1 = (a+b)^* ab (a+b)^*$ est donc une série rationnelle. La multiplicité d'un mot u dans s_1 est le nombre de façons dont u se factorise en $u = v.ab.w$, c'est donc le nombre de facteurs ab qui apparaissent dans u.

— ◦ —

3.5 Ensembles reconnaissables

Définition 1.32 *Soit M un monoïde. Un sous-ensemble $\mathcal{L}$ de M est un* ***ensemble reconnaissable*** *(par morphisme) s'il existe un monoïde fini N, un morphisme φ de M dans N tel que $\mathcal{L} = (\mathcal{L}\varphi)\varphi^{-1}$ (en d'autres termes, le sous-ensemble $\mathcal{L}$ est l'image inverse par φ d'une partie de N). Un langage $\mathcal{L}$ de A^* est un* ***langage reconnaissable*** *si c'est un sous-ensemble reconnaissable de A^*.*

Exemple 7.1 Soit $A = \{a,b\}$ et $N_3 = (\mathbb{Z}/3\mathbb{Z},+)$. Soit $\varphi : A^* \to N_3$ le morphisme défini par $a\varphi = 1$ et $b\varphi = -1$. L'image d'un mot de A^* dans N_3 par φ est donc la différence modulo 3 entre le nombre de a et le nombre de b. Le langage $\mathcal{L}_3 = \{-1,1\}\varphi^{-1}$ est un langage reconnaissable; c'est l'ensemble des mots de A^* dont le nombre de a est différent du nombre de b modulo 3.

Exemple 1.4 Soit φ le morphisme de A^* dans le monoïde M_1 (présenté page 22) défini par $a\varphi = x$ et $b\varphi = y$. Un examen attentif (d'autres techniques nous permettront de le vérifier) nous permet de voir que :

$$\begin{aligned} 1_{M_1}\varphi^{-1} &= 1_{A^*}, & t\varphi^{-1} &= b^+a^+, \\ x\varphi^{-1} &= a^+, & 0_{M_1}\varphi^{-1} &= A^*abA^* = \mathcal{L}_1. \\ y\varphi^{-1} &= b^+, \end{aligned}$$

Le langage $\mathcal{L}_1$, image inverse de 0_{M_1} par φ est donc un langage reconnaissable de A^*.

— ◦ —

4 Automates

4.1 Automates sur un semi-anneau

Définition 1.33 *Un* ***automate $\mathcal{A}$ sur un semi-anneau*** *$\mathbb{K}$ est défini comme un quintuplet $\langle\, Q,\mathbb{K},E,I,T \,\rangle$, où Q est un ensemble fini d'**états**, E une matrice de $\mathbb{K}^{Q\times Q}$, I et T deux*

vecteurs (respectivement ligne et colonne) de $\mathbb{K}^Q$. *E est la matrice de transition de $\mathcal{A}$, I et T sont respectivement les vecteurs initial et final de $\mathcal{A}$.*
Les éléments des supports de E, I et T sont respectivements appelés ***transitions****, états initiaux et états finals (ou terminaux).*

Définition 1.34 *Le* ***graphe sous-jacent à un automate*** $\mathcal{A} = \langle\, Q,\mathbb{K},E,I,T\,\rangle$ *est le graphe orienté* $\mathcal{G}_{\mathcal{A}} = \langle\, Q, \mathsf{Supp}(E)\,\rangle$.

Définition 1.35 *Soit* $\mathcal{A} = \langle\, Q,\mathbb{K},E,I,T\,\rangle$. *Un chemin de $\mathcal{A}$ de longueur k est une suite d'états $(p_0,p_1,\ldots,p_k)$ telle que, pour tout i dans $[1;k]$, (p_{i-1},p_i) est une transition. L'étiquette d'un chemin $\mathcal{C}$ est l'élément $E_{\mathcal{C}} = E_{p_0,p_1} \otimes E_{p_1,p_2} \otimes \ldots \otimes E_{p_{k-1},p_k}$. Ce chemin est* ***réussi*** *si p_0 est un état initial et p_k est un état final. Le* ***calcul*** *correspondant à un chemin réussi $\mathcal{C}$ est $I_{p_0} \otimes E_{\mathcal{C}} \otimes T_{p_k}$. L'élément de $\mathbb{K}$* ***reconnu*** *par l'automate est la somme des calculs des chemins réussis de l'automate, si elle est définie. Deux automates sont* ***équivalents*** *s'ils reconnaissent le même élément de $\mathbb{K}$.*

Définition 1.36 *Un automate* $\mathcal{A} = \langle\, Q,\mathbb{K},E,I,T\,\rangle$ *est* ***normalisé*** *s'il ne comporte qu'un état initial i et un état final t avec $I_i = T_t = 1_{\mathbb{K}}$, que ces états sont distincts, et que, pour tout p dans Q, $E_{p,i} = E_{t,p} = 0_{\mathbb{K}}$.*

Proposition 1.2 *Tout automate est équivalent à un automate normalisé.*

Démonstration. Soit $\mathcal{A} = \langle\, Q,\mathbb{K},E,I,T\,\rangle$ un automate. Soit i et t deux états qui n'appartiennent pas à Q. On pose $Q' = Q \cup \{i,t\}$ et on définit E' dans $\mathbb{K}^{Q' \times Q'}$ par :

$$E'_{p,q} = \begin{cases} E_{p,q} & \text{si } p,q \in Q \\ I_q & \text{si } p = i \text{ et } q \in Q \\ T_p & \text{si } q = t \text{ et } p \in Q \\ 0_{\mathbb{K}} & \text{si } p = t,\ q = i \text{ ou } (p,q) = (i,t) \end{cases}$$

L'automate $\mathcal{A}' = \{Q',\mathbb{K},E',i,t\}$[3] est normalisé. Il y a une bijection entre les chemins réussis de $\mathcal{A}$ et ceux de $\mathcal{A}'$:

$$(p_0,\ldots,p_k) \longmapsto (i,p_0,p_1,\ldots,p_k,t)$$

De plus, le calcul correspondant à un chemin réussi de $\mathcal{A}$ et celui correspondant à son image dans $\mathcal{A}'$ sont égaux. La somme des calculs des deux automates est donc la même; ils sont équivalents. □

— ∘ —

Examinons le rapport entre les produits matriciels qu'on peut effectuer sur E, I ou T et les étiquettes des chemins.

Le produit $I \otimes T = \bigoplus_{p \in Q} I_p \otimes T_p$ est la somme des calculs de longueur 0 de l'automate.

3. Pour alléger l'écriture, on note i (*resp.* t) le vecteur caractéristique de i (*resp.* de t).

La matrice E représente les étiquettes des transitions, c'est-à-dire des chemins de longueur 1; le produit $I \otimes E \otimes T = \bigoplus_{p,q \in Q} I_p \otimes E_{p,q} \otimes T_q$ est donc la somme des calculs de longueur 1 de l'automate.

On peut généraliser cette constatation :

PROPOSITION 1.3 *Soit $\mathcal{A} = \langle Q, \mathbb{K}, E, I, T \rangle$ un automate. La somme des calculs de longueur k de l'automate est $I \otimes E^k \otimes T$. L'élément reconnu par l'automate, s'il est défini, est $I \otimes E^* \otimes T$.*

Démonstration. On montre par récurrence sur la longueur des chemins que $(E^k)_{p,q}$ est la somme des étiquettes des chemins de longueur k entre p et q. C'est trivialement vrai pour $k = 0$. Si le résultat est vrai pour k, tout chemin de p à q de longueur $k+1$ se décompose en un chemin de p à r, pour un certain état r, et une transition de r à q. La somme des étiquettes des chemins de longueur $k+1$ entre p et q est donc

$$\bigoplus_{r \in Q} (E^k)_{p,r} \otimes E_{r,q} = (E^{k+1})_{p,q},$$

ce qui montre le résultat. La somme des calculs de longueur k entre p et q est donc $I_p \otimes E^k_{p,q} \otimes T_q$ et la somme des calculs de longueur k de l'automate est

$$\bigoplus_{p,q \in Q} I_p \otimes (E^k)_{p,q} \otimes T_q = I \otimes E^k \otimes T.$$

La somme des calculs de l'automate est

$$\bigoplus_{k \in \mathbb{N}} I \otimes E^k \otimes T = I \otimes \left(\bigoplus_{k \in \mathbb{N}} E^k \right) \otimes T = I \otimes E^* \otimes T$$

□

PROPOSITION 1.4 *Les coefficients de l'étoile d'une matrice sont rationnels par rapport aux coefficients de la matrice.*

— ∘ —

4.2 Automates sur un monoïde

La définition d'un automate sur un monoïde découle de l'identification du monoïde M avec les singletons du semi-anneau $\mathcal{P}(M)$. On va donner explicitement les définitions que l'on obtient.

DÉFINITION 1.37 *Un **automate $\mathcal{A}$ sur un monoïde** M est défini comme un quintuplet $\langle Q, M, E, I, T \rangle$, où Q est un ensemble fini d'**états**, où E est une matrice de $(\operatorname{Rat} M)^{Q \times Q}$ et I et T des vecteurs de $(\operatorname{Rat} M)^Q$.*

Les chemins, étiquettes des chemins et calculs d'un automate sont définis de la même façon que dans le cas des semi-anneaux.

Définition 1.38 *Soit $\mathcal{A}$ un automate sur M. Un élément de M est* ***accepté*** *par $\mathcal{A}$ s'il appartient à un calcul de $\mathcal{A}$. L'élément de $\mathcal{P}(M)$ reconnu par $\mathcal{A}$ est l'****ensemble reconnu par l'automate****. L'addition du semi-anneau $\mathcal{P}(M)$ étant l'union, l'ensemble reconnu par l'automate $\mathcal{A}$ est l'ensemble des mots acceptés par $\mathcal{A}$.*

La proposition 1.4 a pour corollaire immédiat que l'ensemble reconnu par un automate est rationnel. Réciproquement, à partir d'une expression rationnelle, il existe plusieurs méthodes classiques pour construire un automate qui dénote le même ensemble. Nous n'y revenons pas ici, mais nous aurons l'occasion d'en décrire plusieurs dans les chapitres suivants.

Théorème 1.1 *Un ensemble est rationnel si et seulement s'il peut être reconnu par un automate fini.*

Remarque 1.12 Il convient de souligner la différence entre un ensemble reconnaissable (reconnu par un morphisme) et un ensemble rationnel (reconnu par un automate). Un ensemble rationnel peut ne pas être reconnaissable. Par exemple, dans le monoïde $(\mathbb{N}^2,+)$, l'ensemble $\{(x,y) \mid x \leqslant y\}$ peut être reconnu par un automate et n'est pourtant pas reconnaissable. Nous verrons cependant que dans le monoïde libre ces deux notions coïncident.

Définition 1.39 *Soit $\mathcal{A} = \langle\, Q,M,E,I,T \,\rangle$ un automate. Pour tout état p de Q, le* ***passé*** *de p dans $\mathcal{A}$, noté $\mathsf{Past}_{\mathcal{A}}(p)$, est l'ensemble reconnu par l'automate $\langle\, Q,M,E,I,p \,\rangle$. De même, le* ***futur*** *de p dans $\mathcal{A}$, noté $\mathsf{Fut}_{\mathcal{A}}(p)$, est l'ensemble reconnu par l'automate $\langle\, Q,M,E,p,T \,\rangle$ et, pour tout couple d'états (p,q), l'ensemble de transition $\mathsf{Trans}_{\mathcal{A}}(p,q)$ est l'ensemble reconnu par l'automate $\langle\, Q,M,E,p,q \,\rangle$.*

Le passé (*resp.* le futur) est donc l'ensemble des éléments qui étiquettent des débuts de calculs menant en p (*resp.* des fins de calculs venant de p). Le produit d'un élément du passé par un élément du futur étiquette donc un calcul de l'automate passant par p. Donc $\mathsf{Past}_{\mathcal{A}}(p).\mathsf{Fut}_{\mathcal{A}}(p)$ est inclus dans l'ensemble reconnu par $\mathcal{A}$.
De même, $\mathsf{Past}_{\mathcal{A}}(p).\mathsf{Trans}_{\mathcal{A}}(p,q).\mathsf{Fut}_{\mathcal{A}}(q)$ est inclus dans cet ensemble.

Définition 1.40 *Soit $\mathcal{A} = \langle\, Q,M,E,I,T \,\rangle$ et $\mathcal{B} = \langle\, R,M,F,J,U \,\rangle$ deux automates. Une application μ de Q dans R est un morphisme d'automates de $\mathcal{A}$ dans $\mathcal{B}$, si*

- *pour tout élément p de Q, I_p est inclus dans $J_{p\mu}$ et T_p est inclus dans $U_{p\mu}$,*
- *pour tout couple (p,q) de $Q \times Q$, $E_{p,q}$ est inclus dans $F_{p\mu,q\mu}$.*

Proposition 1.5 *Soit $\mathcal{A}$ et $\mathcal{B}$ deux automates sur un monoïde M. S'il existe un morphisme de $\mathcal{A}$ dans $\mathcal{B}$, l'ensemble reconnu par $\mathcal{A}$ est inclus dans celui reconnu par $\mathcal{B}$.*

Définition 1.41 *Un automate $\mathcal{A} = \langle\, Q,M,E,I,T \,\rangle$ est* ***non-ambigu*** *si, pour tout élément x de M, il existe au plus un seul chemin réussi de $\mathcal{A}$, $(p_0,p_1,\ldots,p_k)$, au calcul duquel appartient x, et, le cas échéant, une seule factorisation de $x = x_0.x_1.x_2\ldots x_k.x_{k+1}$ telle* *que :*

$$\forall i \in [1;k],\ x_i \in E_{p_{i-1},p_i},$$
$$x_0 \in I_{p_0} \qquad et \qquad x_{k+1} \in T_{p_k}.$$

Cette dernière notion est globale, ce qui permet de la définir dans n'importe quel monoïde, ce qui n'est pas le cas du *déterminisme* qui n'a un sens que dans le monoïde libre.

— ◦ —

4.3 Automates sur un alphabet

Lorsqu'on travaille dans le monoïde libre (A^*), on préfère manipuler des automates dans lesquels chaque transition correspond à une seule lettre. Ces automates particuliers, « temps-réel » (puisqu'à chaque opération représentée par une lettre, on effectue une transition), sont les automates classiques. Ils permettent d'établir le lien entre les langages rationnels et reconnaissables.

DÉFINITION 1.42 *Un* ***automate*** $\mathcal{A}$ *est un quintuplet* $\langle Q,A,E,I,T \rangle$*, où* Q *est un ensemble fini d'états,* A *un alphabet fini,* E *un sous-ensemble de* $Q \times A \times Q$ *appelé ensemble des* ***transitions*** *et* I *(resp.* T*) un sous-ensemble de* Q *regroupant les* ***états initiaux****(resp.* ***terminaux****).*
L'étiquette d'une transition (p,a,q) *de* E *est* a*; pour tout* e *dans* E*,* $|e|$ *représente l'étiquette de* e*.*

EXEMPLE 1.5 Soit $\mathcal{A}_1 = \langle Q,A,E,I,T \rangle$ l'automate défini par :

$$\begin{aligned} Q &= \{p,q,r\},\\ A &= \{a,b\},\\ E &= \{(p,a,p),(p,b,p),(p,a,q),(q,b,r),(r,a,r),(r,b,r)\},\\ I &= \{p\},\\ T &= \{r\}. \end{aligned}$$

La représentation graphique de cet automate est donnée figure 4. Les états initiaux sont indiqués par une flèche entrante et les états terminaux par une flèche sortante. Le graphe sous-jacent de cet automate est celui de la figure 1.

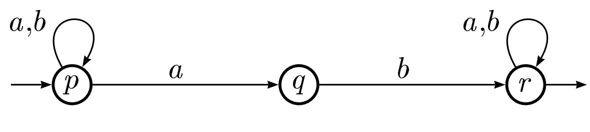


FIG. 4 – *Un automate à trois états*

Définition 1.43 *Soit* $\mathcal{A} = \langle\, Q,A,E,I,T\,\rangle$*. Un* ***chemin*** *de* $\mathcal{A}$ *est une suite de transitions dont les images dans le graphe sous-jacent forment un chemin. L'****étiquette d'un chemin*** *est le mot formé par la concaténation des étiquettes des transitions qui le constituent. Un* ***chemin réussi*** *(aussi appelé* ***calcul****) de* $\mathcal{A}$ *est un chemin dont l'état de départ appartient à* I *et celui d'arrivée à* T*.*

Définition 1.44 *Un mot de* A^* *est* ***accepté*** *par un automate* $\mathcal{A}$ *s'il existe un calcul de* $\mathcal{A}$ *dont ce mot est l'étiquette. Le* ***langage reconnu*** *par un automate est l'ensemble des mots qu'il accepte. Deux automates sont* ***équivalents*** *s'ils reconnaissent le même langage.*

Théorème 1.2 *Un langage est rationnel si et seulement s'il existe un automate qui reconnaît ce langage.*

On verra en effet dans le chapitre 6 qu'à partir d'une expression rationnelle, on peut construire un automate qui reconnaît le langage représenté par l'expression.

Exemple 1.6 On voit facilement que l'automate de la figure 4 accepte tous les mots de $\{a,b\}^*$ qui contiennent un facteur ab et que ce sont les seuls mots acceptés. Le langage reconnu par cet automate est donc $\mathcal{L}_1 = \{a,b\}^*ab\{a,b\}^*$.

Pour des raisons de facilité, ou pour obtenir une description plus compacte, on peut parfois recourir à l'utilisation dans les automates de transitions qui ne sont pas étiquetées par des lettres mais par le mot vide. Ces transitions que nous appelons **transitions spontanées**, parce qu'elles correspondent au passage d'un état dans un autre sans qu'aucune opération ne soit effectuée, sont parfois appelées ε-transitions. Nous renonçons à cette appellation qui fait référence au symbole ε qui dénote alors le mot vide et auquel nous préférons 1_{A^*}.

Définition 1.45 *Un état* q *d'un automate est un* ***état accessible*** *s'il existe un chemin d'un état initial vers* q*; il est* ***co-accessible*** *s'il existe un chemin de* q *vers un état terminal. Un automate est* ***émondé*** *si tous ses états sont accessibles et co-accessibles.*

Proposition 1.6 *Tout automate est équivalent à un automate émondé.*

En effet, si un état d'un automate n'est pas à la fois accessible et co-accessible, on peut le supprimer sans changer le langage reconnu par l'automate.

Définition 1.46 *Un automate sur un alphabet* A *est* ***déterministe*** *s'il n'a qu'un état initial et si, de chaque état, part au plus une transition étiquetée par chaque lettre. De même, il est* ***co-déterministe*** *s'il n'a qu'un état final et si, dans chaque état, arrive au plus une transition étiquetée par chaque lettre.*

Définition 1.47 *Un automate sur un alphabet* A *est* ***complet*** *si de chaque état part au moins une transition étiquetée par chaque lettre.*

On considère généralement cette notion dans le cadre des automates déterministes. De chaque état part alors une et une seule transition étiquetée par chaque lettre. On peut toujours rendre un automate complet en ajoutant un état non co-accessible dans lequel arrive les transitions qu'on rajoute pour obtenir un automate complet.

DÉFINITION 1.48 *Pour tout état p de Q, pour toute lettre a de A, on note*

$$p \cdot a = \{q \in Q \mid (p,a,q) \in E\}$$

l'ensemble des ***successeurs*** *de p par a.*
Si X est un sous-ensemble de Q,

$$X \cdot a = \bigcup_{p \in X} p \cdot a.$$

Les successeurs d'un état p ou d'un ensemble X par un mot $u = u'a$ sont définis récursivement par :

$$p \cdot u = (p \cdot u') \cdot a\,, \quad X \cdot u = (X \cdot u') \cdot a.$$

Le monoïde libre A^* agit ainsi à droite sur l'ensemble $\mathcal{P}(Q)$. Si $\mathcal{A}$ est un automate déterministe, A^* agit à droite sur Q.

On peut symétriquement définir l'ensemble des ***prédécesseurs*** *de p par a et étendre cette notion aux prédécesseurs d'un sous-ensemble X de Q par un mot u. On définit ainsi une action à gauche de A^* sur $\mathcal{P}(Q)$ qui, si l'automate est co-déterministe, est plus particulièrement une action à gauche de A^* sur Q.*

— ∘ —

On l'a vu, les transitions peuvent être considérées comme une matrice de $\mathcal{P}(A)^{Q\times Q}$. Dans les automates sur A, l'accent est mis sur les générateurs. Au lieu d'utiliser la matrice de transition E, on préfère considérer la fonction $\mu : A \mapsto \mathbb{B}^{Q\times Q}$ telle que :

$$E = \sum_{a \in A^*} (a\mu)a$$

DÉFINITION 1.49 *La* ***représentation linéaire*** *d'un automate $\mathcal{A} = \langle\, Q,A,E,I,T\,\rangle$ est un triplet (λ,μ,ν), où λ et ν sont respectivement des vecteurs ligne et colonne de $\mathbb{B}^Q$, et μ un morphisme de A^* dans $\mathbb{B}^{Q\times Q}$ tels que :*

$$\forall p \in Q,\ \lambda_p = 1_{\mathbb{B}} \Leftrightarrow p \in I\,, \quad \nu_p = 1_{\mathbb{B}} \Leftrightarrow p \in T,$$
$$\forall p,q \in Q,\ \forall a \in A,\ (a\mu)_{p,q} = 1_{\mathbb{B}} \Leftrightarrow (p,a,q) \in E.$$

PROPOSITION 1.7 *Soit $\mathcal{A}$ un automate et (λ,μ,ν) sa représentation linéaire. Le monoïde (fini) engendré par $A\mu$ est appelé* ***monoïde de transition*** *de $\mathcal{A}$. Il reconnaît le langage reconnu par $\mathcal{A}$.*

En effet, les mots u acceptés par $\mathcal{A}$ sont exactement ceux pour lesquels il existe un couple $(i,t) \in I \times T$ tel que $(u\mu)_{i,t} \neq 0_{\mathbb{B}}$.

THÉORÈME 1.3 *(Kleene) Soit A un alphabet fini. Un langage de A^* est rationnel si et seulement s'il est reconnaissable.*

La proposition 1.7 montre qu'un langage rationnel est reconnaissable. Réciproquement, soit $\mathcal{L}$ un langage reconnu par le morphisme μ de A^* sur un monoïde fini M. A partir du **graphe de Cayley** de M, on peut facilement construire un automate déterministe qui accepte $\mathcal{L}$:

$$\mathcal{A} = \langle\, M,A,\{(x,a,y) \mid y = x.(a\mu)\},1_M,\mathcal{L}\mu \,\rangle.$$

COROLLAIRE 1.8 *Tout automate est équivalent à un automate déterministe.*

Il suffit en effet de considérer le graphe de Cayley du monoïde de transition de l'automate de départ $\mathcal{A}$. Le monoïde de transition $M_{\mathcal{A}}$ est, rappelons-le, un ensemble de relations de Q dans Q, où Q est l'ensemble des états de $\mathcal{A}$. En fait, on peut obtenir à partir de $\mathcal{A}$ un automate plus petit que ce graphe de Cayley. La seule chose qui importe pour savoir si un élément α de $M_{\mathcal{A}}$ est final, est le fait que $I\alpha$ contienne ou non un état final de $\mathcal{A}$. On peut donc définir la relation d'équivalence sur $M_{\mathcal{A}}$: $\alpha \sim \beta \Leftrightarrow I\alpha = I\beta$. Attention, cette relation n'est qu'une congruence *droite*[4], ce qui permet de quotienter le graphe de Cayley (à droite) du monoïde. Le calcul de l'automate qui en résulte ne nécessite pas le calcul du monoïde de transition. C'est ce qu'on appelle la construction des sous-ensembles[5]. On appelle l'automate obtenu le **déterminisé** de $\mathcal{A}$. .

DÉFINITION 1.50 *Soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate. On définit l'automate **déterminisé** de $\mathcal{A}$, $\mathcal{D} = \langle\, R,A,F,J,U \,\rangle$ par :*

$$\begin{aligned} R &= \{I \cdot u \mid u \in A^*\}, \\ J &= \{I\}, \\ U &= \{X \in R \mid X \cap T \neq \emptyset\}, \\ F &= \{(X,a,Y) \in R \times A \times R \mid Y = X \cdot a\}. \end{aligned}$$

*De façon duale, on définit l'automate **co-déterminisé** de $\mathcal{A}$, $\mathcal{C} = \langle\, S,A,G,H,V \,\rangle$ par :*

$$\begin{aligned} S &= \{u \cdot T \mid u \in A^*\}, \\ H &= \{X \in S \mid X \cap I \neq \emptyset\}, \\ V &= \{T\}, \\ G &= \{(X,a,Y) \in R \times A \times R \mid X = a \cdot Y\}. \end{aligned}$$

Le déterminisme d'un automate lui confère quelques propriétés :

PROPOSITION 1.9 *Soit $\mathcal{A}$ un automate déterministe qui reconnaît un langage $\mathcal{L}$. Alors, pour tout état p,*

4. Ce n'est pas forcément la plus grossière qui sature l'image de $\mathcal{L}$ dans $M_{\mathcal{A}}$.
5. « subset construction » en anglais, voir [23]

i) *pour tout état q distinct de p,* $\mathsf{Past}_{\mathcal{A}}(p) \cap \mathsf{Past}_{\mathcal{A}}(q) = \emptyset$.

ii) *pour tout mot u de* $\mathsf{Past}_{\mathcal{A}}(p)$, *on a* $\mathsf{Fut}_{\mathcal{A}}(p) = u^{-1}\mathcal{L}$.

Démonstration. Soit i l'état initial de $\mathcal{A}$. Le premier point est trivial, à cause du déterminisme, un mot donné ne peut mener au plus qu'en un seul état à partir de i et ne peut donc appartenir qu'au passé de cet état. Quant au second point, tout mot du langage qui accepte u comme préfixe se décompose en $u.v$, avec v qui appartient à $\mathsf{Fut}_{\mathcal{B}}(i \cdot u)$. Réciproquement, pour tout mot v de $\mathsf{Fut}_{\mathcal{B}}(i \cdot u)$, le mot $u.v$ est dans le langage. □

Voyons la relation entre les futurs des états d'un automate et ceux de son déterminisé.

LEMME 1.10 *Soit $\mathcal{A}$ un automate et $\mathcal{D}$ son déterminisé. Alors, pour tout état X de $\mathcal{D}$, on a :*

$$\mathsf{Fut}_{\mathcal{D}}(X) = \bigcup_{p \in X} \mathsf{Fut}_{\mathcal{A}}(p).$$

Démonstration. La preuve est par récurrence sur la longueur des mots. Par définition du déterminisé, le mot vide appartient au futur de X (en d'autres termes, X est final) si et seulement si il appartient au futur d'un des états de $\mathcal{A}$ qui sont dans X. Soit u un mot de longueur non nulle; u s'écrit $u = a.u'$. On obtient :

$$\begin{aligned} u \in \mathsf{Fut}_{\mathcal{D}}(X) &\iff u' \in \mathsf{Fut}_{\mathcal{D}}(X \cdot a) \\ &\iff u' \in \bigcup_{q \in X \cdot a} \mathsf{Fut}_{\mathcal{A}}(q) \\ &\iff u' \in \bigcup_{p \in X} \bigcup_{q \in p \cdot a} \mathsf{Fut}_{\mathcal{A}}(q) \\ &\iff u = a.u' \in \bigcup_{p \in X} \mathsf{Fut}_{\mathcal{A}}(p). \end{aligned}$$

□

On voit que les quotients du langage jouent un rôle particulier. Dans un automate déterministe quelconque, il se peut que deux états aient le même futur et correspondent donc au même quotient. On peut définir un automate déterministe canonique dans lequel chaque quotient correspond à un seul état :

DÉFINITION 1.51 *L'automate minimal d'un langage $\mathcal{L}$ sur A^* est l'automate déterministe $\mathcal{A}_{\mathcal{L}} = \langle Q,A,E,\{i\},T \rangle$, défini par :*

$$\begin{aligned} Q &= \{u^{-1}\mathcal{L} \mid u \in A^*\} \setminus \{\emptyset\} \\ i &= \mathcal{L} \\ T &= \{p \in Q \mid 1_{A^*} \in p\} \\ E &= \{(p,a,q) \mid a^{-1}p = q\} \end{aligned}$$

Remarque 1.13 L'automate ainsi défini est émondé. On peut définir l'automate minimal comme l'automate des quotients gauches de $\mathcal{L}$, éventuellement ensemble vide compris, ce qui donne un automate qui, le cas échéant, a un état de plus qui n'est pas co-accessible, mais l'automate est alors complet. Cependant, dans la suite, nous utiliserons essentiellement l'automate minimal émondé.

Par construction, le futur d'un état étiqueté par un quotient $u^{-1}\mathcal{L}$ est égal à ce quotient.

Il est facile de montrer que tout automate déterministe $\mathcal{A}$ émondé qui accepte un langage $\mathcal{L}$ s'envoie par quotient surjectif sur l'automate minimal de $\mathcal{L}$. Il suffit d'associer à un état p l'état de l'automate minimal qui correspond à $\mathsf{Fut}_{\mathcal{A}}(p)$ (qui est un quotient de $\mathcal{L}$).

On peut construire l'automate minimal par raffinement successif de partitions d'états sur lesquelles on examine l'action des lettres. Nous ne reviendrons pas ici sur cet algorithme dû à Moore et dont une version donnée par Hopcroft est en $O(n \log n)$. On pourra aussi consulter [50] pour une analyse en moyenne de cet algorithme sur certaines classes de langages.

On s'intéresse ici à une autre méthode, algorithmiquement moins performante mais qui nous donnent certaines informations sur des classes d'automates que nous étudierons plus tard. Elle est due à Brzozowski.

Proposition 1.11 *Le déterminisé d'un automate co-déterministe est un automate minimal.*

Démonstration. Si un automate $\mathcal{A}$ est co-déterministe, les futurs de ses états sont disjoints deux à deux. Le futur d'un état du déterminisé $\mathcal{D}$, est, par construction, l'union des futurs des états de $\mathcal{A}$ auquel il correspond. Les futurs des états de $\mathcal{D}$ sont donc distincts. L'automate $\mathcal{D}$ est par conséquence minimal. □

Corollaire 1.12 *Si un automate est à la fois déterministe et co-déterministe, il est minimal.*

La portée de ce corollaire est relativement faible, puisque nous verrons que de tels automates ne peuvent reconnaître qu'une classe restreinte de langages; ce résultat nous sera toutefois utile lorsque nous étudierons les langages à groupe ou réversibles.

Examinons maintenant le monoïde de transition de l'automate minimal d'un langage. Comme l'automate minimal est un quotient de tout automate déterministe qui reconnaît le langage, son monoïde de transition est un quotient du monoïde de transition de chacun de ces automates déterministes.

D'autre part, on a vu qu'on peut construire un automate déterministe qui reconnaît le langage à partir de n'importe quel monoïde qui reconnaît le langage. Il s'agit du graphe de Cayley, dont le monoïde de transition est le monoïde lui-même (puisque l'action des générateurs à droite sur les états est la multiplication du monoïde).

On obtient donc la proposition suivante :

Proposition 1.13 *Le monoïde de transition de l'automate minimal d'un langage sur A^*, appelé* ***monoïde syntaxique*** *du langage est un quotient de n'importe quel monoïde qui reconnaît le langage. Il existe un morphisme canonique de A^* dans ce monoïde; nous l'appellerons le* ***morphisme syntaxique****.*

Nous avons vu que dans le monoïde libre existe la notion d'image miroir d'un mot. On peut appliquer aux automates une transformation en correspondance avec cette opération.

Définition 1.52 *Soit $\mathcal{A}$ un automate dont la représentation linéaire est (λ,μ,ν). L'automate* ***transposé*** *de $\mathcal{A}$ est l'automate $\mathcal{A}^t$ défini par la représentation linéaire (ν^t,μ^t,λ^t) où λ^t et ν^t sont les vecteurs transposés respectifs de λ et ν et où μ^t est un morphisme qui, à tout mot u de A^* associe la matrice transposée de $\overline{u}\mu$.*

L'automate transposé de $\mathcal{A}$ reconnaît l'image miroir de $\mathcal{L}$ (puisque $\lambda\otimes u\mu\otimes\nu = \nu^t\otimes\overline{u}\mu^t\otimes\nu^t$).

Pour conclure, on va rappeler une construction classique sur les automates. Étant donné deux automates $\mathcal{A}$ et $\mathcal{B}$, calculer un automate qui reconnaît l'union des langages reconnus par $\mathcal{A}$ et $\mathcal{B}$ est très simple, il suffit de considérer l'union des deux automates. Calculer l'automate qui reconnaît l'intersection des langages est un peu plus compliqué. Cette opération s'appelle le produit. Il faut souligner ici que le langage que reconnaît le produit de deux automates n'est pas le produit des langages mais leur intersection.

Définition 1.53 *Soit $\mathcal{A} = \langle\, Q,A,E,I,T\,\rangle$ et $\mathcal{B} = \langle\, R,A,F,J,U\,\rangle$ deux automates. Le* ***produit*** *de $\mathcal{A}$ et de $\mathcal{B}$ est l'automate $\mathcal{C} = \langle\, Q\times R,A,G,I\times J,T\times U\,\rangle$, avec*

$$K = \{((p,q),a,(p',q')) \mid (p,a,q)\in E,(p',a,q')\in F\}.$$

En pratique, on ne construit que la partie accessible du produit. Il peut parfois être intéressant de calculer le produit d'un automate par lui-même (qu'on appelle son **carré**). En effet, chaque chemin du carré correspond à deux chemins de l'automate; ceci permet d'effectuer certaines comparaisons.

— ∘ —

4.4 Automates à multiplicité

On peut définir les automates sur A à multiplicité dans un semi-anneau $\mathbb{K}$, comme des automates sur le semi-anneau $\mathbb{K}\langle\langle A^*\rangle\rangle$. Toutefois, là encore, on s'intéressera aux automates « temps-réel », c'est-à-dire les automates dont les transitions sont étiquetées par des lettres pondérées, donc par des sommes de monômes de degré 1 :

Définition 1.54 *Un* ***automate à multiplicité*** *dans un semi-anneau $\mathbb{K}$ sur un alphabet A est un sextuplet $\mathcal{A} = \langle\, Q,A,\mathbb{K},E,I,T\,\rangle$, où Q est un ensemble fini d'états, E un sous-ensemble fini de $Q\times A\times\mathbb{K}_*\times Q$ qui sont les transitions de $\mathcal{A}$, et I et T deux vecteurs*

de $\mathbb{K}^Q$.
On suppose qu'entre deux états p *et* q*, il n'y a au plus qu'une transition par lettre de* A*. Au besoin, on remplace deux transitions* (p,a,k,q) *et* (p,a,k',q) *par la transition* $(p,a,k \oplus k',q)$.

On peut remarquer que les transitions sont étiquetées par des éléments de $\mathbb{K}_*$. En effet, ajouter une transition étiquetée par $0_{\mathbb{K}}$ ne change pas la série reconnue (ou *réalisée*).

Définition 1.55 *L'****automate sous-jacent*** *d'un automate* $\mathcal{A} = \langle\, Q,A,\mathbb{K},E,I,T\,\rangle$ *à multiplicité est un automate* $\mathcal{B} = \langle\, Q,A,F,J,U\,\rangle$ *(sans multiplicité) tel que*

$$\begin{aligned} F =& \{(p,a,q) \mid \exists k \in \mathbb{K}_*, (p,a,k,q) \in E\}, \\ J =& \{p \mid I_p \neq 0_{\mathbb{K}}\}, \\ U =& \{p \mid T_p \neq 0_{\mathbb{K}}\}. \end{aligned}$$

On dira qu'un automate à multiplicité est déterministe, co-déterministe ou complet si son automate sous-jacent l'est.

On peut définir une représentation linéaire pour les automates avec multiplicité de la même manière que pour les automates « classiques ».

Définition 1.56 *La* ***représentation linéaire*** *d'un automate* $\mathcal{A} = \langle\, Q,A,\mathbb{K},E,I,T\,\rangle$ *est un triplet* (λ,μ,ν)*, où* λ *et* ν *sont respectivement des vecteurs ligne et colonne de* $\mathbb{K}^Q$*, et* μ *un morphisme de* A^* *dans* $\mathbb{K}^{Q\times Q}$ *tels que :*

$$\forall p \in Q,\ \lambda_p = I_p\,, \quad \nu_p = T_p,$$
$$\forall p,q \in Q,\ \forall a \in A,\ (a\mu)_{p,q} = k \Leftrightarrow (p,a,k,q) \in E.$$

Si (p,a,q) *n'est pas une transition de l'automate sous-jacent de* $\mathcal{A}$*,* $(a\mu)_{p,q} = 0_{\mathbb{K}}$.

Remarque 1.14 Si α est réalisé par un automate dont la représentation linéaire est (λ,μ,ν), pour tout mot u de A^*, la série $u^{-1}\alpha$ est réalisée par un automate dont la représentation linéaire est $(\lambda \otimes u\mu,\mu,\nu)$.

Le problème de la déterminisation des automates à multiplicité est de loin beaucoup plus délicat que dans le cas des automates sans multiplicité. On a en effet, un lien similaire entre les futurs des états d'un automate déterministe et les quotients de la série.

Proposition 1.14 *Soit* $\mathcal{A}$ *un automate déterministe à multiplicité dans* $\mathbb{K}$ *qui réalise une série* s*. Alors, pour tout état* p *de* $\mathcal{A}$*, pour tout mot* u *de* $\mathsf{Past}_{\mathcal{A}}(p)$*,*

$$\langle \mathsf{Past}_{\mathcal{A}}(p),u\rangle \otimes \mathsf{Fut}_{\mathcal{A}}(p) = u^{-1}s.$$

Démonstration. Comme $\mathcal{A}$ est déterministe, pour tout mot dont u est un préfixe, et qu'on peut écrire $u.v$,

$$\langle s,u.v\rangle = \langle \mathsf{Past}_{\mathcal{A}}(p),u\rangle \otimes \langle \mathsf{Fut}_{\mathcal{A}}(p),v\rangle.$$

On obtient donc directement le résultat. □

On le voit, chaque quotient d'une série réalisée par un automate déterministe est multiple (dans $\mathbb{K}$) d'une série prise dans un ensemble fini. Toute série rationnelle ne respecte pas cette propriété. Par exemple, la série $a^* + (2a)^*$ sur $\mathbb{N}$ a pour quotients $\{a^* + 2^n(2a)^* \mid n \in \mathbb{N}\}$. On peut vérifier qu'il n'y a pas d'ensemble fini de séries tel que chacun de ces quotients est un multiple d'un élément de cet ensemble. Les séries réalisables par un automate fini sont donc un sous-ensemble strict des séries rationnelles.

Définition 1.57 *Soit s un série de $\mathbb{K}\mathrm{Rat}\,A^*$. On dit que s est une* ***série séquentielle***[6] *si s peut être réalisée par un automate déterministe.*

On l'a vu, les quotients d'une série séquentielle appartiennent à un cône finiment engendré et ce n'est en général pas le cas des séries rationnelles. Toutefois, les quotients des séries rationnelles respectent la propriété suivante qui est caractéristique :

Théorème 1.4 *Une série formelle sur A^* à coefficients dans $\mathbb{K}$ est rationnelle si et seulement si elle appartient à un $\mathbb{K}$-semi-module de type fini clos par quotient (par rapport aux mots de A^*).*

Démonstration. Soit s une série formelle et $g = (g_1, g_2, \dots, g_n)$ un système générateur d'un semi-module clos par quotient contenant s. On considère g comme un vecteur colonne. Comme il s'agit d'un système générateur, il existe un vecteur ligne λ de $\mathbb{K}^n$ et une application μ de A^* dans $\mathbb{K}^{n\times n}$ tels que :

$$\begin{aligned} s &= \lambda \otimes g \\ \forall u \in A^*, u^{-1}g &= (u\mu) \otimes g \end{aligned}$$

Comme le quotient est une action à droite des mots sur les séries, μ est un morphisme. On pose $\nu = (\nu_1, \dots, \nu_n)$, vecteur colonne tel que, pour tout i de $[1;n]$, $g_i = \langle s_1, 1_{A^*}\rangle$. Le triplet (λ, μ, ν) est la représentation linéaire d'un automate. Pour tout mot u,

$$\begin{aligned} \langle s,u\rangle &= \langle \lambda \otimes g, u\rangle \\ &= \lambda \otimes \langle g,u\rangle \\ &= \lambda \otimes \langle u^{-1}g, 1_{A^*}\rangle \\ &= \lambda \otimes (u\mu) \otimes \nu \end{aligned}$$

L'automate ainsi construit reconnaît s qui est donc rationnelle.

Considérons à présent une série rationnelle s réalisée par un automate $\mathcal{A} = \langle\, Q, A, \mathbb{K}, E, I, T\,\rangle$ à n états. Pour tout mot u de A^*,quel que soit le mot w acceptant u comme préfixe ($w = u.v$), on a :

$$\langle u^{-1}s, v\rangle = \langle s,w\rangle = \bigoplus_{q\in Q} \langle \mathsf{Past}_{\mathcal{A}}(p), u\rangle \otimes \langle \mathsf{Fut}_{\mathcal{A}}(p), v\rangle.$$

6. On appelle série séquentielle ce qui était traditionnellement appelé série sous-séquentielle.

Donc, on a la relation

$$u^{-1}s = \bigoplus_{q \in Q} \langle \mathsf{Past}_{\mathcal{A}}(p), u \rangle \otimes \mathsf{Fut}_{\mathcal{A}}(p).$$

Les futurs des états de l'automate engendrent donc un $\mathbb{K}$-semi-module de type fini clos par quotient et qui contient s. □

On peut revenir aux séries séquentielles et en donner une caractérisation :

THÉORÈME 1.5 *Une série formelle sur A^* à coefficients dans $\mathbb{K}$ est rationnelle si et seulement si ses quotients appartiennent à un $\mathbb{K}$-cône finiment engendré.*

Démonstration. Le fait qu'une série séquentielle respecte cette propriété est un corollaire immédiat de la proposition 1.14.

Si les quotients d'une série formelle appartiennent à un cône finiment engendré par une famille $(g_1, g_2, \ldots, g_n)$, ils appartiennent au sous-module engendré par les mêmes générateurs, qui est stable pour l'opération quotient, et la série est donc rationnelle. Dire que les quotients appartiennent à un cône signifie que pour toute lettre a, pour tout i de $[1;n]$, il existe j dans $[1;n]$ et $k_{i,j}$ dans $\mathbb{K}$ tel que $a^{-1}g_i = k_{i,j}g_j$. On peut donc trouver un morphisme μ de A^* dans $\mathbb{K}^{n \times n}$ tel que, pour toute lettre a de A, chaque ligne de $a\mu$ n'a qu'un coefficient non nul. Comme, de même, la série s est proportionnelle à l'un des g_i, l'automate correspondant à la représentation linéaire (λ, μ, ν) ainsi construite est déterministe. □

De même qu'on a défini l'action des lettres sur les états d'un automate, dans le cas d'un automate avec multiplicité, on peut définir la **fonction de production** d'un mot sur un état :

DÉFINITION 1.58 *Soit $\mathcal{A} = \langle Q, A, \mathbb{K}, E, I, T \rangle$ un automate à multiplicité et (λ, μ, ν) sa représentation linéaire. Pour tout état p de Q, on pose $\chi^{(p)}$ le vecteur de $\mathbb{K}^Q$ caractéristique de p ($\chi^{(p)}_p = 1_{\mathbb{K}}$ et pour tout $q \neq p$, $\chi^{(p)}_q = 0_{\mathbb{K}}$). Pour tout mot u, on définit la* ***production*** *de u à partir d'un état p par :*

$$p * u = \chi^{(p)}.(u\mu).$$

Si X est un sous-ensemble de Q,

$$X * a = \bigoplus_{p \in X} p * a.$$

Parmi les automates à multiplicité, citons deux familles particulières. Premièrement, ceux dont les coefficients appartiennent à $\mathrm{Rat}\, B^*$, où B est un alphabet fini. Il s'agit alors de **transducteurs**; ils ont été largement étudiés dans le passé. C'est le même type de questions que celles qui ont été résolues pour les transducteurs que nous allons nous poser au sujet d'une seconde famille, celle des automates à coefficients dans un semi-anneau $(\mathsf{max}, +)$.

— ∘ —

Chapitre 2

Automate universel

En 1971, J.H. Conway [19] a défini la *matrice des facteurs* d'un langage rationnel. Son but était de donner un moyen de décider si un langage rationnel $\mathcal{L}$ appartenait à la clôture rationnelle d'une famille finie de langages rationnels $(\mathcal{L}_i)_{i\in I}$. Depuis, cette construction a été plusieurs fois retrouvée. O. Carton [12] a montré que cette construction pouvait être faite dans n'importe quel monoïde. J. Sakarovitch [55] a mis en lumière une autre propriété de cette matrice des facteurs (facilement transformable en automate) : il existe un morphisme de tout automate qui reconnaît le langage dans cet « automate des facteurs ». Cette propriété incite à donner à cet objet le nom d'automate universel que nous employons ici. Elle avait été exploitée dans [4] pour le problème de l'obtention d'un automate minimal non-deterministe. On trouvera dans [55] une étude complète des propriétés de l'automate universel.

Toutefois, par souci de cohérence et afin de fixer certaines conventions, nous reprenons ces résultats dans la première partie de ce chapitre.

Le paragraphe 1 présente des définitions et propriétés qui figurent déjà, sauf mention explicite, ailleurs ([19, 4, 12, 55]). Elles sont évidemment nécessaires à la compréhension des paragraphes suivants. Les résultats principaux en sont que, pour tout langage reconnaissable, il existe un automate fini qui reconnaît le langage et dont les états sont caractérisés par le couple formé de leur passé et de leur futur qui représente une *factorisation* maximale du langage. Cet automate doit son qualificatif d'automate universel au fait que tout automate émondé qui reconnaît le langage s'y envoie par morphisme.

Le paragraphe 2 présente une construction de l'automate universel à partir de l'automate minimal, en évitant le calcul du monoïde syntaxique, ce qui rend la complexité de la construction dépendante uniquement de la taille du résultat.

La paragraphe suivant présente l'*écorché* de l'automate universel. Il s'agit d'une représentation qui permet de faire apparaître les propriétés de l'automate universel en codant à l'aide de transitions spontanées un certain nombre de transitions « redondantes » de l'automate universel. D'un point de vue pratique, ceci permet de diminuer le nombre de transitions à stocker, mais aussi de pouvoir dessiner des automates universels relativement gros.

Le dernier paragraphe montre comment établir un lien entre l'automate universel d'un langage et un automate donné qui reconnaît ce même langage, ce qui permettra ensuite de retrouver dans le premier des propriétés héritées du second. Ceci est en fait la clé des raisonnements que nous allons effectuer par la suite. L'automate universel peut en effet être construit, d'une part, à partir d'un automate particulier (qu'on ne sait *a priori* pas construire) duquel il hérite certaines propriétés et, d'autre part, à partir de l'automate minimal reconnaissant le même langage, ce qui permet une construction effective.

— ∘ —

1 Définitions et propriétés de l'automate universel

1.1 Automate universel dans un monoïde quelconque

Factorisations d'un ensemble. La définition de l'automate universel, basée sur la notion de facteurs, est possible pour n'importe quel sous-ensemble $\mathcal{L}$ de n'importe quel monoïde N.

Naturellement, le nombre d'états de l'automate universel d'un ensemble quelconque n'est pas nécessairement fini, ni même dénombrable.

DÉFINITION 2.1 *Une* ***sous-factorisation*** *d'un sous-ensemble $\mathcal{L}$ de N est un k-uplet de sous-ensembles non vides de N, $(R_1,R_2,\ldots,R_k)$ appartenant à $(\mathcal{P}(N)\setminus\{\emptyset\})^k$, tel que $R_1.R_2\ldots R_k$ est inclus dans $\mathcal{L}$. Les sous-factorisations sont partiellement ordonnées par inclusion. Une* ***factorisation*** *de $\mathcal{L}$ est une sous-factorisation maximale. Si une factorisation de $\mathcal{L}$ est un couple (L,R), on dit que L est un* ***facteur gauche*** *et R un* ***facteur droit*** *de $\mathcal{L}$.*

REMARQUE 2.1 Dans ce qui suit, on considérera les factorisations qui sont des couples, sauf mention explicite. Puisqu'elle est maximale, une factorisation (L,R) de $\mathcal{L}$ est entièrement déterminée par la donnée de L (ou de R) :

$$L = \mathsf{max}\{X \subseteq N \mid X.R \subseteq \mathcal{L}\}\,, \quad R = \mathsf{max}\{X \subseteq N \mid L.X \subseteq \mathcal{L}\}.$$

EXEMPLE 1.7 Les factorisations du langage $\mathcal{L}_1 = A^*abA^*$ sont

$$(A^*,A^*abA^*), \quad (A^*aA^*,A^*bA^*) \quad \text{et } (A^*abA^*,A^*).$$

En effet, pour toute sous-factorisation (L,R) de $\mathcal{L}_1$, s'il existe un mot de L ne contenant pas de a, tout mot de R contient un facteur ab et s'il existe un mot de R ne contenant pas de b, tout mot de L contient un facteur ab. Il n'est pas difficile de voir que les trois factorisations données sont bien maximales.

EXEMPLE 7.2 Soit $N_3 = (\mathbb{Z}/3\mathbb{Z},+)$ et $\mathcal{L}_3 = \{1,2\}$. La figure 1 représente la table du monoïde; les éléments de $\mathcal{L}_3$ sont encadrés, les rectangles indiquent les factorisations. Cette façon « géométrique » a été étudiée dans [20].

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

FIG. 1 – *La table du groupe $\mathbb{Z}/3\mathbb{Z}$ et les factorisations de $\mathcal{L}_3$.*

Les factorisations sont donc les suivantes :

$$(\{0\},\{1,2\}),\quad (\{1\},\{0,1\}),\quad (\{2\},\{0,2\}),$$
$$(\{1,2\},\{0\}),\quad (\{0,1\},\{1\}),\quad (\{0,2\},\{2\}).$$

EXEMPLE 8.1 Soit a^* le monoïde librement engendré par a et $\mathcal{L}_2 = a^+$ l'ensemble des mots de A^* qui contiennent au moins une lettre. Les factorisations de $\mathcal{L}_2$ sont (a^*,a^+) et (a^+,a^*).

EXEMPLE 9 Soit N_5 le monoïde des fonctions de $\mathbb{R}$ dans $\mathbb{R}$ munies de la composition et $\mathcal{L}_5$ le singleton contenant la fonction identité. Les factorisations de $\mathcal{L}_5$ sont les couples maximaux (L,R) tels que

- L est un ensemble non vide d'applications injectives telles que quel que soit α et β appartenant à L, quel que soit (x,y) dans $\mathbb{R}^2$, $x\alpha = y\beta$ entraîne $x = y$.
- R est l'ensemble des fonctions réalisant l'inverse de α sur $\mathsf{Im}\,\alpha$, pour tout α dans L.

Le couple suivant est un exemple de factorisation de l'identité :

$$(\{\mathsf{arctg},\mathsf{arctg}+\pi\},\{\alpha \mid X \subseteq \mathsf{Supp}(\alpha) \subseteq \mathbb{R},\forall x \in X, x\alpha = \mathsf{tg}(x)\}).$$

Le nombre de factorisations n'est ici pas dénombrable. En effet, chaque bijection α de $\mathbb{R}$ sur lui-même induit une factorisation (maximale) de l'identité : $(\{\alpha\},\{\alpha^{-1}\})$. On obtient donc ainsi un sous-ensemble non dénombrable des factorisations de l'identité.

— ◦ —

Définition de l'automate universel. Nous allons construire l'automate dont les états sont les factorisations de l'ensemble à reconnaître, qui a un maximum de transitions, et qui respecte les propriétés suivantes :

- Un état (L,R) est initial (*resp.* final) si et seulement si 1_N appartient à L (*resp.* à R).
- Un élément x de N étiquette une transition entre deux états (L,R) et (L',R') si et seulement si $L.x.R'$ est inclus dans $\mathcal{L}$.

Définition 2.2 *Soit $\mathcal{L}$ un sous-ensemble de N et Q l'ensemble de ses factorisations. L'**automate universel de** $\mathcal{L}$ est $\mathcal{U}_{\mathcal{L}} = \langle\, Q,N,E,I,T \,\rangle$, avec:*

$$\begin{aligned} I &= \{(L,R) \in Q \mid 1_N \in L\},\\ T &= \{(L,R) \in Q \mid 1_N \in R\},\\ E &= \{((L,R),X,(L',R')) \in Q \times \mathcal{P}(N) \times Q \mid X = \{x \in N \mid L.x.R' \subseteq \mathcal{L}\}\}. \end{aligned}$$

Remarque 2.2 Dans la définition précédente, la condition $L.x.R' \subseteq \mathcal{L}$ est équivalente, à cause de la maximalité des factorisations à l'une des conditions $L.x \subseteq L'$ ou $x.R' \subseteq R$. De même, la condition $1_N \in L$ est équivalente à $R \subseteq \mathcal{L}$ et $1_N \in R$ est équivalent à $L \subseteq \mathcal{L}$.

Exemple 1.8 L'automate universel de $\mathcal{L}_1$ a trois états (les trois factorisations de $\mathcal{L}_1$ présentées exemple 1.7 page 46). Il est représenté figure 2.

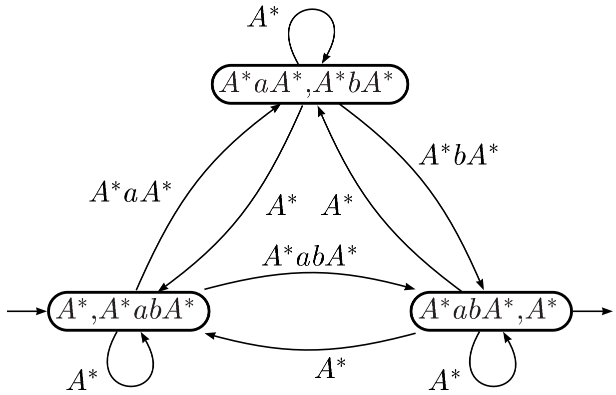


Fig. 2 – *L'automate universel de $\mathcal{L}_1$.*

Exemple 8.2 L'automate universel de $\mathcal{L}_2$ a deux états. Il est représenté figure 3.

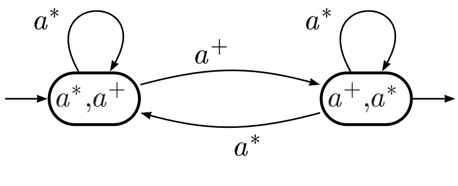


Fig. 3 – *L'automate universel de $\mathcal{L}_2$.*

Exemple 7.3 L'automate universel de $\{1,2\}$, dans $\mathbb{Z}/3\mathbb{Z}$, a six états (les six factorisations). Il est représenté figure 4. Sur la figure, chaque état est étiqueté par le facteur gauche de la factorisation à laquelle il correspond.

Parmi toutes les factorisations, il y en a deux qui vont jouer un rôle particulier.

Définition 2.3 *Soit $\mathcal{L}$ un sous-ensemble de N. Soit $L_i = \max\{X \subseteq N \mid X.\mathcal{L} \subseteq \mathcal{L}\}$ et $R_t = \max\{X \subseteq N \mid \mathcal{L}.X \subseteq \mathcal{L}\}$. Les couples $(L_i,\mathcal{L})$ et $(\mathcal{L},R_t)$ sont des factorisations*

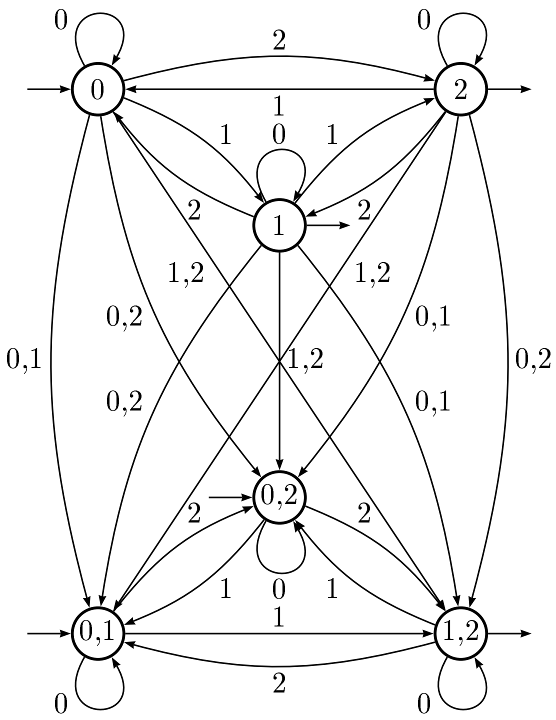


Fig. 4 – *L'automate universel de* $\mathcal{L}_3$.

et sont appelées respectivement ***factorisation initiale*** *et* ***factorisation terminale****. La factorisation initiale* (resp. *terminale) est un état initial* (resp. *final) de l'automate universel.*

— ∘ —

Propriétés. On vérifie à présent que l'automate construit a les propriétés voulues, c'est-à-dire qu'il reconnaît ce qu'on veut, bien sûr, mais aussi que les factorisations représentent effectivement le passé et le futur des états. C'est l'objet des deux propositions suivantes.

Proposition 2.1 *L'automate* $\mathcal{U}_\mathcal{L}$ *reconnaît l'ensemble* $\mathcal{L}$.

Démonstration. Soit $p_i = (L_i,\mathcal{L})$ et $p_t = (\mathcal{L},R_t)$ les factorisations initiale et terminale. $L_i.\mathcal{L}.R_t$ est inclus dans $\mathcal{L}.R_t$, donc dans $\mathcal{L}$. Le triplet $((L_i,\mathcal{L}),\mathcal{L},(\mathcal{L},R_t))$ est donc une transition de E, et tout mot de $\mathcal{L}$ est accepté par $\mathcal{U}_\mathcal{L}$.

Réciproquement, si u est reconnu par $\mathcal{U}_\mathcal{L}$, il existe des factorisations $(L_0,R_0), \ldots , (L_k,R_k)$ de $\mathcal{L}$ et $(u_1, \ldots ,u_k)$ une factorisation de u en k éléments tels que $L_{i-1}.u_i.R_i$ est inclus dans $\mathcal{L}$ et 1_N appartient à L_0 et à R_k. Alors $L_0.u.R_k$ est inclus dans $\mathcal{L}$, donc u est un élément de $\mathcal{L}$. □

Proposition 2.2 *Pour chaque état* $p = (L,R)$ *de l'automate* $\mathcal{U}_\mathcal{L}$*, les facteurs* L *et* R *sont* *respectivement égaux au passé et au futur de p dans $\mathcal{U}_\mathcal{L}$:*

$$L = \mathsf{Past}_{\mathcal{U}_\mathcal{L}}(L,R)\ , \quad R = \mathsf{Fut}_{\mathcal{U}_\mathcal{L}}(L,R).$$

Démonstration. Soit $p_i = (L_i,\mathcal{L})$ et $p_t = (\mathcal{L},R_t)$ les factorisations initiale et terminale. Pour toute factorisation $p = (L,R)$, $L.R$ est inclus dans $\mathcal{L}$, donc $L_i.L.R$ aussi et (p_i,L,p) est une transition de $\mathcal{U}_\mathcal{L}$. Donc L est inclus dans le passé de p. De même, R est inclus dans le futur de p. Comme (L,R) est une factorisation (maximale) de $\mathcal{L}$ qui est l'ensemble reconnu par l'automate, la factorisation (L,R) est égale à la factorisation $(\mathsf{Past}_{\mathcal{U}_\mathcal{L}}(p),\mathsf{Fut}_{\mathcal{U}_\mathcal{L}}(p))$. □

COROLLAIRE 2.3 *Tout automate universel est émondé.*

Démonstration. Chaque état correspond à une factorisation dont les facteurs représentent le passé et le futur de l'état. Ces facteurs ne sont pas l'ensemble vide, l'état est donc accessible et co-accessible. □

Nous allons maintenant voir ce qui fait de l'automate que nous avons défini un automate « universel » pour l'ensemble reconnu.

PROPOSITION 2.4 *Soit $\mathcal{A}$ un automate émondé reconnaissant un sous-ensemble de $\mathcal{L}$. Alors, il existe un morphisme φ de $\mathcal{A}$ dans $\mathcal{U}_\mathcal{L}$. De plus, $\mathcal{U}_\mathcal{L}$ est le plus petit*[1] *automate reconnaissant $\mathcal{L}$ ayant cette propriété.*

Démonstration. Soit p un état de $\mathcal{A}$. Le passé et le futur de p forment une sous-factorisation de $\mathcal{L}$. On pose[2] :

$$R_p = \mathsf{max}\{X \subseteq N \mid \mathsf{Past}_\mathcal{A}(p).X \subseteq \mathcal{L}\} = \mathsf{Past}_\mathcal{A}(p)^{-1}\mathcal{L} \quad \text{et } L_p = \mathsf{max}\{X \subseteq N \mid X.R_p \subseteq \mathcal{L}\}.$$

Il s'agit bien d'une factorisation. On vérifie que si p est initial ou final, (L_p,R_p) aussi.
Si (p,u,q) est une transition de $\mathcal{A}$, comme $\mathsf{Past}_\mathcal{A}(p).u$ est inclus dans le passé de q, $\mathsf{Past}_\mathcal{A}(p).u.R_q$ est inclus dans $\mathcal{L}$, donc $u.R_q$ est inclus dans R_p et $L_p.u.R_q$ est inclus dans $\mathcal{L}$. L'élément u fait donc partie de l'étiquette de la transition entre (L_p,R_p) et (L_q,R_q).
Supposons qu'il existe un automate $\mathcal{A}'$ dans lequel tout automate reconnaissant $\mathcal{L}$ s'envoie par morphisme. S'il existe un morphisme non injectif de $\mathcal{U}_\mathcal{L}$ dans $\mathcal{A}'$, deux états p et q de $\mathcal{U}_\mathcal{L}$ ont une même image, dont le passé contient $L_p \cup L_q$ et le futur $R_p \cup R_q$. Comme (L_p,R_p) et (L_q,R_q) sont des factorisations, $(L_p \cup L_q).(R_p \cup R_q)$ n'appartient pas à $\mathcal{L}$ et $\mathcal{A}'$ reconnaît strictement plus que l'ensemble $\mathcal{L}$. Donc tout morphisme de $\mathcal{U}_\mathcal{L}$ dans $\mathcal{A}'$ est injectif. Le cardinal de l'ensemble des états et de l'ensemble des transitions de $\mathcal{A}'$ est donc supérieur ou égal à celui de $\mathcal{U}_\mathcal{L}$. □

L'automate universel peut, d'après ce qu'on en a dit, être infini. Nous allons voir sous quelle condition il est fini.

— ∘ —

1. Comme on n'a fait aucune supposition sur l'ensemble reconnu, l'automate est éventuellement infini. Cet automate est le « plus petit » dans le sens où il est inclus dans tout automate ayant cette propriété.

2. On voit que cette définition n'est pas symétrique. On peut ainsi définir deux morphismes canoniques (non nécessairement distincts) de $\mathcal{A}$ dans $\mathcal{U}_\mathcal{L}$. Ce ne sont d'ailleurs pas nécessairement les seuls morphismes de $\mathcal{A}$ dans $\mathcal{U}_\mathcal{L}$ (voir [55]).

1.2 Automate universel d'un ensemble reconnaissable

On suppose le sous-ensemble $\mathcal{L}$ de N reconnaissable. Ceci signifie qu'il existe un monoïde M fini, un morphisme φ de N dans M et P inclus dans M tel que $\mathcal{L} = P\varphi^{-1}$.

PROPOSITION 2.5 *L'automate universel d'un ensemble a un nombre fini d'états si et seulement si cet ensemble est reconnaissable.*

Démonstration. Supposons l'ensemble $\mathcal{L}$ reconnaissable. Soit (L,R) une factorisation de $\mathcal{L}$. On a :

$$L.R \subseteq \mathcal{L}$$
$$L\varphi.R\varphi \subseteq \mathcal{L}\varphi$$
$$L\varphi\varphi^{-1}.R\varphi\varphi^{-1} \subseteq \mathcal{L}\varphi\varphi^{-1} = \mathcal{L}$$

Comme L est inclus dans $L\varphi\varphi^{-1}$, et R dans $R\varphi\varphi^{-1}$ et puisque (L,R) est une factorisation, $L = L\varphi\varphi^{-1}$ et $R = R\varphi\varphi^{-1}$. Les factorisations de $\mathcal{L}$ dans N sont donc les images inverses des factorisations de P dans M qui est fini. Elles sont donc en nombre fini.

Réciproquement, supposons le nombre de factorisations finies. A tout élément x du monoïde N, on associe la factorisation (L_x,R_x), telle que $R_x = \{y \in N \mid x.y \in \mathcal{L}\} = x^{-1}\mathcal{L}$. L'ensemble des factorisations munies de la loi induite par cette application est un monoïde fini qui reconnaît $\mathcal{L}$. En effet, x est un élément de $\mathcal{L}$ si et seulement si R_x contient 1_N. □

La proposition précédente assure que le nombre d'états de l'automate universel d'un ensemble reconnaissable est fini. Toutefois, pour que l'on puisse parler d'automate fini, il faut qu'on puisse donner une description finie, non seulement des états, mais aussi des transitions. La proposition suivante répond à cette contrainte.

PROPOSITION 2.6 *Les étiquettes des transitions de l'automate universel d'un ensemble reconnaissable sont reconnaissables.*

Démonstration. L'argument est le même que dans la preuve précédente; si le triplet $((L_p,R_p),X,(L_q,R_q))$ est une transition,

$$L_p\varphi\varphi^{-1}.X\varphi\varphi^{-1}.R_q\varphi\varphi^{-1} = L_p.X\varphi\varphi^{-1}.R_q \subseteq \mathcal{L}.$$

Par maximalité de X, on obtient $X = X\varphi\varphi^{-1}$ qui est donc reconnaissable. □

EXEMPLE 1.9 Soit M_1 le monoïde présenté exemple 1.4 page 22. Soit φ le morphisme de A^* dans M_1 défini par $a\varphi = x$ et $b\varphi = y$. Comme nous l'avons déjà vu page 31, le langage $\mathcal{L}_1$ est l'image inverse de 0_{M_1} par ce morphisme. Les factorisations de $\mathcal{L}_1$ dans A^* correspondent donc aux factorisations de 0_{M_1} dans M_1. Ce sont $(M_1,0)$, $(\{x,z,0\},\{y,z,0\})$ et $(0,M_1)$ qui correspondent bien aux factorisations données page 48. L'automate universel de 0 dans M_1 est présenté figure 5. Si on applique φ^{-1} à chaque étiquette de l'automate, on retrouve l'automate universel de $\mathcal{L}_1$ dessiné figure 2.

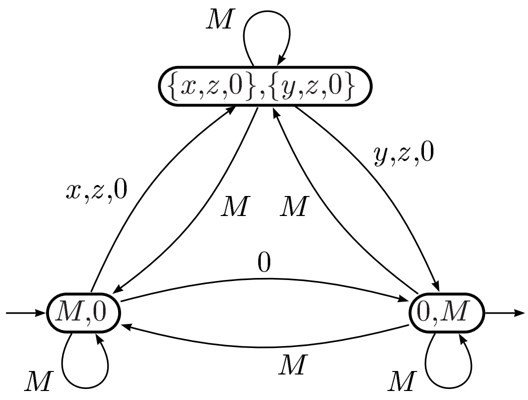


FIG. 5 – *L'automate universel de* 0 *dans* M_1.

EXEMPLE 7.4 Soit $A = \{a,b\}$ et $\mathcal{L}'_3 = \{u \in A^* \mid |u|_a \neq |u|_b \mod 3\}$. Le monoïde syntaxique de ce langage est $N_3 = \mathbb{Z}/3\mathbb{Z}$, avec φ_3 le morphisme de A^* sur N_3 défini par $a\varphi_3 = 1$ et $b\varphi_3 = 2$. L'image de $\mathcal{L}'_3$ par φ_3 est $\mathcal{L}_3 = \{1,2\}$. L'automate minimal, qui est dans ce cas le graphe de Cayley du monoïde syntaxique, est représenté figure 6.

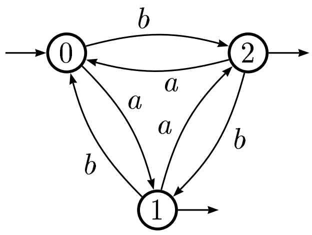


FIG. 6 – *L'automate minimal de* $\mathcal{L}'_3$.

On peut calculer les images inverses des éléments de N_3 par φ :

$$\begin{aligned}
0\varphi^{-1} &= \mathcal{L}_{3,0} = (ab + (a^2 + b)(ba)^*(b^2 + a))^* \\
1\varphi^{-1} &= \mathcal{L}_{3,1} = (ab + (a^2 + b)(ba)^*(b^2 + a))^*(a + b^2)(ab)^* \\
2\varphi^{-1} &= \mathcal{L}_{3,2} = (ab + (a^2 + b)(ba)^*(b^2 + a))^*(a^2 + b)(ba)^*
\end{aligned}$$

Pour obtenir les factorisations de $\mathcal{L}'_3$, il suffit donc de remplacer chaque élément de N_3 par son image inverse selon φ dans les factorisations données dans l'exemple 7.2 (page 46). De même, pour obtenir l'automate universel de $\mathcal{L}'_3$, il suffit de remplacer chaque étiquette de l'automate représenté figure 4 par son image inverse selon φ.

— ◦ —

1.3 Automate universel et générateurs du monoïde

Les transitions de l'automate universel d'un langage de A^* défini plus haut sont étiquetées par des langages. Lorsque l'on travaille avec un monoïde finiment engendré (en particulier dans le monoïde libre A^*), il est plus habituel de traiter des automates dont les étiquettes sont étiquetées par des lettres. On va voir qu'on peut définir un automate universel sur un alphabet.

Définition 2.4 *Soit $\mathcal{L}$ un sous-ensemble d'un monoïde N engendré par les éléments d'un ensemble A et Q l'ensemble des factorisations de $\mathcal{L}$ dans N. L'automate A-universel de $\mathcal{L}$ est $\mathcal{U}_{\mathcal{L}} = \langle Q,A,E,I,T \rangle$, avec :*

$$\begin{aligned} I =& \{(L,R) \in Q \mid 1_N \in L\}, \\ T =& \{(L,R) \in Q \mid 1_N \in R\}, \\ E =& \{((L,R),a,(L',R')) \in Q \times A \times Q \mid L.a.R' \subseteq \mathcal{L}\}. \end{aligned}$$

Exemple 8.3 L'automate $\{a\}$-universel de $\mathcal{L}_2 = a^+$ est représenté figure 7.

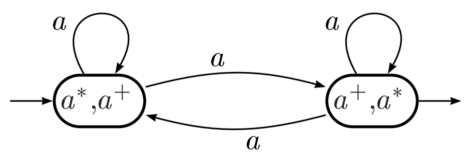


Fig. 7 – *L'automate $\{a\}$-universel de $\mathcal{L}_2$.*

Exemple 7.5 L'automate $\{1\}$-universel de $\{1,2\}$, dans $\mathbb{Z}/3\mathbb{Z}$ est représenté figure 8.

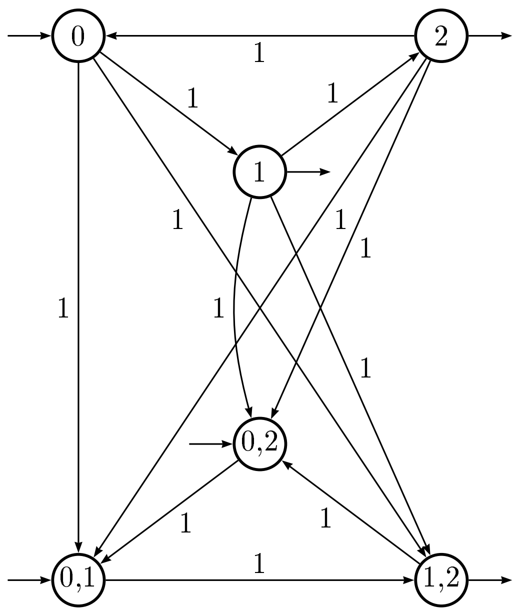


Fig. 8 – *L'automate $\{1\}$-universel de $\mathcal{L}_3$.*

Les états de l'automate A-universel sont les mêmes que ceux de l'automate universel. Les étiquettes des transitions sont exactement l'intersection des étiquettes de l'automate universel avec A. Nous allons voir que ceci ne diminue pas la puissance de l'objet.

Proposition 2.7 *L'automate A-universel de $\mathcal{L}$ reconnaît $\mathcal{L}$.*

Démonstration. Il est clair que l'automate reconnaît au plus $\mathcal{L}$ puisqu'il est une projection de l'automate universel . D'autre part, tout automate étiqueté par des éléments de A qui reconnaît cet ensemble s'envoie dans l'automate A-universel par morphisme. Celui-ci reconnaît donc l'ensemble $\mathcal{L}$. □

Remarque. Lorsqu'il n'y a pas d'ambiguïté, on appellera l'automate A-universel automate universel. C'est le cas, par exemple, lorsqu'on travaille avec le monoïde libre engendré par un alphabet fini A.

— ∘ —

1.4 Automate universel d'un langage rationnel de A^*

Un langage rationnel de A^* est, d'après le théorème de Kleene, reconnaissable. Il admet donc un automate universel fini et c'est la combinaison des deux paragraphes précédents qui forme le cadre dans lequel nous allons travailler. Nous utiliserons en effet l'automate A-universel des langages étudiés.

L'automate universel sera donc, dans ce cas, un automate fini dont les étiquettes sont des lettres.

Exemple 1.10 L'automate (A)-universel du langage $\mathcal{L}_1$ est obtenu à partir de l'automate de la figure 2 en remplaçant chaque étiquette par son intersection avec l'alphabet. Le résultat est présenté figure 9.

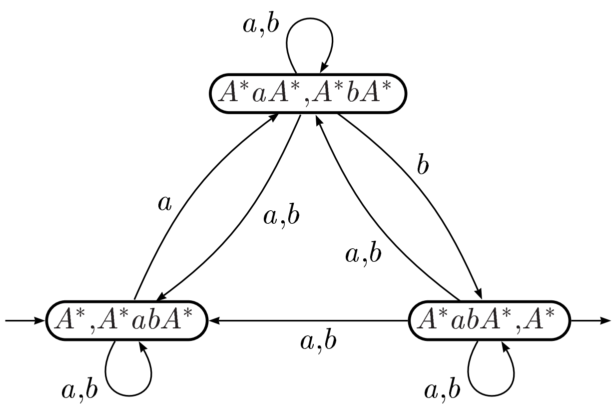


Fig. 9 – *L'automate (A)-universel de $\mathcal{L}_1$.*

Exemple 7.6 L'automate (A)-universel du langage $\mathcal{L}'_3$ est obtenu en prenant l'intersection des langages qui étiquettent l'automate universel avec l'alphabet : $\mathcal{L}_{3,0} \cap A = \emptyset$, $\mathcal{L}_{3,1} \cap A = \{a\}$, $\mathcal{L}_{3,2} \cap A = \{b\}$. L'automate universel de $\mathcal{L}'_3$ présenté figure 10 est donc obtenu à partir de la figure 4 en remplaçant 1 par a, 2 par b et en supprimant les transitions étiquetées par 0.

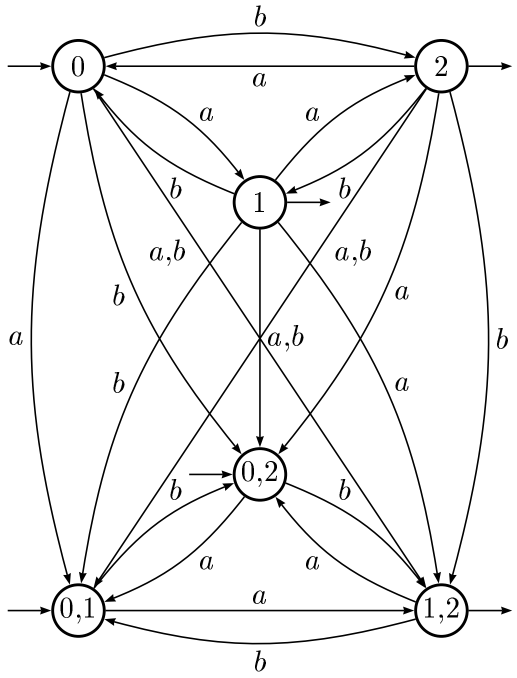


Fig. 10 – *L'automate universel de $\mathcal{L}'_3$.*

On peut remarquer que la définition de l'automate universel est symétrique, il n'y a donc pas d'automate « co-universel ». En d'autres termes, on a la proposition suivante :

Proposition 2.8 *Soit $\mathcal{L}$ un langage rationnel de A^*. L'automate universel $\mathcal{U}_{\overline{\mathcal{L}}}$ de l'image miroir de $\mathcal{L}$ est l'automate transposé de l'automate universel $\mathcal{U}_{\mathcal{L}}$ de $\mathcal{L}$.*

Démonstration. Soit (L,R) et (L',R') deux factorisations de $\mathcal{L}$. Alors $(\overline{R},\overline{L})$ et $(\overline{R'},\overline{L'})$ sont des factorisations de $\overline{\mathcal{L}}$.
Si (L,R) est un état initial de $\mathcal{U}_{\mathcal{L}}$, le mot vide 1_{A^*} appartient à L donc à $\overline{L}$ et $(\overline{R},\overline{L})$ est un état final de $\mathcal{U}_{\overline{\mathcal{L}}}$. De même, si (L,R) est un état final de $\mathcal{U}_{\mathcal{L}}$, $(\overline{R},\overline{L})$ est un état initial de $\mathcal{U}_{\overline{\mathcal{L}}}$.
Si il existe $a \in A$ tel que $((L,R),a,(L',R'))$ est une transition de $\mathcal{U}_{\mathcal{L}}$, alors $L.a.R'$ est inclus dans $\mathcal{L}$, donc $\overline{R'}.a.\overline{L}$ est inclus dans $\overline{\mathcal{L}}$ et $((\overline{R'},\overline{L'}),a,(\overline{R},\overline{L}))$ est une transition de $\mathcal{U}_{\overline{\mathcal{L}}}$.
Donc l'automate transposé de l'automate universel $\mathcal{U}_{\mathcal{L}}$ est un sous-automate de $\mathcal{U}_{\overline{\mathcal{L}}}$. Réciproquement, par dualité, on obtient l'inclusion inverse, ces deux automates sont donc égaux. □

Pour n'importe quel automate $\mathcal{A}$ reconnaissant le langage $\mathcal{L}$, il existe un morphisme de $\mathcal{A}$ dans l'automate universel. En particulier, l'automate minimal d'un langage s'envoie dans l'automate universel. La propriété de minimalité signifie entre autre que tout morphisme non injectif donne un automate qui reconnaît un langage strictement plus grand. Comme l'automate universel ne reconnaît que $\mathcal{L}$, tout morphisme qui envoie l'automate minimal dans $\mathcal{U}_{\mathcal{L}}$ est injectif, il en est de même pour l'automate co-déterministe minimal. Ce fait peut être exprimé par la proposition suivante.

PROPOSITION 2.9 *L'automate minimal et l'automate co-déterministe minimal d'un langage rationnel $\mathcal{L}$ sont des sous-automates de l'automate universel $\mathcal{U}_{\mathcal{L}}$.*

— ∘ —

2 Calcul effectif de l'automate universel

Nous donnons ici une nouvelle méthode de calcul de l'automate universel. Comme l'a fait remarquer O. Carton ([12]), on peut effectuer le calcul de l'automate universel d'un langage $\mathcal{L}$ dans le monoïde syntaxique $M_{\mathcal{L}}$ de $\mathcal{L}$. On calcule alors la table de ce monoïde et on repère les couples maximaux (L,R) de sous-ensembles de $M_{\mathcal{L}}$ tels que $L.R$ est inclus dans P, image canonique de $\mathcal{L}$ dans $M_{\mathcal{L}}$.

Toutefois, si on considère $M_{\mathcal{L}}$ comme monoïde de transition de l'automate minimal de $\mathcal{L}$ – ce qui est une façon courante de le calculer –, on peut remarquer deux choses. D'une part, si un élément d'image α appartient à un facteur gauche L, tout élément d'image β qui agit de la même façon (à droite) sur l'état initial i de l'automate minimal appartient au même facteur gauche :

$$\forall\alpha\in L,\forall\beta\in M_{\mathcal{L}},\quad i\alpha=i\beta\Rightarrow\beta\in L.$$

D'autre part, si un élément d'image α appartient à un facteur droit R, tout élément d'image β qui agit de la même façon à gauche sur l'ensemble des état terminaux T de l'automate minimal appartient au même facteur droit :

$$\forall\alpha\in R,\forall\beta\in M_{\mathcal{L}},\quad T\alpha^{-1}=T\beta^{-1}\Rightarrow\beta\in R.$$

On peut donc fixer deux relations d'équivalences sur $M_{\mathcal{L}}$:

$$\alpha\sim_i\beta\Leftrightarrow i\alpha=i\beta,\text{ et}\qquad \alpha\sim_T\beta\Leftrightarrow T\alpha^{-1}=T\beta^{-1}.$$

Le produit de deux éléments α et β de $M_{\mathcal{L}}$ appartient à P si et seulement si l'intersection de $i\alpha$ et $T\beta^{-1}$ n'est pas vide. On peut donc calculer les factorisations dans $(M/\sim_i)\times(M/\sim_T)$. Remarquons au passage que $M/\sim_i$ est isomorphe à l'ensemble Q des états de l'automate minimal, puisque celui-ci est déterministe et que chaque état est accessible.

Comme, par maximalité, une factorisation est caractérisée par son facteur gauche, on peut décrire les factorisations par des ensembles d'états de l'automate minimal.

La nouvelle description que l'on obtient est donnée par la proposition suivante :

THÉORÈME 2.1 *Soit $\mathcal{A}_{\mathcal{L}}=\langle Q,A,E,\{i\},T\rangle$ l'automate minimal d'un langage rationnel $\mathcal{L}$. Soit $P=\{u\cdot T\mid u\in A^*\}$ l'ensemble des états du co-déterminisé de $\mathcal{A}_{\mathcal{L}}$.*

Soit $P_\cap$ la clôture de P par intersection, privée de l'ensemble vide. Alors, l'automate universel de $\mathcal{U}_\mathcal{L}$ est isomorphe à $\langle P_\cap,A,F,J,U \rangle$, avec :

$$\begin{aligned} J =& \{X \mid i \in X\} \\ U =& \{X \mid X \subseteq T\} \\ F =& \{(X,a,Y) \mid X \cdot a \subseteq Y \text{ et } \forall p \in X,\, p \cdot a \neq \emptyset\}. \end{aligned}$$

LEMME 2.10 *Soit $\mathcal{A} = \langle Q,A,E,i,T \rangle$ l'automate minimal d'un langage $\mathcal{L}$ et $P_\cap$ défini à partir de $\mathcal{A}$ comme dans le théorème ci-dessus. Il existe une bijection entre les éléments de $P_\cap$ et l'ensemble F des factorisations de $\mathcal{L}$:*

$$\begin{aligned} P_\cap &\longrightarrow F \\ X &\longmapsto \left(\bigcup_{p \in X} \mathsf{Past}_\mathcal{A}(p), \bigcap_{p \in X} \mathsf{Fut}_\mathcal{A}(p) \right) \end{aligned} \qquad \begin{aligned} F &\longrightarrow P_\cap \\ (L,R) &\longmapsto i \cdot L \end{aligned}$$

Démonstration. Examinons la première application. Soit X dans $P_\cap$ et son image (L,R). L'ensemble R n'est pas vide car X est un sous-ensemble d'un état du co-déterminisé de $\mathcal{A}$. L'ensemble L n'est pas vide car chaque élément de X est un état accessible de $\mathcal{A}$. Pour tout u dans L et tout v dans R, il existe un élément p de X tel que u est dans le passé de p et, par définition, v est dans le futur de p (comme de n'importe quel autre élément de X). Donc $u.v$ est un mot de $\mathcal{L}$ et (L,R) est une sous-factorisation.

Il faut maintenant montrer que (L,R) est une factorisation. On pose $L' = \{u \in A^* \mid u.R \subseteq \mathcal{L}\}$ et $R' = \{v \in A^* \mid L'.v \subseteq \mathcal{L}\}$; (L',R') est une factorisation qui domine (L,R). On pose $X' = i \cdot L'$. Si $X = X'$, alors $L = L'$ et

$$\begin{aligned} R' =& \{v \mid \forall u \in L, u.v \in \mathcal{L}\} \\ =& \{v \mid \forall u \in L, i \cdot (u.v) \in T\} \\ =& \{v \mid X \cdot v \in T\} \\ =& \bigcap_{p \in X} \mathsf{Fut}_\mathcal{A}(p) \\ =& R. \end{aligned}$$

Donc, si (L,R) n'est pas maximal, $X \subset X'$. Quel que soit v tel que $X \subseteq v \cdot T$, quel que soit u dans L', $u.v$ est un mot de $\mathcal{L}$, donc $X' \subseteq v \cdot T$. Donc X n'appartient pas à $P_\cap$, ce qui est contradictoire.

Cette application est injective. En effet, les passés des états de l'automate minimal sont disjoints, deux sous-ensembles de Q distincts donnent donc deux factorisations dont les facteurs gauches sont distincts.

Cette application est surjective. En effet, pour toute factorisation (L,R), on peut définir $X = i \cdot L$. Comme il s'agit d'une factorisation, $R = \{v \mid X \cdot R \subseteq T\}$; donc X apparaît aussi comme l'intersection des éléments de $\{Y \subseteq Q \mid \exists v \in R, Y = v \cdot T\}$; donc X appartient bien à $P_\cap$ et l'image de X est évidemment (L,R). □

Démonstration du théorème 2.1. Soit $\mathcal{A} = \langle Q,A,E,i,T \rangle$ l'automate minimal de $\mathcal{L}$. Posons $\mathcal{B} = \langle P_\cap,A,F,J,U \rangle$ l'automate défini dans le théorème et $\mathcal{U}_\mathcal{L} = \langle S,A,G,K,V \rangle$ l'automate universel de $\mathcal{L}$. Le lemme précédent établit qu'il existe une bijection entre $P_\cap$ et S. Il suffit maintenant de montrer qu'il y a correspondance entre les états initiaux, les états terminaux et entre les transitions. Soit φ la bijection de $P_\cap$ sur S définie dans le lemme.

Si $X \in P_\cap$ est initial, i appartient à X et le mot vide appartient au facteur gauche de $X\varphi$; $X\varphi$ est donc initial.
Si une factorisation $p = (L,R)$ est un état initial de $\mathcal{U}_\mathcal{L}$, le mot 1_{A^*} est dans L, donc i appartient à $p\varphi^{-1}$ qui est donc un état initial de $\mathcal{B}$.

Si $X \in P_\cap$ est final, X est inclus dans T, donc le mot vide appartient au facteur droit de $X\varphi$; $X\varphi$ est donc final.
Si une factorisation $p = (L,R)$ est un état final de $\mathcal{U}_\mathcal{L}$, L est inclus dans $\mathcal{L}$, donc $p\varphi^{-1}$ est inclus dans T et est donc un état final de $\mathcal{B}$.

Si (X,a,Y) est une transition de $\mathcal{B}$, alors $X \cdot a \subseteq Y$ et, pour tout r dans X, $r \cdot a$ n'est pas vide. Donc, si L et L' sont les facteurs gauches respectifs de $X\varphi$ et $Y\varphi$, $L.a \subseteq L'$. Donc $X\varphi,a,Y\varphi$ est une transition de $\mathcal{U}_\mathcal{L}$.
Si (p,a,q) est une transition de $\mathcal{U}_\mathcal{L}$, avec L et L' facteurs gauches de p et q respectivement, on a $L.a \subseteq L'$; donc $p\varphi^{-1} \cdot a \subseteq q\varphi^{-1}$, et comme $L.a$ est un sous-facteur gauche du langage, pour tout r dans $p\varphi^{-1}$, $r \cdot a$ n'est pas vide. Donc $(p\varphi^{-1},a,q\varphi^{-1})$ est une transition de $\mathcal{B}$. □

COROLLAIRE 2.11 *Soit $\mathcal{L}$ un langage rationnel. Soit $\mathcal{A}_\mathcal{L}$ son automate minimal et n le nombre d'états de $\mathcal{A}_\mathcal{L}$. Le nombre d'états de l'automate universel est au plus $2^n - 1$.*

Démonstration. Les états de l'automate universel sont, on l'a vu, indexés par des parties non vides d'états de $\mathcal{A}_\mathcal{L}$. Leur nombre est donc inférieur à $2^n - 1$. □

REMARQUE 2.3 Cette borne peut être effectivement atteinte pour tout n. Posons $\mathcal{A}_n = \langle \mathbb{Z}/n\mathbb{Z},\{a;b\},E,\{0\},[1;n-1] \rangle$, avec

$$E = \{(n,a,n+1) \mid n \in \mathbb{Z}/n\mathbb{Z}\} \cup \{(n,b,n+1) \mid n \in [1;n-1]\} \cup \{(0,b,0)\}$$

Cet automate est minimal. Les états de son co-déterminisé sont les sous-ensembles à $n-1$ éléments ainsi que $\mathbb{Z}/n\mathbb{Z}$ tout entier. A tout sous-ensemble non vide de $\mathbb{Z}/n\mathbb{Z}$ correspond une factorisation du langage.

REMARQUE 2.4 On peut effectuer la construction précédente sans calculer exactement l'ensemble des états de l'automate universel mais en considérant tous les sous-ensembles non vides de l'ensemble des états de l'automate minimal. On obtient alors un automate qui contient l'automate universel et qui compte exactement $2^n - 1$ états. Cet automate correspond à **l'automate des sous-ensembles d'ordre 0** présenté par R. Cohen et J. Brzozowski [18]. Nous allons par la suite généraliser cette construction pour obtenir, à partir de n'importe quel automate qui accepte le langage, un automate qui contient l'automate universel. Un autre automate, l'automate **fondamental**, présenté par O. Matz et A.Potthoff [45] et construit à partir de l'automate minimal, contient lui aussi l'automate universel.

Exemple 10.1 Soit $\mathcal{L}_4$ le langage dont l'automate minimal est présenté figure 11 a). Le monoïde de transition de l'automate minimal (engendré par les actions des lettres sur les états) compte vingt-quatre éléments. Il s'agit en effet de l'ensemble des applications d'un ensemble de trois éléments dans lui-même, hormi les trois transpositions. Utilisons le théorème 2.1 pour calculer l'automate universel. Les états du co-déterminisé de l'automate minimal sont :

$$\{p\},\{q\},\{r\},\{p,q\},\{p,r\},\{q,r\},\{p,q,r\}.$$

Cet ensemble est clos pour l'intersection (sauf le vide); ce sont donc les états de l'automate universel.

La table de transition de l'automate universel est la suivante :

	p	q	r	p,q	p,r	q,r	p,q,r
p	–	$a+c$	b	$a+c$	b	$a+b+c$	$a+b+c$
q	$b+c$	–	a	$b+c$	$a+b+c$	a	$a+b+c$
r	$a+c$	–	b	$a+c$	$a+b+c$	b	$a+b+c$
p,q	–	–	–	c	b	a	$a+b+c$
p,r	–	–	b	$a+c$	b	b	$a+b+c$
q,r	c	–	–	c	$a+b+c$	–	$a+b+c$
p,q,r	–	–	–	c	b	–	$a+b+c$

Les états initiaux sont ceux qui contiennent p et les états terminaux sont inclus dans $\{r\}$.

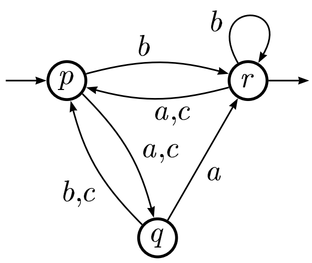
L'automate universel de ce langage compte sept états et est présenté figure 11 b). La flèche en pointillé arrivant sur l'état du bas indique qu'on peut arriver dans cet état en lisant n'importe quelle lettre à partir de n'importe quel état.

— ∘ —

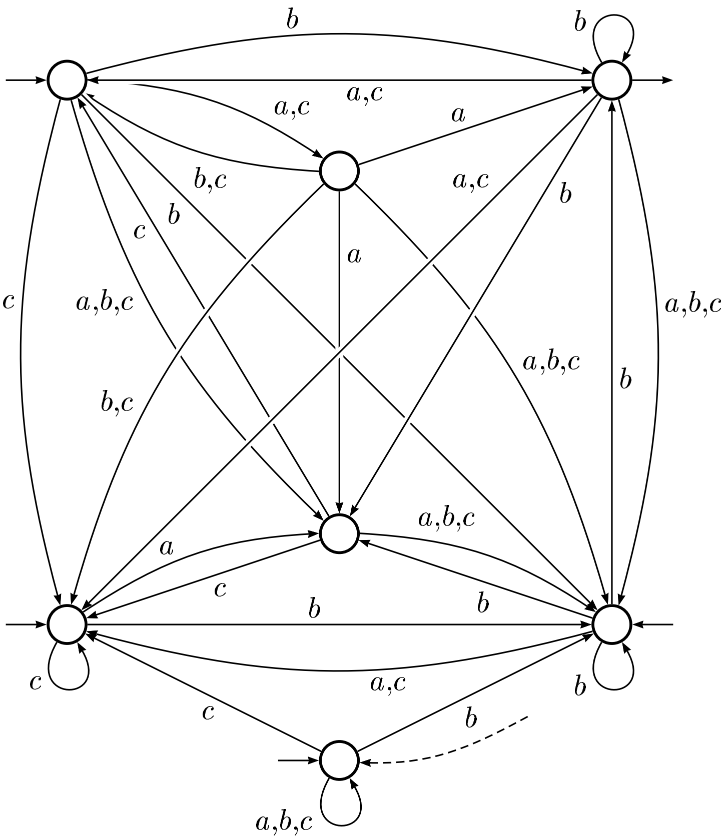
3 Écorché de l'automate universel

Nous introduisons ici une nouvelle présentation de l'automate universel qui permettra de mettre en relief certaines propriétés des automates universels que nous verrons par la suite, mais aussi, ce qui n'est pas négligeable, de pouvoir dessiner des automates universels sans être asphyxié par la profusion des transitions.

Pour cela, il peut être intéressant de distinguer les transitions « significatives » et celles qui découlent d'une « perte d'information ». Ainsi, si (L,R) et (L',R') sont deux factorisations telles que $L \subseteq L'$, pour toute transition partant de (L',R') étiquetée par une lettre a et allant dans un état q, il existe une transition partant de (L,R) étiquetée par a et allant dans le même état q.



(a) L'automate minimal de $\mathcal{L}_4$.



(b) L'automate universel de $\mathcal{L}_4$.

FIG. 11 – *Deux automates canoniques de $\mathcal{L}_4$.*

On peut donc supprimer ces dernières et considérer qu'il existe une transition spontanée de (L,R) à (L',R'). Dans le même esprit, on peut ne retenir que les transitions spontanées minimales, c'est-à-dire celles dont la clôture permet de retrouver les autres. C'est le sens de la définition suivante.

DÉFINITION 2.5 *Soit $\mathcal{L}$ un langage de A^* et Q l'ensemble de ses factorisations. L'**écorché de l'automate universel** $\check{\mathcal{U}}_{\mathcal{L}} = \langle\, Q,A,E,I,T \,\rangle$[3] est défini par :*

$$\begin{aligned} I =& \{p\}, \text{ avec } p \text{ factorisation initiale,} \\ T =& \{q\}, \text{ avec } q \text{ factorisation finale,} \\ E =& \{((L,R),a,(L',R')) \in Q \times A \times Q \mid (L,R') \in \mathsf{max}\{(X,Y) \mid X.a.Y \subseteq \mathcal{L}\}\} \\ & \cup \{((L,R),1_{A^*},(L',R')) \in Q \times \{1_{A^*}\} \times Q \mid L' \in \mathsf{min}\{X \mid L \subset X\}. \end{aligned} \tag{1}$$

3. Nous utilisons cette métaphore anatomique pour souligner le fait que cette nouvelle présentation permet d'aller au-delà des apparences (touffues) de l'automate universel pour comprendre comment il « fonctionne ».

REMARQUE 2.5 Les facteurs gauches appartenant à l'ensemble $\mathsf{min}\{X \mid L \subset X\}$ correspondent aux facteurs droits de l'ensemble $\mathsf{max}\{Y \mid Y \subset R\}$. On peut définir un ordre sur les états de l'automate universel : $(L,R) < (L',R') \Leftrightarrow L \subset L'$. Les transitions spontanées de l'écorché de l'automate universel forment le graphe orienté de cet ordre.

EXEMPLE 1.11 Les factorisations de $\mathcal{L}_1 = A^*abA^*$ sont :

$$p = (A^*,A^*abA^*), \qquad q = (A^*aA^*,A^*bA^*) \qquad \text{et} \qquad r = (A^*abA^*,A^*).$$

On voit que les facteurs gauches forment une chaîne décroissante; les transitions spontanées de l'écorché de l'automate universel sont donc $(q,1_{A^*},p)$ et $(r,1_{A^*},q)$. Les sous-factorisations maximales de $\mathcal{L}_1$ à trois facteurs avec a comme facteur central sont :

$$(A^*,a,A^*bA^*) \qquad \text{et} \qquad (A^*abA^*,a,A^*);$$

les transitions étiquetées par a sont donc (p,a,q) et (r,a,r). De même, les transitions étiquetées par b sont (p,b,p) et (q,b,r). L'écorché de l'automate universel de $\mathcal{L}_1$ est donc celui de la figure 12.

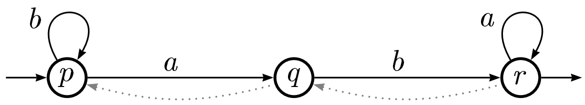


FIG. 12 – *Écorché de l'automate universel de* $\mathcal{L}_1$.

EXEMPLE 8.4 Les factorisations du langage $\mathcal{L}_2 = a^+$ étant (a^*,a^+) et (a^+,a^*), le calcul de l'écorché de l'automate universel se fait simplement. Le résultat est présenté figure 13.

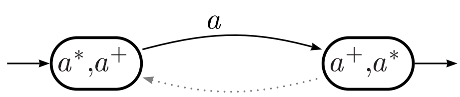
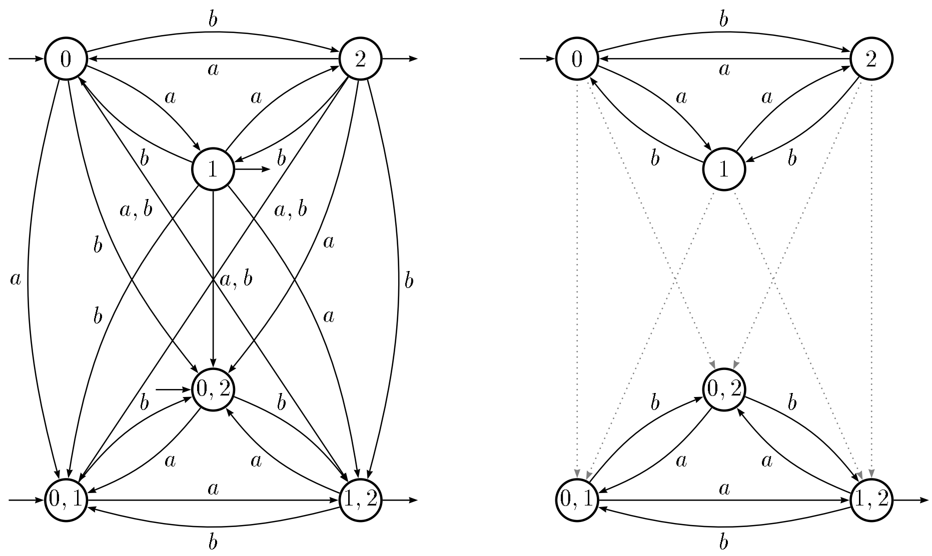


FIG. 13 – *Écorché de l'automate universel de* $\mathcal{L}_2$.

EXEMPLE 7.7 La figure 14 montre d'une part l'automate $(\{a\})$-universel du langage $\mathcal{L}'_3$ et d'autre part l'écorché de ce même automate.

EXEMPLE 10.2 Reprenons l'exemple du langage $\mathcal{L}_4$ présenté exemple 10.1 page 59. Les factorisations y étaient représentées par des sous-ensembles d'états de l'automate minimal. Les inclusions entre factorisations suivent les inclusions entre ensembles.

Si une lettre a apparaît dans deux cases distinctes de la table de transition de $\mathcal{U}_{\mathcal{L}_4}$ indexées par $(\mathcal{L}_{i_1},\mathcal{L}_{j_1})$ et $(\mathcal{L}_{i_2},\mathcal{L}_{j_2})$ et que $\mathcal{L}_{i_1} \subseteq \mathcal{L}_{i_2}$ (le passé de i_1 est inclus dans celui



(a) L'automate universel de $\mathcal{L}_3'$ ...   (b) ... et son écorché.

FIG. 14 – *Deux façons de représenter l'automate universel.*

de i_2) et $\mathcal{L}_{j_2} \subseteq \mathcal{L}_{j_1}$ (le futur de j_1 est inclus dans le futur de j_2), la lettre a ne figurera pas dans la case $(\mathcal{L}_{i_1},\mathcal{L}_{j_1})$ de la table de transition de $\check{\mathcal{U}}_{\mathcal{L}_4}$.

D'autre part, on indique une transition spontanée en $(\mathcal{L}_i,\mathcal{L}_j)$ si $\mathcal{L}_i \subset \mathcal{L}_j$ et qu'il n'existe pas k tel que $\mathcal{L}_i \subset \mathcal{L}_k \subset \mathcal{L}_j$.

La table de transition de $\check{\mathcal{U}}_{\mathcal{L}_4}$ est donc la suivante :

	p	q	r	p,q	p,r	q,r	p,q,r
p	–	$a+c$	–	1_{A^*}	1_{A^*}	–	–
q	b	–	a	1_{A^*}	–	1_{A^*}	–
r	a	–	–	–	1_{A^*}	1_{A^*}	–
p,q	–	–	–	–	–	a	1_{A^*}
p,r	–	–	b	a	–	–	1_{A^*}
q,r	c	–	–	–	a	–	1_{A^*}
p,q,r	–	–	–	c	b	–	a

Le seul état initial est celui qui correspond à p et le seul état final reste r. L'automate obtenu est représenté figure 15. Les transitions spontanées sont indiquées par des pointillés gris.

L'écorché de l'automate universel compte peu de transitions non spontanées. Plus précisément :

PROPOSITION 2.12 *Soit $\check{\mathcal{U}}_{\mathcal{L}}$ l'écorché de l'automate universel d'un langage rationnel de A^*. Alors, pour toute lettre a de A, pour tout état p de $\check{\mathcal{U}}_{\mathcal{L}}$, il y a au plus une transition étiquetée*

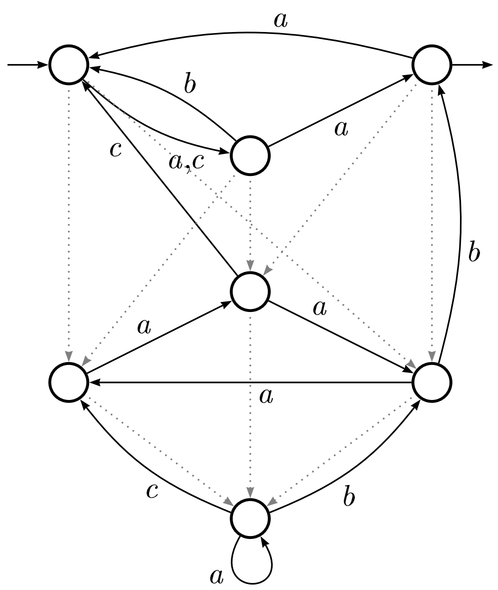


Fig. 15 – *Écorché de l'automate universel de $\mathcal{L}_4$.*

par a qui part de (resp. arrive dans) p.[4]

Démonstration. Soit $p = (L,R)$ un état de $\check{\mathcal{U}}_\mathcal{L}$ tel qu'il existe une transition sortant de p et étiquetée par a.

$$R' = \mathsf{max}\{Y \mid L.a.Y \subseteq \mathcal{L}\} \text{ et } L' = \mathsf{max}\{X \mid X.R' \subseteq \mathcal{L}\}.$$

Alors $q = (L',R')$ est bien une factorisation. Toute factorisation (L'',R'') telle que $L.a.R'' \subseteq \mathcal{L}$ vérifie $R'' \subseteq R'$, donc (p,a,q) est la seule transition étiquetée par a qui part de p. Donc, les transitions non spontanées de $\check{\mathcal{U}}_\mathcal{L}$ forment un automate déterministe, et, de même, co-déterministe. □

On va voir que retrouver l'automate universel d'un langage à partir de son écorché consiste à calculer la clôture des transitions par rapport aux transitions spontanées :

Proposition 2.13 *Soit $\mathcal{U}_\mathcal{L}$ l'automate universel d'un langage rationnel $\mathcal{L}$ et $\check{\mathcal{U}}_\mathcal{L}$ son écorché. Pour toute transition (p,a,q) de $\mathcal{U}_\mathcal{L}$, il existe deux états p' et q' et une transition (p',a,q') dans $\check{\mathcal{U}}_\mathcal{L}$ ainsi que des chaînes de transitions spontanées de p à p' et de q' à q. De même, pour tout état initial p de $\mathcal{U}_\mathcal{L}$ (resp. tout état final q de $\mathcal{U}_\mathcal{L}$), il existe une suite de transitions spontanées de p_i à p (resp de q à p_t) dans $\check{\mathcal{U}}_\mathcal{L}$, où p_i est la factorisation initiale et p_t la factorisation finale.*

Démonstration. Soit $p = (L,R)$ et $q = (L',R')$. Par définition de l'automate $\mathcal{U}_\mathcal{L}$, on a $L.a.R' \subseteq \mathcal{L}$. Soit (L_1,R_2) un couple maximal tel que $L \subseteq L_1$, $R' \subseteq L_2$ et $L_1.a.R_2$. Il existe une transition dans $\check{\mathcal{U}}_\mathcal{L}$ entre (L_1,R_1) et (L_2,R_2). D'autre part, comme $L \subseteq L_1$, il

4. S'il ne comporte pas de transition spontanée, un tel automate est dit réversible; nous reviendrons sur cette famille d'automate au chapitre suivant.

existe une chaîne de transitions spontanées entre p et (L_1,R_1) et, de même, une chaîne entre (L_2,R_2) et q.

Si $p = (L,R)$ est un état initial de $\mathcal{U}_{\mathcal{L}}$, et $p_i = (L_i,\mathcal{L})$ la factorisation initiale, L contient L_i, donc il existe une chaîne de transitions spontanées entre p_i et p. De même pour les états terminaux. □

— ◦ —

4 Développement d'un automate

4.1 Motivations et définitions

On peut généraliser la construction correspondant au théorème 2.1 pour n'importe quel automate qui accepte le langage :

Proposition 2.14 *Soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate qui accepte un langage rationnel $\mathcal{L}$. Soit $Q_d = \{Q \cdot u \mid u \in A^*\}$ les états du déterminisés de $\mathcal{A}$ et $Q_c = \{u \cdot Q \mid u \in A^*\}$ les états de son co-déterminisé.*
Soit S l'ensemble des couples maximaux (L,R) composés d'un sous-ensemble de Q_d et d'un sous-ensemble de Q_c tels que l'intersection de chaque élément de L avec chaque élément de R est non vide. Alors, l'automate universel de $\mathcal{U}_{\mathcal{L}}$ est isomorphe à $\langle\, S,A,F,J,U \,\rangle$, avec :

$$\begin{aligned} J =& \{(L,R) \mid I \in L\} \\ U =& \{(L,R) \mid \forall X \in L, X \cap T \neq \emptyset\} \\ F =& \{((L,R),a,(L',R')) \mid L \cdot a \subseteq L' \text{ et } \forall X \in L, X \cdot a \neq \emptyset\}. \end{aligned}$$

Démonstration. Soit $M_{\mathcal{A}}$ le monoïde de transition de $\mathcal{A}$ et φ le morphisme canonique de A^* dans $M_{\mathcal{A}}$. Les factorisations de $\mathcal{L}$ sont reconnues par $M_{\mathcal{A}}$, et, comme on l'a déjà dit, deux éléments α et β de $M_{\mathcal{A}}$ tels que $I\alpha = I\beta$ sont dans les images des mêmes facteurs gauches de $\mathcal{L}$. Ces images caractérisent donc les facteurs gauches; il s'agit des états du déterminisé de $\mathcal{A}$. De même, deux éléments α et β de $M_{\mathcal{A}}$ tels que $T\alpha^{-1} = T\beta^{-1}$ sont dans les images des mêmes facteurs droits. Les facteurs droits sont donc caractérisés par les états du co-déterminisé de $\mathcal{A}$. Pour qu'un couple de mots (u,v) forme une sous-factorisation du langage reconnu par $\mathcal{A}$, il faut que $I \cdot (u.v) \cap T \neq \emptyset$, c'est-à-dire que $I \cdot u \cap v \cdot T \neq \emptyset$. Il existe donc une bijection entre les factorisations du langage et l'ensemble S :

$$(L,R) \longmapsto (I \cdot L, R \cdot T)$$

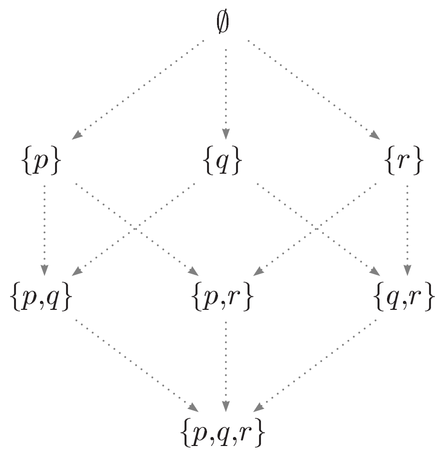
Une factorisation représente un état initial de l'automate universel si et seulement si son facteur gauche contient le mot vide, c'est-à-dire si et seulement si l'élément de S correspondant contient l'élément I. De même, elle représente un état terminal si et seulement si son facteur gauche est inclus dans le langage, c'est-à-dire si et seulement si tout élément de l'élément de S correspondant a une intersection non vide avec T.

Il existe une transition étiqueté par a entre deux états de l'automate universel si et seulement si les facteurs gauches L et L' des factorisations correspondant à ces états respectent $L.a \subseteq L'$, c'est à dire si et seulement si, pour tout X dans $I \cdot L$, il existe Y dans $I \cdot L'$ tel que $X \cdot a = Y$. □

On peut remarquer que, pour toute factorisation (L,R) du langage, pour tout mot u de L et tout mot v de A^*, si $I \cdot u \subseteq I \cdot v$, alors v appartient aussi à L. On peut donc représenter cette factorisation par l'ensemble $\mathsf{min}\{I \cdot u \mid u \in L\}$.

Chaque factorisation est ainsi caractérisée par une anti-chaîne de $\mathcal{P}(Q) \setminus \{\emptyset\}$.

Exemple 11 Supposons que l'ensemble des états d'un automate est $Q = \{p,q,r\}$. L'ensemble $\mathcal{P}(Q)$ est ordonné selon le schéma suivant :



Les anti-chaînes de $\mathcal{P}(Q) \setminus \{\emptyset\}$ sont donc :

$$\begin{array}{ccc}
\{\{p\}\}, & \{\{q\}\}, & \{\{r\}\}, \\
\{\{p,q\}\}, & \{\{p,r\}\}, & \{\{q,r\}\}, \\
 & \{\{p,q,r\}\}, & \\
\{\{p\},\{q\}\}, & \{\{p\},\{r\}\}, & \{\{q\},\{r\}\}, \\
\{\{p\},\{q,r\}\}, & \{\{q\},\{p,r\}\}, & \{\{r\},\{p,q\}\}, \\
\{\{p,q\},\{p,r\}\}, & \{\{p,q\},\{q,r\}\}, & \{\{p,r\},\{q,r\}\}, \\
 & \{\{p\},\{q\},\{r\}\}, & \\
 & \{\{p,q\},\{p,r\},\{q,r\}\}, &
\end{array}$$

Évidemment, chaque anti-chaîne de $\mathcal{P}(Q)$ ne correspond pas nécessairement à un facteur gauche. Toutefois, on peut construire un automate dont l'ensemble des états est indexé par les anti-chaînes de $\mathcal{P}(Q)$. Celui-ci sera plus gros que l'automate universel (on verra qu'il le contient) et reconnaîtra le même langage que $\mathcal{A}$. Sa définition nous permettra de montrer qu'il hérite de certaines propriétés de $\mathcal{A}$.

— ◦ —

Construction du développé d'un automate

Définition 2.6 *Soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate reconnaissant le langage rationnel $\mathcal{L}$. Soit S l'ensemble des anti-chaînes de $\{I \cdot u \mid u \in A^*\}$. Le* ***développé de l'automate*** *$\mathcal{A}$, noté $\mathcal{V}_\mathcal{A} = \langle\, S,A,F,J,U \,\rangle$, est défini par :*

$$\begin{aligned} J =& \{P \in S \mid \exists Y \in P,\ Y \subseteq I\}, \\ U =& \{P \in S \mid \forall X \in P,\ X \cap T \neq \emptyset\}, \\ F =& \{(P,a,P') \in S \times A \times S \mid \forall X \in P,\ \exists Y \in P',\ Y \subseteq X \cdot a\}. \end{aligned}$$

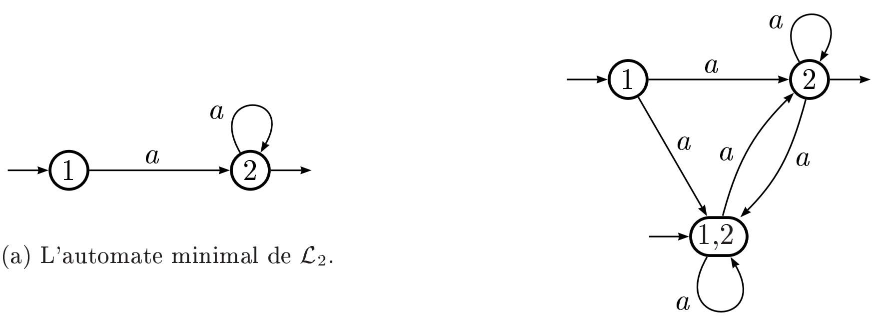
En fait, chaque anti-chaîne P représente l'ensemble des éléments minimaux d'un sous-ensemble $\overline{P}$ de $\mathcal{P}(Q)$ fermé supérieurement ($X \in \overline{P}$ et $X \subseteq Y \Rightarrow Y \in \overline{P}$). On peut alors réexprimer la définition du développé :

$$\begin{aligned} J =& \{P \in S \mid \exists I \in \overline{P}\}, \\ U =& \{P \in S \mid P \subseteq \overline{\{\{p\} \mid p \in T\}}\}, \\ F =& \{(P,a,P') \in S \times A \times S \mid P \cdot a \subseteq \overline{P'}\}. \end{aligned}$$

Remarque 2.6 On peut remarquer dès à présent que, d'une part, l'automate $\mathcal{V}_\mathcal{A}$ dépend fortement de l'automate $\mathcal{A}$ de départ et n'est donc en aucun cas un automate canonique du langage et que, d'autre part, la construction de l'automate $\mathcal{V}_\mathcal{A}$ est fortement orientée. On pourrait, en effet, construire un automate dual dont les états seraient les anti-chaînes des sous-ensembles obtenus à partir de T en faisant agir les lettres à gauche.

D'autre part, la structure de l'automate $\mathcal{V}_\mathcal{A}$ dépend en partie de la structure de $\mathcal{A}$. On peut par exemple montrer que $\mathcal{V}_\mathcal{A}$ contient le déterminisé de $\mathcal{A}$. (Il suffit de considérer le sous-automate de $\mathcal{V}_\mathcal{A}$ dont les états sont de la forme $\{I \cdot u\}$, pour u dans A^*.)

Exemple 8.5 La figure 16 présente l'automate minimal du langage $\mathcal{L}_2$ et son développé.



(a) L'automate minimal de $\mathcal{L}_2$.

(b) Le développé de l'automate minimal de $\mathcal{L}_2$.

Fig. 16 – *Deux automates reconnaissant $\mathcal{L}_2$.*

Exemple 1.12 Afin d'alléger les notations, on notera un sous-ensemble par un mot dont les lettres représentent les éléments du sous-ensemble. Par exemple, le mot pq représente l'ensemble $\{p,q\}$. L'automate $\mathcal{B}_1$ (figure 17) reconnaît le langage $\mathcal{L}_1 = A^*abA^*$. Calculons

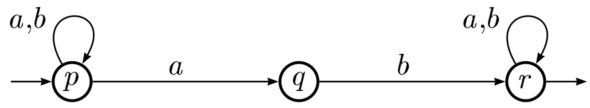


Fig. 17 – *L'automate $\mathcal{B}_1$ reconnaît $\mathcal{L}_1$.*

le développé de cet automate. L'ensemble $\{I \cdot u \mid u \in A^*\}$, qui est l'ensemble des états du déterminisé est

$R = \{p, pq, pr, pqr\}$. La table de transition du déterminisé est la suivante :

$$\begin{array}{c|c|c|c|c|} & p & pq & pr & pqr \\ \hline a & pq & pq & pqr & pqr \\ b & p & pr & pr & pr \end{array} \qquad (2)$$

Les anti-chaînes de R sont $S = \{\{p\},\{pq\},\{pr\},\{pqr\},\{pq,pr\}\}$. La définition du développé et la table de transition du déterminisé permettent de dire, par exemple : « Il y a une transition étiquetée par b de $\{pq\}$ vers n'importe quel état qui contient un ensemble plus petit que pr ». On a la matrice de transition suivante :

	p	pq	pr	pqr	pq,pr
p	a,b	a			a
pq	a,b	a	b		a,b
pr	a,b	a	a,b	a	a,b
pqr	a,b	a	a,b	a	a,b
pq,pr	a,b	a	b		a,b

De plus, l'état initial est $\{p\}$; les états finals sont $\{pqr\}$ et $\{pr\}$. (Par exemple, $\{pq,pr\}$ n'est pas final car l'intersection de pq avec T est vide.)

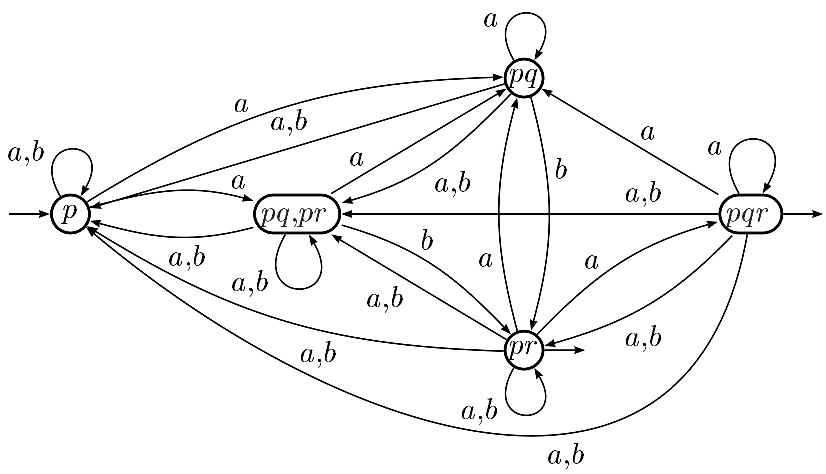


Fig. 18 – *Le développé de l'automate $\mathcal{B}_1$.*

On peut étendre la condition d'existence d'une transition aux chemins :

LEMME 2.15 *Soit P et P' deux états de $\mathcal{V}_\mathcal{A}$ et u un mot de A^*. Il existe un chemin étiqueté par u entre P et P' si et seulement si, pour tout X de P, il existe Y dans P' tel que $Y \subseteq X \cdot u$.*

Démonstration. La preuve est par récurrence sur la longueur de u.
Pour $|u| = 0$, la proposition est vraie (pour tout état p, $p \cdot 1_{A^*} = p$ est l'identité).
S'il existe un chemin entre P et P' étiqueté par $u = a.u'$, il existe un état P_1 tel que (P,a,P_1) est une transition et il existe un chemin étiqueté par u' entre P_1 et P'. Donc, pour tout X dans P, il existe Z dans P_1 et Y dans P' tels que $Y \subseteq Z \cdot u' \subseteq (X \cdot a) \cdot u' = X \cdot u$.

Réciproquement, si, pour tout X dans P, il existe Y dans P' tel que $Y \subseteq X \cdot u$, avec $u = a.u'$, on pose P_1 ensemble des états minimaux de $\{X \cdot a \mid X \in P\}$, P_1 est un état de $\mathcal{V}_\mathcal{A}$, (P,a,P_1) est une transition de $\mathcal{V}_\mathcal{A}$ et, quel que soit Z dans P_1, il existe X dans P et Y dans P' tel que $Z = X \cdot a$ et $Y \subseteq X \cdot (a.u') \subseteq Z \cdot u'$. Donc il existe un chemin entre P_1 et P' étiqueté par u'. □

— ∘ —

4.2 Propriétés de l'automate développé

Le lemme 2.15 nous permet d'exprimer le passé et le futur d'un état de $\mathcal{V}_\mathcal{A}$ de façon simple.

PROPOSITION 2.16 *Soit P un état de $\mathcal{V}_\mathcal{A}$ et u dans A^*.
u appartient à $\mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P)$ si et seulement si il existe Y dans P inclus dans $I \cdot u$.
u appartient à $\mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P)$ si et seulement si, pour tout X dans P, $(X \cdot u) \cap T$ est non vide.
Ce qui peut s'écrire:*

$$\mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P) = \bigcup_{Y \in P} \bigcap_{p \in Y} \mathsf{Past}_\mathcal{A}(p), \qquad \mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P) = \bigcap_{X \in P} \bigcup_{p \in X} \mathsf{Fut}_\mathcal{A}(p).$$

Démonstration. Si u appartient à $\mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P)$, il existe un chemin étiqueté par u entre un état initial P_0 et P; il existe un élément X de P_0 inclus dans I et d'après le lemme 2.15, il existe Y dans P tel que $Y \subseteq X \cdot u \subseteq I \cdot u$.

Réciproquement, si u est tel qu'il existe Y dans P tel que $Y \subseteq I \cdot u$, il existe un chemin entre $\{I\}$, qui est initial, et P, donc u appartient à $\mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P)$.

Si u appartient à $\mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P)$, il existe un chemin étiqueté par u entre P et un état final P_1. Donc, pour tout X dans P, il existe Y dans P_1 tel que $Y \subseteq X \cdot u$; comme $Y \cap T \neq 0_\mathbb{B}$, on obtinet $(X \cdot u) \cap T \neq 0_\mathbb{B}$,

Réciproquement, si u est tel que pour tout X dans P, $(X \cdot u) \cap T \neq 0_\mathbb{B}$, il existe clairement un chemin entre P et l'état correspondant aux éléments minimaux de $\{X \cdot u \mid X \in P\}$, qui est terminal; donc u appartient à $\mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P)$. □

Les propriétés suivantes de $\mathcal{V}_\mathcal{A}$ découlent directement de ces deux formules.

PROPOSITION 2.17 *Soit $\mathcal{A}$ un automate. L'automate $\mathcal{V}_\mathcal{A}$ reconnaît le même langage que l'automate $\mathcal{A}$.*

Démonstration. Considérons l'état $\{I\}$ qui est initial. D'après la proposition 2.16,

$$\mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(\{I\}) = \bigcup_{p\in I} \mathsf{Fut}_\mathcal{A}(p) = \mathcal{L}(\mathcal{A}).$$

Donc l'automate $\mathcal{V}_\mathcal{A}$ accepte tous les mots du langage $\mathcal{L}(\mathcal{A})$.
Réciproquement, quel que soit l'état terminal P de $\mathcal{V}_\mathcal{A}$,

$$\begin{aligned}\mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P) &= \bigcup_{X\in P}\bigcap_{p\in X} \mathsf{Past}_\mathcal{A}(p)\\ &\subseteq \bigcup_{X\in P}\bigcup_{p\in X\cap T} \mathsf{Past}_\mathcal{A}(p)\\ &\subseteq \bigcup_{p\in T} \mathsf{Past}_\mathcal{A}(p) = \mathcal{L}(\mathcal{A})\end{aligned}$$

Donc tous les mots acceptés par $\mathcal{V}_\mathcal{A}$ appartiennent à $\mathcal{L}(\mathcal{A})$. $\square$

PROPOSITION 2.18 *Soit $\mathcal{A}$ un automate. Il existe un morphisme injectif de l'automate universel $\mathcal{U}_\mathcal{L}$ du langage $\mathcal{L}(\mathcal{A})$ dans l'automate $\mathcal{V}_\mathcal{A}$, défini par :*

$$\begin{aligned}\mathcal{U}_\mathcal{L} &\longrightarrow \mathcal{V}_\mathcal{A}\\ (L,R) &\longmapsto P_L = \mathsf{min}\{I\cdot u \mid u \in L\}.\end{aligned}$$

L'image inverse de ce morphisme est donné par les égalitées :

$$L = \mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P_L)\ ,\quad R = \mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P_L).$$

Démonstration. Soit (L,R) une factorisation de $\mathcal{L}$ et P_L l'ensemble des éléments minimaux de $\{I\cdot u \mid u \in L\}$.
D'après la proposition 2.16,

$$\begin{aligned}\mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P_L) &= \bigcup_{X\in P_L}\bigcap_{p\in X} \mathsf{Past}_\mathcal{A}(p)\\ &= \bigcup_{u\in L}\bigcap_{p\in I\cdot u} \mathsf{Past}_\mathcal{A}(p)\\ &\supseteq \bigcup_{u\in L} u = L\end{aligned}\quad,\quad \begin{aligned}\mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P_L) &= \bigcap_{X\in P_L}\bigcup_{p\in X} \mathsf{Fut}_\mathcal{A}(p)\\ &= \bigcap_{u\in L}\bigcup_{p\in I\cdot u} \mathsf{Fut}_\mathcal{A}(p)\\ &\supseteq \bigcap_{u\in L}\{v \mid u.v \in \mathcal{L}\} = R\end{aligned}$$

Comme (L,R) est une factorisation (donc maximale) de $\mathcal{L}$, on obtient $L = \mathsf{Past}_{\mathcal{V}_\mathcal{A}}(P_L)$ et $R = \mathsf{Fut}_{\mathcal{V}_\mathcal{A}}(P_L)$.
Il existe donc une application injective des états de l'automate universel dans ceux de $\mathcal{V}_\mathcal{A}$.
Il reste à montrer que cette application est un morphisme.

Si (L,R) est initial ou final respectivement, l'état P_L l'est aussi.

Soit (L,R) et (L',R') deux factorisations telles que $L.a \subseteq L'$. Donc quel que soit u dans L, $u.a$ appartient à L'; donc quel que soit X dans P_L, il existe Y dans $P_{L'}$ tel que Y est inclus dans $X \cdot a$. Il y a donc une transition entre P_L et $P_{L'}$ étiquetée par a. □

Cette proposition peut se réexprimer de la façon suivante:

COROLLAIRE 2.19 *Soit $\mathcal{A}$ un automate. L'automate $\mathcal{V}_{\mathcal{A}}$ contient l'automate universel du langage reconnu par $\mathcal{A}$.*

Non seulement ce résultat justifie la définition de $\mathcal{V}_{\mathcal{A}}$ comme « approximation » de l'automate universel, mais elle permet de voir les états de l'automate universel comme des parties de $\{I \cdot u \mid u \in \mathcal{L}\}$, ce qui donne un point d'ancrage supplémentaire pour travailler avec l'automate universel.

REMARQUE 2.7 La taille du développé d'un automate est difficile à évaluer. Le nombre de ses états est le nombre d'anti-chaînes de $\mathcal{P}(Q)$, où Q est un ensemble de n états. Avoir une expression close pour exprimer cette quantité en fonction de n est un problème ouvert connu sous le nom de problème de Dedekind. On trouvera en annexe un bref rappel de ce qui est connu sur ces nombres.

— ◦ —

4.3 Écorché du développé

De même qu'on a défini l'écorché de l'automate universel, on peut définir l'écorché du développé d'un automate. Comme la définition du développé est orientée, la définition de son écorché le sera aussi. Ainsi, les transitions non spontanées de l'écorché de l'automate universel formaient un sous-automate réversible (au plus une transition entrante et une transition sortante pour chaque lettre), alors que ces transitions formeront un sous-automate déterministe de l'écorché du développé.

DÉFINITION 2.7 *Soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate reconnaissant le langage rationnel $\mathcal{L}$. Soit S l'ensemble des anti-chaînes de $\{I \cdot u \mid u \in A^*\}$. On ordonne partiellement les élément de S :*

$$P,P' \in S,\ P \preccurlyeq P' \Leftrightarrow \overline{P} \subseteq \overline{P'}.$$

*L'**écorché du développé** de l'automate $\mathcal{A}$, noté $\check{\mathcal{V}}_{\mathcal{A}} = \langle\, S,A,F,J,U \,\rangle$, est défini par :*

$$\begin{aligned} J =& \{I\},\\ U =& \min_{\preccurlyeq}\{P \in S \mid \forall x \in P, x \cap T \neq \emptyset\},\\ F =& \{(P,a,P') \in S \times A \times S \mid P' = \min_{\subseteq}\{X \cdot a \mid X \in P\}\}\\ &\cup\{(P,1_{A^*},P') \in S \times A \times S \mid P' \in \min_{\preccurlyeq}\{R \mid P \prec R\}\}. \end{aligned}$$

Remarque 2.8 Deux remarques sur cette définition :

i) Noter que selon l'ordre défini sur S, les singletons sont des éléments maximaux.

ii) T' n'appartient pas nécessairement à S; si c'est le cas, $U = \{T'\}$.

Exemple 1.13 En appliquant la définition et en se basant sur la table 2, on calcule l'écorché du développé de l'automate $\mathcal{B}_1$:

	p	pq	pr	pqr	pq,pr
p	b	a			
pq		a	b		1_{A^*}
pr			b	a	1_{A^*}
pqr		1_{A^*}	$b,1_{A^*}$	a	
pq,pr	1_{A^*}	a	b		

L'état initial est $\{p\}$; l'état final est $\{pr\}$.

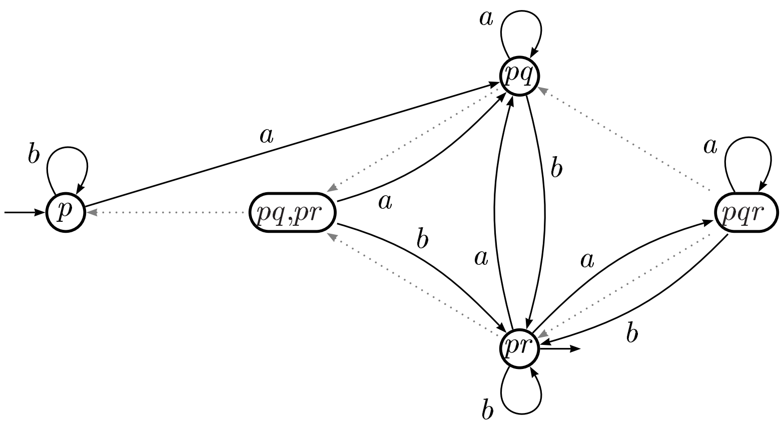


Fig. 19 – *L'écorché du développé de $\mathcal{B}_1$.*

Exemple 12.1 On considère l'automate $\mathcal{A}_{r_1}$ présenté figure 20.

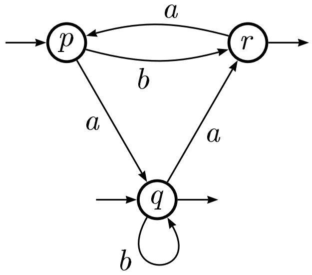


Fig. 20 – *L'automate $\mathcal{A}_{r_1}$*

La table de transition de l'écorché du développé de $\mathcal{A}_{r_1}$ (sans les transitions spontanées) est donnée par le tableau 3.

États	a	b	I	T
p	q	r	0	0
q	r	q	0	0
r	p	$-$	0	0
$\{p,q\}$	$\{q,r\}$	$\{q,r\}$	0	0
$\{q,r\}$	$\{p,r\}$	$-$	0	1
$\{p,r\}$	$\{p,q\}$	$-$	0	0
$\{p,q,r\}$	$\{p,q,r\}$	$-$	0	0
pq	qr	qr	1	0
qr	pr	q	0	0
pr	pq	r	0	0
$\{p,qr\}$	$\{q,pr\}$	$\{q,r\}$	0	0
$\{q,pr\}$	$\{r,pq\}$	$\{q,r\}$	0	0
$\{r,pq\}$	$\{p,qr\}$	$-$	0	0
$\{pq,pr\}$	$\{pq,qr\}$	r	0	0
$\{pq,qr\}$	$\{pr,qr\}$	q	0	0
$\{pr,qr\}$	$\{pq,pr\}$	$\{q,r\}$	0	0
$\{pq,pr,qr\}$	$\{pq,pr,qr\}$	$\{q,r\}$	0	0

(3)

L'automate obtenu est présenté figure 21. On peut remarquer que l'état $\{p,q,r\}$ n'est pas co-accessible. Il ne l'est donc pas non plus dans le développé. Cet état n'appartiendra donc pas à l'automate universel.

De même que pour l'automate universel, on peut retrouver le développé d'un automate à partir de son écorché en calculant la clôture des transitions par rapport aux transitions spontanées.

PROPOSITION 2.20 *Soit $\mathcal{A}$ un automate, $\mathcal{V}_{\mathcal{A}}$ son développé et $\check{\mathcal{V}}_{\mathcal{A}}$ son écorché. Pour toute transition (p,a,q) de $\mathcal{V}_{\mathcal{A}}$, il existe un état q' et une transition (p,a,q') dans $\check{\mathcal{V}}_{\mathcal{A}}$ ainsi qu'une chaîne de transitions spontanées de q' à q.*
De même, pour tout état initial p de $\mathcal{V}_{\mathcal{A}}$ (resp. tout état final q de $\mathcal{V}_{\mathcal{A}}$), il existe une suite de transitions spontanées de l'état initial à p (resp. de p à un état final) dans $\check{\mathcal{V}}_{\mathcal{A}}$.

Démonstration. Soit (p,a,q) une transition de $\mathcal{V}_{\mathcal{A}}$. Par définition, pour tout X dans p, il existe Y dans q tel que $Y \subseteq X \cdot a$. Soit $q' = \mathsf{min}\{X \cdot a \mid X \in p\}$. Alors, (p,a,q') est une transition de $\check{\mathcal{V}}_{\mathcal{A}}$ et $q \preccurlyeq q'$, donc il existe une chaîne de transitions spontanées de q' à q.
Si p est un état initial de $\mathcal{V}_{\mathcal{A}}$, alors $I \in p$, donc $p \preccurlyeq \{I\}$ et il existe une chaîne de transitions spontanées de $\{I\}$ à p.
Si p est un état final de $\mathcal{V}_{\mathcal{A}}$, alors il existe un état final p' de $\check{\mathcal{V}}_{\mathcal{A}}$ tel que $p' \preccurlyeq p$ et il existe une chaîne de transitions spontanées de p à p'. □

— ∘ —

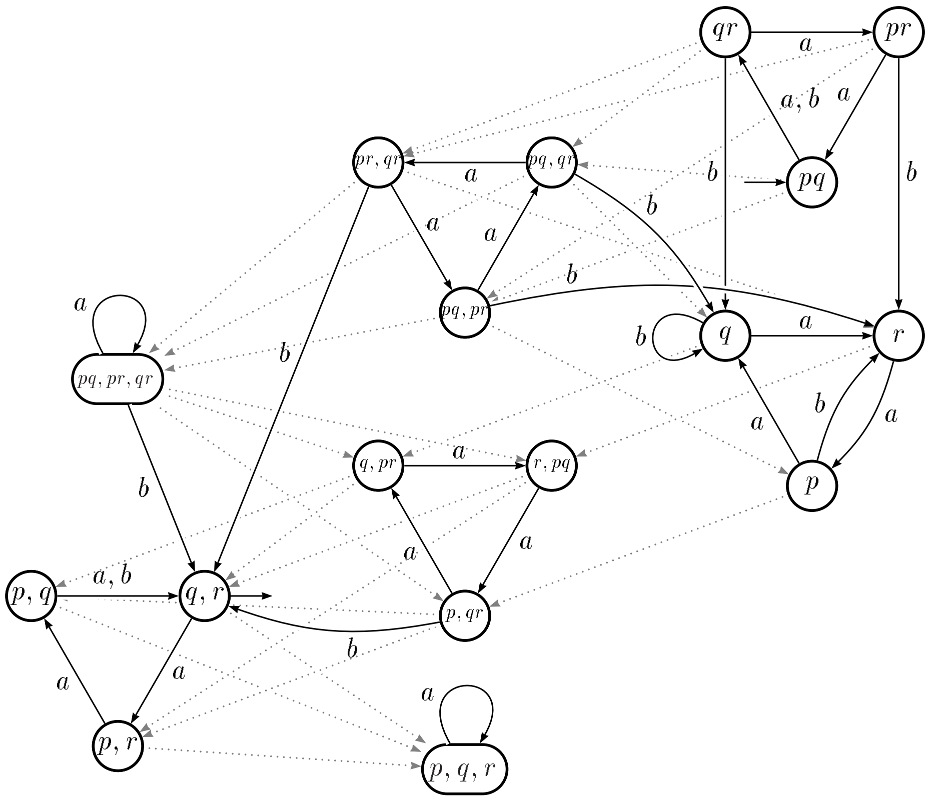


FIG. 21 – *Écorché du développé de* $\mathcal{A}_{r_1}$.

Chapitre 3

Automates universels et langages réversibles

Les langages réversibles forment une famille de langages rationnels qui se trouvent à la croisée de différents domaines.

Une restriction de cette famille est celle des langages réalisés par des automates réversibles déterministes. Les premières extensions des résultats sur la hauteur d'étoile aux langages réversibles se restreignent à cette famille [17, 31]. Nous aurons l'occasion d'y revenir. Dans un autre domaine, D. Angluin [2] classe ces langages selon le nombre d'états terminaux de l'automate minimal qui les reconnaît et montre qu'un langage d'une classe donnée peut être inféré par un panel de mots du langage. Les automates *inverses* sont eux aussi des automates réversibles « déterministes » sur un alphabet $A \cup \overline{A}$, où, à chaque transition (p,a,q) correspond une transition $(q,\overline{a},p)$. Ces automates se révèlent particulièrement utiles pour l'étude des semi-groupes inversifs [62, 57]. T.E. Hall [29] a établi que les langages clos par multiplication et reconnus par des automates réversibles déterministes sont des étoiles de *codes bipréfixes*. C'est encore à des automates réversibles déterministes que les automates quantiques sur un alphabet unaire ont été comparés [1]. Il a été montré que si l'automate quantique doit donner la bonne réponse avec une probabilité forte, il peut être simulé par un automate réversible.

La famille des langages reconnus par des automates réversibles déterministes n'est cependant pas stable par union. Les langages réversibles eux, forment une *variété positive* de langages. Une caractérisation algébrique en a été donnée par J.-E. Pin [51], ce qui permet de décider de la réversibilité d'un langage réversible. P.-C. Héam [36] a montré que ces langages jouent un rôle central dans l'étude topologique des langages rationnels.

Toutefois, ces résultats ne donnent pas d'algorithme raisonnable pour construire un automate réversible qui accepte un langage réversible donné. C'est ce que nous nous proposons de faire, à travers le résultat suivant :

Théorème *L'automate universel d'un langage réversible contient un automate quasi-réversible équivalent.*

Pour ce faire, il nous faut évidemment introduire les automates *quasi-réversibles* dont on montre qu'ils sont équivalents aux automates réversibles, dans le sens où, non seulement ils reconnaissent les mêmes langages, mais aussi parce qu'on peut facilement les transformer en automates réversibles.

Avant de se lancer dans l'étude des langages réversibles, nous étudierons d'abord une sous-classe de langages, les langages à groupe. On montre que les automates universels des langages à groupe se décomposent en « étages » et que leurs composantes fortement connexes sont elles-mêmes des automates à groupe. Cette première étude nous permet d'introduire des notions qui apparaîtront plus complexes lorsqu'elles seront appliquées aux langages réversibles auxquels est consacrée la dernière partie de ce chapitre.

Nous y montrons, dans un premier temps, que les pelotes de l'automate universel d'un tel langage sont réversibles, puis que l'automate universel d'un langage réversible contient un sous-automate quasi-réversible qui accepte le langage. Nous donnons un algorithme qui permet d'obtenir un tel automate.

1 Langages réversibles

DÉFINITION 3.1 *Un automate $\mathcal{A}$ sur un alphabet A est un **automate réversible** si pour tout état q de $\mathcal{A}$, pour toute lettre a de A, il existe au plus une transition étiquetée par a qui part de q et au plus une qui y arrive. Un langage $\mathcal{L}$ est un **langage réversible** s'il existe un automate réversible qui reconnaît $\mathcal{L}$.*

Malheureusement, si un automate réversible n'est pas déterministe, c'est-à-dire s'il a plusieurs états initiaux, il se peut que l'automate minimal du langage réalisé ne soit pas réversible. Nous aurons l'occasion de le vérifier sur des exemples. P.-C. Héam [36] a d'ailleurs montré qu'il existe des langages réversibles qui ne peuvent pas être reconnus par un automate réversible déterministe.

La première question, naturelle, qui se pose donc est de savoir si, étant donné un langage rationnel, il est réversible ou non. J.-E. Pin [51] a apporté une réponse à ce problème. La caractérisation d'un langage réversible ne peut pas être faite à la seule vue de son monoïde syntaxique. Celui-ci doit en effet avoir la propriété que ses idempotents commutent, mais ce n'est pas suffisant; il faut aussi que la partie du monoïde qui reconnaît le langage respecte une propriété de clôture :

THÉORÈME 3.1 *[51] Soit $\mathcal{L}$ un langage reconnaissable, M son monoïde syntaxique, P une partie de M et φ un morphisme de A^* dans M tels que $\mathcal{L} = P\varphi^{-1}$. Le langage $\mathcal{L}$ est réversible si et seulement si les deux propriétés suivantes sont vérifiées :*

i) *Les idempotents de M commutent.*

ii) *Quels que soient u, v et w appartenant à A^*, si $u.v^+.w$ appartient à $\mathcal{L}$, alors $u.w$ aussi.*

Les automates réversibles peuvent avoir plusieurs états initiaux ou terminaux. On va voir que ce « défaut de réversibilité » peut être étendu dans une certaine mesure, sans altérer les propriétés de l'automate.

Définition 3.2 *Un automate $\mathcal{A} = \langle Q, A, E, I, T \rangle$ est un* **automate quasi-réversible** *si quel que soit le couple de transitions (e_1, e_2) de E étiquetées par la même lettre, ayant même origine ou même destination, ni e_1 ni e_2 n'appartiennent à une composante fortement connexe.*

En d'autres termes, si l'on a l'une des deux configurations suivantes,

l'état r n'appartient pas à la même composante fortement connexe que p ou q, qui eux, peuvent éventuellement faire partie de la même composante fortement connexe.

Remarque 3.1 Un automate localement réversible (*cf.* [36]), c'est-à-dire dont les composantes fortement connexes sont réversibles, n'est pas forcément quasi-réversible. On peut par exemple, dans un automate localement réversible, avoir une des deux configurations précédentes, avec p et r dans la même composante fortement connexe et q en dehors.

Il s'avère que la classe de langages ainsi reconnus n'est pas plus riche que les langages réversibles :

Lemme 3.1 *Le langage reconnu par un automate quasi-réversible est réversible.*

Démonstration. Soit $\mathcal{A}$ un automate quasi-réversible. Soit n le nombre de paires de transitions en contradiction avec l'hypothèse de réversibilité. On montre par récurrence sur n que l'on peut transformer un automate quasi-réversible en un automate réversible. Si n est nul, l'automate $\mathcal{A}$ est réversible. Supposons qu'on sait effectuer la transformation pour tout entier strictement inférieur à n. Soit $\{e_1, e_2\}$ une paire de transitions en contradiction avec l'hypothèse de réversibilité. On considère $\mathcal{A}_1$ et $\mathcal{A}_2$, deux copies de $\mathcal{A}$. On supprime e_1 de $\mathcal{A}_1$ et e_2 de $\mathcal{A}_2$. L'union de ces deux copies reconnaît le même langage que $\mathcal{A}$. Le nombre de paires de transitions qui contredisent la réversibilité est strictement inférieur à n dans chacune des deux copies, on peut donc les transformer en automates réversibles dont l'union est un automate réversible équivalent à $\mathcal{A}$. $\square$

Il se peut que la représentation réversible soit de taille exponentielle par rapport à la représentation quasi-réversible. Ceci a été montré par P.-C. Héam [35] sur l'exemple suivant :

Exemple 13 Soit k un entier positif et $\mathcal{L}_{r_2} = (ab + ba + aa)^{2k}$. Ce langage (fini) est reconnu par l'automate minimal suivant qui est quasi-réversible : P.-C. Héam montre que la taille du plus petit automate réversible qui reconnaît $\mathcal{L}_{r_2}$ est au moins $\left(\frac{3\sqrt{2}}{4}\right)^{\frac{k}{6}}$.

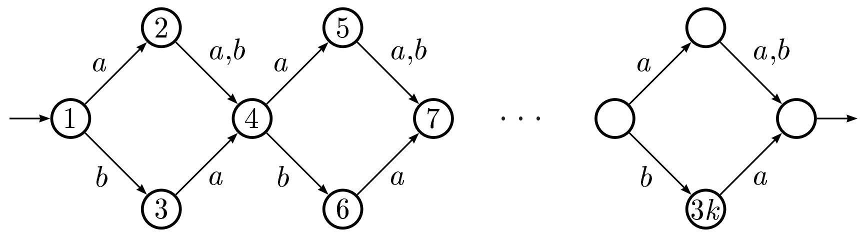


FIG. 1 – *L'automate minimal de* $\mathcal{L}_{r_2}$

On peut donc se demander si, en pratique, une bonne représentation d'un langage réversible n'est pas une représentation quasi-réversible. L'intérêt d'un automate réversible est que la lecture d'un mot, aussi bien par la gauche que par la droite, ne peut se faire que sur un nombre borné de chemins.

On peut définir une mesure de non déterminisme d'un automate de la façon suivante. On considère que l'automate est à multiplicité dans le semi-anneau $(\mathbb{N} \cup \{-\infty\}, \mathsf{max}, +)$; chaque transition a pour valeur 0, sauf celles qui sont non déterministes qui ont valeur 1. Ainsi, la valeur d'un chemin est le nombre de transitions non déterministes qu'on y a rencontré et la valeur d'un mot est le nombre maximal de choix non déterministes qu'on peut avoir à effectuer pour parcourir un calcul étiqueté par ce mot. La mesure de non-déterminisme de l'automate est la borne supérieure des coefficients de la série ainsi réalisée. Notons que cette mesure de non déterminisme est sensiblement différente de celle introduite par I. Simon [59].

On peut, de même, définir une mesure de non co-déterminisme. Il est facile de voir qu'un automate est quasi-réversible si, et seulement si, ses mesures de non déterminisme et de non co-déterminisme sont toutes deux finies.

Le petit lemme suivant est élémentaire. C'est pourtant à lui qu'on recourra dans de nombreuses preuves sur les langages réversibles.

LEMME 3.2 *Soit $\mathcal{A}$ un automate réversible. Il existe un entier strictement positif k tel que, pour tout mot u de A^*, tout chemin étiqueté par u^k est une boucle.*

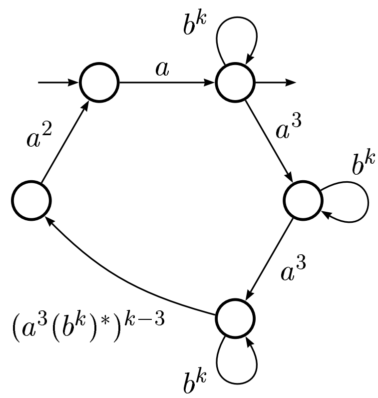
Démonstration. Considérons le monoïde de transition de $\mathcal{A}$. Soit u un mot dont l'image dans ce monoïde est un idempotent. Pour tout état p de $\mathcal{A}$, soit $p \cdot u = \emptyset$, soit $p \cdot u = q$ et, comme l'image de u est un idempotent, $q \cdot u = q$, et, comme $\mathcal{A}$ est réversible, $p = q$. Un idempotent du monoïde de transition de $\mathcal{A}$ est donc une identité partielle, en d'autres termes, tout mot dont l'image dans le monoïde de transition est un idempotent, ne peut étiqueter que des boucles.

Pour tout élément α du monoïde de transition, il existe un entier strictement positif k_α tel que α^{k_α} est un idempotent. En prenant, par exemple, le plus petit multiple commun de tous les k_α, pour α dans le monoïde de transition, on obtient un entier k tel que α^k est un idempotent pour tout α. L'entier k vérifie le lemme. $\square$

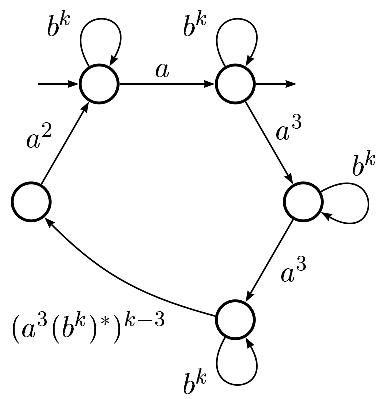
Pour clore cette présentation des langages réversibles, il faut signaler qu'il existe un certain antagonisme entre réversibilité et déterminisme ou même non-ambiguïté :

PROPOSITION 3.3 *Il existe des langages réversibles tels qu'il n'existe aucun automate réversible non ambigu qui reconnaît l'un de ces langages.*

Démonstration. La preuve de cette proposition se présente sous forme d'exemple. En guise d'exercice, on peut montrer que tout automate réversible qui reconnaît le langage $a^*ba^* + b^*ab^*$ compte au moins deux calculs étiquetés par ab (de même pour ba). On va ici développer un autre exemple dans lequel l'ensemble des mots étiquetant plusieurs chemin est infini. Considérons le langage reconnu par l'automate réversible $\mathcal{A}_{r_1}$ présenté exemple 12.1 page 71. Les langages $a(b^*a^3)^*$ et $b^*(a^3)^*a$ sont des sous-ensembles du langage accepté par $\mathcal{A}_{r_1}$. Soit $\mathcal{B}$ un automate réversible qui accepte $\mathcal{L}_{r_1}$. Il existe un entier k tel que, quel que soit le mot u de A^*, tout chemin de $\mathcal{B}$ étiqueté par u^k est une boucle. Le mot $a(b^ka^3)^k$ est accepté par $\mathcal{B}$. La configuration suivante apparaît donc dans $\mathcal{B}$:



Si le calcul étiqueté par a^{3k+1} est unique dans $\mathcal{B}$, comme $b^k(a^{3k+1})$ est aussi accepté par $\mathcal{B}$, on obtient la configuration suivante qui accepte un langage strictement plus gros que $\mathcal{L}$:



Donc, il existe au moins deux calculs étiquetés par a^{3k+1} dans $\mathcal{B}$, et, tout mot de $a(a^{3k})^*$ étiquette au moins deux chemins. □

— ◦ —

2 Langages à groupe

Une sous-classe importante des langages réversibles est celle des langages à groupes :

Définition 3.3 *Un automate $\mathcal{A}$ est un **automate à groupe** s'il est réversible et complet. Un langage $\mathcal{L}$ est un **langage à groupe** s'il peut être accepté par un automate à groupe.*

Remarque 3.2 Dans un automate à groupe, chaque lettre induit une bijection de l'ensemble des états sur lui-même, le monoïde de transition d'un tel automate est donc un groupe.

Lemme 3.4 *L'automate minimal d'un langage à groupe est un automate à groupe.*

Démonstration. Le monoïde de transition de l'automate minimal d'un langage est le monoïde syntaxique du langage, qui est le quotient du monoïde de transition de n'importe quel automate qui accepte le langage. Un monoïde quotient d'un groupe est un groupe, donc l'automate minimal d'un langage à groupe est un automate à groupe. □

Les propriétés très particulières des langages à groupe permettent de les étudier de façon bien plus simple que les langages réversibles. Avant d'entamer l'étude des automates universels de langages réversibles, nous allons donc, en guise de hors d'œuvre, étudier les automates universels de langages à groupe, ce qui, espérons-le, facilitera la digestion du plat principal.

— ∘ —

3 Automate universel d'un langage à groupe

3.1 Structure générale de l'automate universel d'un langage à groupe

L'automate minimal d'un langage à groupe est un automate à groupe. Afin d'exploiter ses propriétés lors de l'étude de l'automate universel, nous considérons que ce dernier est construit selon la méthode exposée dans le théorème 2.1 et que ses états sont indexés par des sous-ensemble de Q (ensemble des états de l'automate minimal). Dans ce qui suit, un état p de $\mathcal{U}_{\mathcal{L}}$ sera donc un sous-ensemble de Q.

Définition 3.4 *Soit $\mathcal{U}_{\mathcal{L}}$ l'automate universel d'un langage à groupe $\mathcal{L}$. On dit qu'un état p appartient à l'**étage** k de $\mathcal{U}_{\mathcal{L}}$ si $\mathsf{Card}(p) = k$.*

Les étages ainsi définis forment une véritable hiérarchie dans l'automate universel : on peut dire qu'un étage est supérieur à un autre s'il existe une transition entre un état du premier et un état du second. La proposition suivante assure que cette relation est bien une relation d'ordre. Toutefois, cet ordre n'est généralement pas total, comme le montre l'exemple 15.

Proposition 3.5 *Soit p_1 et p_2 deux états de l'automate universel $\mathcal{U}_{\mathcal{L}}$ d'un langage à groupe $\mathcal{L}$ appartenant respectivement aux étages k_1 et k_2. S'il y a une transition de p_1*

à p_2 dans $\mathcal{U}_{\mathcal{L}}$, alors $k_1 \leqslant k_2$.
En particulier, si p_1 et p_2 sont dans la même composante fortement connexe, ce sont deux états du même étage.

Démonstration. Soit p_1 et p_2 deux états de $\mathcal{U}_{\mathcal{L}}$ et a une lettre de A qui étiquette une transition entre ces deux états. Par définition de l'automate universel, $p_1 \cdot a \subseteq p_2$. Comme l'automate minimal est complet et que chaque lettre agit fidèlement sur les états, $\mathsf{Card}(p_1 \cdot a) = \mathsf{Card}(p_1)$, donc $\mathsf{Card}(p_1) \leqslant \mathsf{Card}(p_2)$. □

La proposition suivante résume les propriétés de ces étages :

Proposition 3.6 *Soit $\mathcal{U}_{\mathcal{L}}$ l'automate universel d'un langage à groupe et son automate minimal $\mathcal{A} = \langle\, Q,A,E,\{i\},T\,\rangle$.*
i) *Le premier étage est isomorphe à $\mathcal{A}$.*
ii) *Le dernier étage est isomorphe à l'automate co-déterministe minimal.*
iii) *Le nombre d'étages est au plus égal au cardinal de T.*
iv) *Si deux états appartiennent au même étage, soit ils ne sont pas accessibles l'un-l'autre, soit ils sont dans la même composante fortement connexe.*

Démonstration. i) Il n'est en effet pas difficile de voir que le morphisme qui envoie un état p de $\mathcal{A}$ sur l'état $\{p\}$ de $\mathcal{U}_{\mathcal{L}}$ est bijectif sur le premier étage.

ii) Il suffit de considérer l'image miroir du langage et d'utiliser le résultat de la proposition 2.8.

iii) Puisque les représentants des factorisations sont calculés en utilisant les ensembles $u \cdot T$, pour $u \in A^*$ et que, dans un automate à groupe, chacun de ces ensembles est de même cardinal que T, le cardinal des états de l'automate universel est compris entre 1 et le cardinal de T.

iv) Soit p_1 et p_2 deux états appartenant au même étage. Ils ont donc même cardinal. S'il existe un chemin de p_1 à p_2 étiqueté par u, alors $p_1 \cdot u \subseteq p_2$; comme l'automate minimal est un automate à groupe, $p_1 \cdot u = p_2$. Il existe v dans A^* tel que l'action de $u.v$ sur Q est l'identité, donc $p_1 \cdot (u.v) = p_2 \cdot v = P$; il existe donc un chemin de l'état p_1 à p_2 étiqueté par v. □

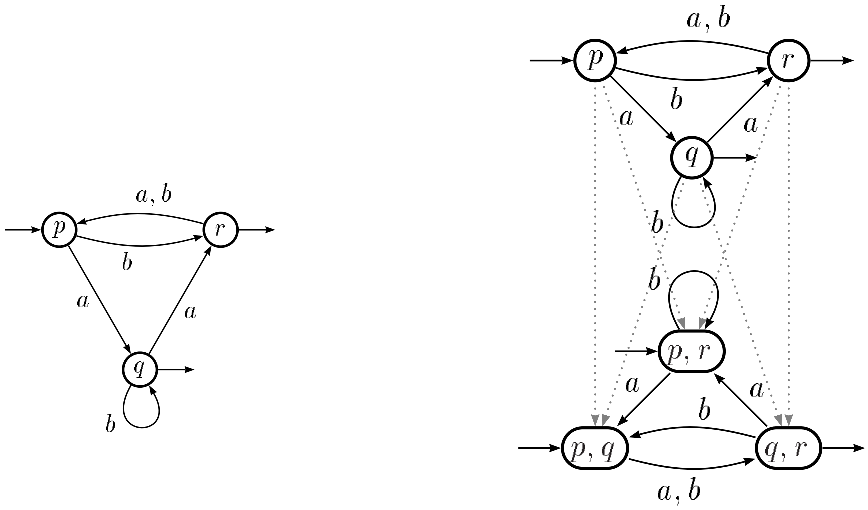
Corollaire 3.7 *Soit $\mathcal{A}$ un automate à groupe émondé ayant un état initial et un état final. Alors $\mathcal{A}$ est à la fois l'automate minimal et universel du langage qu'il reconnaît.*

Démonstration. Comme l'automate est en même temps déterministe et co-déterministe, il est minimal. D'après la proposition précédente, l'automate universel n'a qu'un étage qui est isomorphe à l'automate minimal. □

Les automates universels présentés ci-après sont écorchés. Toutefois, par souci de clarté, on indique tous les états initiaux et terminaux.

Exemple 14 Cet exemple et le suivant sont des langages dont le monoïde syntaxique est le groupe $\mathfrak{S}_3$. On verra que la structure de l'automate universel (ainsi d'ailleurs que celle de l'automate minimal) ne dépend pas seulement du monoïde syntaxique mais aussi de la partie qui reconnaît le langage. Les états des automates universels sont étiquetés par les ensembles d'états des automates minimaux auxquels ils correspondent. Le premier exemple est le langage $\mathcal{L}_{g_1}$ reconnus par les automates de la figure 2. On peut remarquer que l'automate minimal n'est pas le graphe de Cayley du groupe syntaxique. On voit que les étages de l'automate universel dépendent bien du nombre d'états terminaux de l'automate minimal et non pas du nombre d'éléments du groupe qui forment la partie reconnaissant le langage.

Exemple 15.1 La structure des automates reconnaissants le langage $\mathcal{L}_{g_2}$ est sensiblement différente, bien que les lettres a et b représentent les mêmes générateurs de $\mathfrak{S}_3$.



(a) L'automate minimal de $\mathcal{L}_{g_1}$. (b) L'automate universel de $\mathcal{L}_{g_1}$.

Fig. 2 – *Deux automates reconnaissant* $\mathcal{L}_{g_1}$.

L'automate universel du langage $\mathcal{L}_{g_2}$, présenté figure 4 comporte quatre étages. Les transitions spontanées entre les étages sont représentées par des arcs en pointillés gris. On voit que les étages ne sont pas totalement ordonnés : il n'y a pas de transition entre l'étage 2 (composé de trois sous-automates « en triangle ») et l'étage 3 (formé des états « *pqr* » et « *stu* »).

— ◦ —

3.2 Structure des composantes fortement connexes de l'automate universel

Les composantes fortement connexes de l'automate universel d'un langage à groupe sont très particulières.

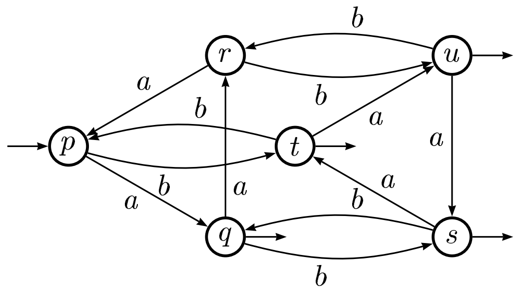


Fig. 3 – *L'automate minimal de* $\mathcal{L}_{g_2}$

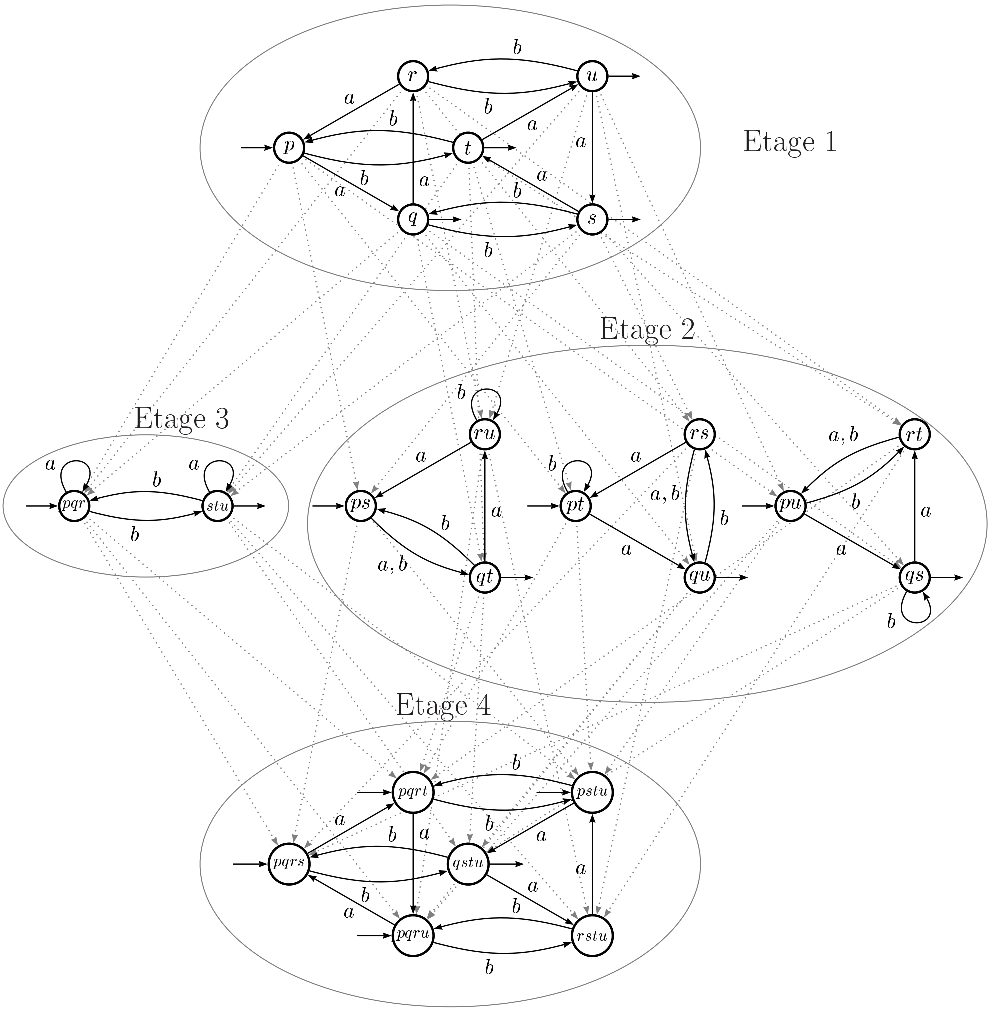


Fig. 4 – *L'automate universel de* $\mathcal{L}_{g_2}$

Chaque état d'un automate universel correspond à un sous-ensemble d'états de l'automate minimal. Comme l'automate minimal est lui-même l'image par morphisme du graphe de Cayley du monoïde syntaxique, chacun de ses états correspond à un sous-ensemble d'éléments du monoïde syntaxique.

Dans le cas des langages à groupe, l'action des lettres sur les états du graphe de Cayley est fidèle et transitive. Comme les états de l'automate minimal sont des ensembles d'états du graphe de Cayley, l'automate minimal, qui est l'orbite de l'état initial sous l'action des lettres, est, nous l'avons déjà vu, un automate à groupe (avec une seule composante fortement connexe).

Pour la même raison, chaque composante fortement connexe de l'automate universel représente l'orbite d'un sous-ensemble d'état de l'automate minimal sous l'action des lettres et est donc un automate à groupe.

THÉORÈME 3.2 *Soit $\mathcal{U}_\mathcal{L}$ l'automate universel d'un langage à groupe $\mathcal{L}$. Chaque pelote de $\mathcal{U}_\mathcal{L}$ est un sous-automate à groupe de $\mathcal{U}_\mathcal{L}$. Le monoïde de transition de chaque pelote de $\mathcal{U}_\mathcal{L}$ est un quotient du groupe syntaxique. Tout état de $\mathcal{U}_\mathcal{L}$ appartient à une pelote.*

Démonstration. Soit p un état de $\mathcal{U}_\mathcal{L}$ et u un mot de A^*. Il existe un mot v tel que l'action de $u.v$ sur les états de l'automate minimal est l'identité. Donc $p \cdot (u.v) = p$ et il existe une boucle étiquetée par $u.v$ autour de p. A partir de p, on peut donc lire n'importe quel mot u sans sortir de la composante fortement connexe de p : celle-ci est donc complète. D'autre part, puisqu'on n'a fait aucune supposition sur p, il suit que tout état de p appartient à une pelote.

L'état q d'une pelote atteint à partir d'un autre état p de la même pelote, en lisant un mot u, fait partie du même étage, donc $q = p \cdot u$, ce qui caractérise uniquement q à partir de p et u; la pelote est donc déterministe (du point de vue de ses transitions). La pelote est aussi co-déterministe, puisque l'image miroir d'un langage à groupe est un langage à groupe.

Soit $\mathcal{P}$ une pelote de l'automate universel, Q' son ensemble d'états et $G' \subseteq \mathsf{S}_{Q'}$ son groupe de transition. Si G est le groupe syntaxique, c'est-à-dire le monoïde de transition de l'automate minimal, il existe un morphisme surjectif de G sur G' :

$$\begin{array}{ccc} G & \longrightarrow & G' \\ \\ \alpha & \longmapsto & \alpha' : \begin{array}{ccc} Q' & \to & Q' \\ p & \mapsto & \{x\alpha \mid x \in p\} \end{array} \end{array}$$

$\square$

REMARQUE 3.3 Le fait que le groupe de transition d'une pelote soit le quotient du groupe de transition de l'automate minimal ne signifie pas que la pelote, en tant que graphe, est un quotient de l'automate minimal. Ce phénomène est illustré par l'exemple de la figure 5.

Pour conclure cet examen des automates universels des langages à groupe, on montre que l'écorché de l'automate universel contient les mêmes pelotes que l'automate universel.

PROPOSITION 3.8 *Soit $\mathcal{L}$ un langage à groupe. Les pelotes formées par les transitions non spontanées de l'écorché de l'automate universel $\check{\mathcal{U}}_\mathcal{L}$ sont exactement les pelotes de l'automate universel $\mathcal{U}_\mathcal{L}$. Si on ne conserve dans l'automate universel que les transitions des pelotes, on obtient un automate à groupe qui reconnaît $\mathcal{L}$.*

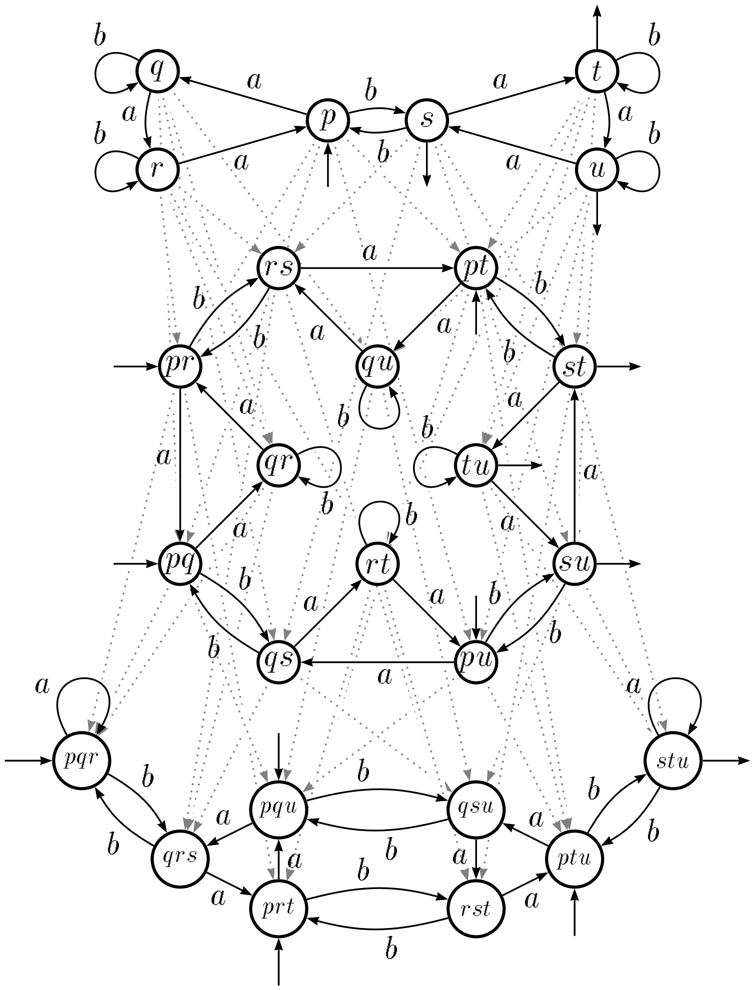


FIG. 5 – *L'automate universel d'un langage à groupe.*

Démonstration. Soit (p,a,p') une transition d'une pelote de $\mathcal{U}_{\mathcal{L}}$. On a vu que $p \cdot a = p'$. Soit (L,R) et (L',R') les factorisations qui correspondent respectivement à p et à q. On a :

$$R' = \bigcap_{x \in p'} \mathsf{Fut}_{\mathcal{A}}(x) = \bigcap_{x \in p \cdot a} \mathsf{Fut}_{\mathcal{A}}(x).$$

Donc (L,R') est bien un couple maximal tel que $L.a.R'$ est inclus dans $\mathcal{L}$. Donc cette transition appartient bien à $\check{\mathcal{U}}_{\mathcal{L}}$.

Soit (p,a,p') une transition n'appartenant pas à une pelote de $\mathcal{U}_{\mathcal{L}}$. Donc $p \cdot a$ est strictement inclus dans p'. Il existe $p' = p \cdot a$ qui correspond à un état dont le facteur gauche est strictement inclus dans celui de p', donc avec un facteur droit strictement plus grand que celui de p'. Le couple (L,R'') (avec des notations évidentes) est supérieur au couple (L,R'). La transition (p,a,p') n'apparaît donc pas dans $\check{\mathcal{U}}_{\mathcal{L}}$.

Le dernier point est trivial, on a vu que chaque pelote est un automate à groupe; leur union forme donc un automate à groupe qui contient l'automate minimal et reconnaît donc tout le langage. □

— ◦ —

4 Automate universel d'un langage réversible

4.1 Structure générale de l'automate universel d'un langage réversible

L'automate minimal d'un langage réversible n'est pas nécessairement réversible. Cette simple constatation rend les choses, certes plus intéressantes, mais aussi un tantinet plus délicates. Indexer les états de l'automate universel en fonction de l'automate minimal ne paraît en effet pas judicieux pour exploiter les propriétés de réversibilité du langage. C'est donc un automate réversible qui reconnaît le langage qui nous servira de référence.

Il faut bien souligner qu'*a priori*, on ne connaît rien de l'automate réversible de référence si ce n'est qu'il est réversible (évidemment) et qu'il reconnaît le langage. Cet automate inconnu va nous apporter des informations sur l'automate universel qui, lui, est connu et constructible à partir de l'automate minimal.

Les états de l'automate minimal d'un langage à groupe nous permettaient d'indexer ceux l'automate universel car ils étaient les images de i sous l'action des mots de A^*. Dans le cas d'un automate réversible $\mathcal{A}$, ce n'est donc pas tant les états qui nous intéresseront, que les images de l'ensemble des états initiaux (I) sous l'action des mots de A^*. Encore une fois, nous sommes ramenés aux états du déterminisé et indexer les états de l'automate universel ne consistera en rien d'autre que leur associer les états du développé $\mathcal{A}$ par la fonction donnée à la proposition 2.18.

Ce sont donc des ensembles d'ensembles qui indexent désormais les états de l'automate universel.

On ne peut donc plus décrire un étage à l'aide d'un simple entier. Toutefois, cette notion demeure.

DÉFINITION 3.5 *Soit* $\mathcal{A} = \langle\, Q,A,E,\{i\},T\,\rangle$ *un automate réversible reconnaissant un langage* $\mathcal{L}$.

Le ***profil*** *d'un état* P *du développé* $\mathcal{V}_\mathcal{A}$ *est une fonction* κ_P *de* $[1;\mathsf{Card}(Q)]$ *dans* $\mathbb{N}$ *qui, à tout* n *associe le nombre d'éléments de* P *de cardinal* n. *On dit qu'un état* P *appartient à l'****étage*** κ *de* $\mathcal{V}_\mathcal{A}$ *si* κ *est le profil de* P.

EXEMPLE 12.2 Considérons le développé de $\mathcal{A}_{r_1}$ présenté dans l'exemple 12. Voici les profils de quelques-uns de ses états :

Etats	$\{pq,pr\}$	$\{p,qr\}$	$\{p,q,r\}$	$\{pqr\}$
Profils	[0,2,0]	[1,1,0]	[3,0,0]	[0,0,1]

On va voir que, de même que dans le cas des groupes, les étages ainsi définis sont ordonnés :

PROPOSITION 3.9 *Soit* P_1 *et* P_2 *deux états du développé* $\mathcal{V}_\mathcal{A}$ *d'un automate réversible* $\mathcal{A}$ *appartenant respectivement aux étages* κ_1 *et* κ_2. *S'il y a une transition de* P_1 *à* P_2 *dans l'automate universel de* $\mathcal{L}$, *alors* $\kappa_1 \leqslant_{\mathsf{lex}} \kappa_2$.
En particulier, si p_1 *et* p_2 *sont dans la même composante fortement connexe, ce sont deux états du même étage.*

Démonstration. Soit (P,a,Q) une transition de $\mathcal{V}_{\mathcal{A}}$. Considérons la fonction qui à X appartenant à P associe $X \cdot a$. On peut séparer les éléments de P en deux parties. D'une part, ceux pour lesquels X et $X \cdot a$ ont le même cardinal; à cause de la réversibilité, la fonction est une bijection sur ce sous-ensemble de P. D'autre part, ceux pour lesquels le cardinal de X est strictement supérieur au cardinal de $X \cdot a$; alors, à cause de la réversibilité et parce que les éléments de P sont incomparables, il n'existe pas X' de même cardinal que $X \cdot a$ tel que $X' \cdot a = X \cdot a$ (ce qui entraînerait $X' \subseteq X$). Donc κ_Q est lexicographiquement plus grand que κ_P. □

Exemple 12.3 L'écorché de l'automate universel du langage réversible $\mathcal{L}_{r_1}$ est représenté figure 7. Là encore, on a indiqué tous les états initiaux et terminaux.

Les états sont disposés de haut en bas selont leur signature, de « *pq* », « *qr* » et « *pq* » dont la signature est [0,1,0], à « $\{p,q\}$ », « $\{q,r\}$ » et « $\{p,q\}$ » dont la signature est [2,0,0]. Les états « $\{pqr\}$ » et « $\{p,q,r\}$ » du développé de $\mathcal{A}_{r_1}$ ne sont respectivement pas accessible et co-accessible, ils ne peuvent donc pas faire partie de l'automate universel. On vérifie que les autres états appartiennent bien à $\mathcal{U}_{\mathcal{L}_{r_1}}$.

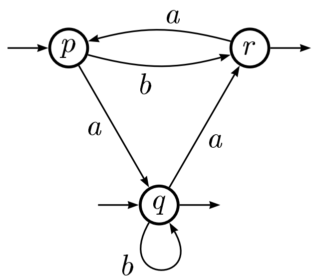


Fig. 6 – *Un automate réversible acceptant* $\mathcal{L}_{r_1}$

Dans certains cas particuliers, on peut aussi utiliser la construction de l'automate universel à partir de l'automate minimal pour retrouver certains résultats. Ainsi, le corollaire 3.7 pour les automates à groupe s'applique partiellement aux automates réversibles :

Proposition 3.10 *Soit $\mathcal{A}$ un automate réversible émondé ayant un état initial et un état final. Alors $\mathcal{A}$ est l'automate minimal et universel du langage.*

Démonstration. De même que pour les langages à groupes, l'automate est en même temps déterministe et co-déterministe, il est donc minimal. Les états de l'automate universel sont indexés par la clôture par intersection des états du co-déterminisé, c'est-dire, en l'occurrence, par les singletons. Il existe une transition $(\{p\},a,\{q\})$ dans l'automate universel si et seulement si $p \cdot a = q$, donc si et seulement s'il existe une transition (p,a,q) dans l'automate minimal. De même pour les états initiaux et terminaux. □

— ∘ —

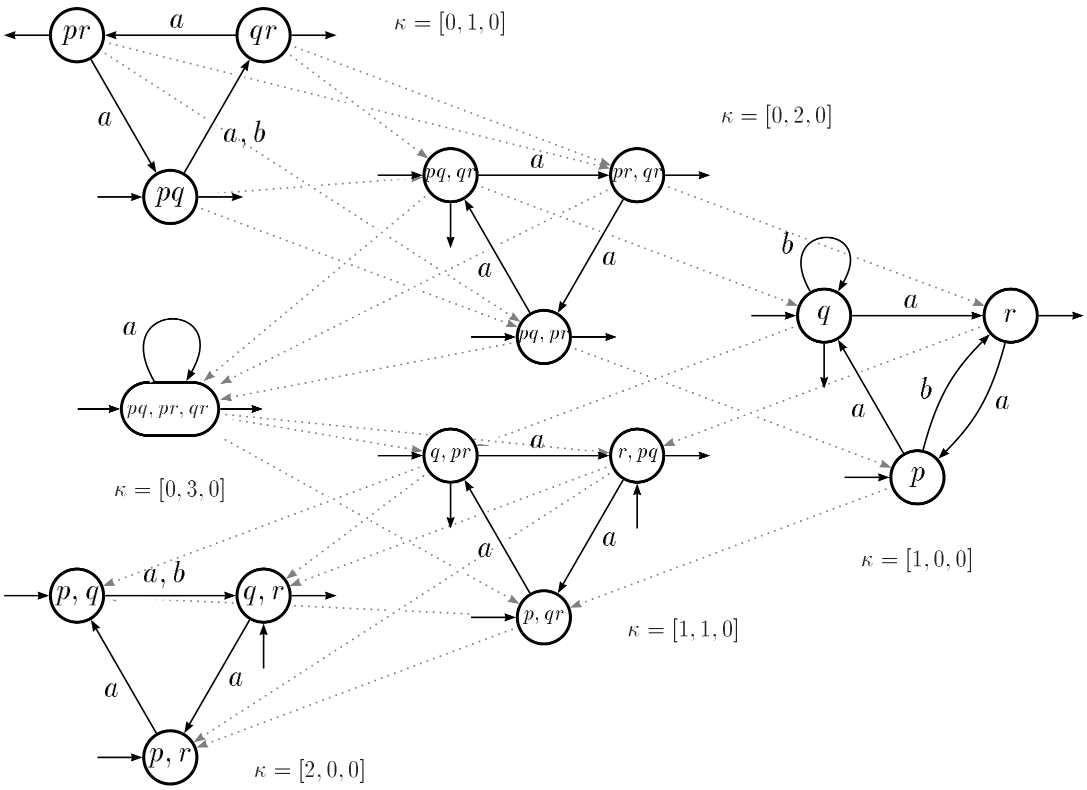


FIG. 7 – *L'automate universel de $\mathcal{L}_{r_1}$ (écorché)*

4.2 Structure des pelotes de l'automate universel

De même que chaque pelote de l'automate universel d'un langage à groupe est un automate à groupe, pour les langage réversibles, la propriété se transmet. Les deux énoncés suivants correspondent au théorème 3.2. Ils sont énoncés pour le développé d'un automate réversible. La propriété pour l'automate universel en est un simple corollaire.

PROPOSITION 3.11 *Soit $\mathcal{L}$ un langage réversible, $\mathcal{A}$ un automate réversible qui accepte $\mathcal{L}$ et $\mathcal{V}_\mathcal{A}$ son développé. Soit P et Q les sous-ensembles représentant deux états p et q d'une même pelote de l'automate $\mathcal{V}_\mathcal{A}$. Alors, pour tout mot u (resp. v) étiquetant un chemin de p à q (resp. de q à p), $P \cdot u = Q$ et $Q \cdot v = P$.*

Démonstration. D'après la proposition 3.9 les états d'une même pelote ont le même profil. On montre que u induit une bijection entre les éléments de P de cardinal i et ceux de Q de même cardinal. C'est-à-dire que quel que soit X dans P, il existe Y dans Q, de même cardinal que X tel que $X \cdot u = Y$. De même, on montre que v induit une bijection de Q sur P.
Si cette propriété est fausse, soit i minimal contredisant cette affirmation. Il existe X appartenant à P, de cardinal i et Y appartenant à Q de cardinal j tel que $Y \subset X \cdot u$. On

a donc $\mathsf{Card}(Y) = j < i$; il existe donc X_j appartenant à P tel que $X_j \cdot u = Y \subset X \cdot u$. Comme l'automate $\mathcal{A}$ est réversible, ceci entraîne $X_j \subset X$ et contredit le fait que P est une anti-chaîne. □

Théorème 3.3 *Les pelotes de l'automate universel d'un langage réversible sont réversibles.*

Démonstration. La proposition 3.11 montre que l'état d'une pelote du développé d'un automate réversible atteint à partir d'un autre état de la même pelote est caractérisé par l'étiquette du chemin qui les relie. Les pelotes de l'automate développé sont donc déterministes [1]. Comme l'automate universel est un sous-automate du développé, ses pelotes sont déterministes. De plus, l'image miroir d'un langage réversible est un langage réversible; les pelotes de l'automate transposé sont donc déterministes. Par conséquence, les pelotes de l'automate universel d'un langage réversible sont aussi co-déterministes. Finalement, elles sont réversibles. □

Remarque 3.4 Le fait que les pelotes de l'automate universel soient réversibles n'est pas caractéristique des langages réversibles. En fait, ceci entraîne seulement que le point ii) de la caractérisation du théorème 3.1 est vérifié. Par exemple, les pelotes de l'automate universel du langage a^*b^*, dans le monoïde syntaxique duquel les idempotents ne commutent pas, sont réversibles.

Les pelotes de l'automate universel d'un langage réversible sont fortement liées à la structure du monoïde syntaxique de ce langage :

Proposition 3.12 *Soit $\mathcal{U}_{\mathcal{L}}$ l'automate universel d'un langage réversible et p un état appartenant à une pelote $\mathcal{P}$ de $\mathcal{U}_{\mathcal{L}}$. Soit u un mot correspondant à un idempotent du monoïde syntaxique. Alors, s'il existe un chemin dans $\mathcal{P}$ partant de p étiqueté par u, ce chemin est une boucle autour de p étiquetée par u.*

Démonstration. Si u correspond à un idempotent, comme le futur d'un état de l'automate universel est l'image inverse d'une partie du monoïde syntaxique, s'il existe un mot v tel que $u.v$ est dans le futur de p, alors pour tout entier k, le mot $u^k.v$ est dans le futur de p. On suppose qu'il existe un chemin, interne à la pelote de p, qui part de p étiqueté par un mot u correspondant à un idempotent. Par le lemme des tiroirs, il existe deux entiers i,j ($j > 0$) et un état $q = (L,R)$ tels que u^i étiquette un chemin de p à q et que u^j étiquette une boucle autour de q. Donc $L.u^j \subseteq L$, et comme u correspond à un idempotent et que les facteurs sont reconnus par le monoïde syntaxique, on a $L.u \subseteq L$ et donc, le mot u étiquette une boucle autour de q. Les pelotes sont réversibles donc en particulier co-déterministes, le chemin de p à q étiqueté par u^i est donc i fois la boucle autour de q et $p = q$. □

— ◦ —

1. En ce qui concerne les transitions, on fait ici abstraction des états initiaux ou terminaux

5 Construction d'un automate réversible

Nous allons montrer ici que l'automate universel d'un langage réversible contient un sous-automate quasi-réversible qui reconnaît le même langage. Comme d'une part, on sait construire l'automate universel à partir de l'automate minimal et que, d'autre part, on a vu que la transformation d'un automate quasi-réversible en un automate réversible est simple, même si elle peut se révéler avoir une complexité exponentielle; ceci permet de construire un automate réversible pour un langage réversible donné.

Pour obtenir ce résultat, nous allons considérer un automate réversible (*a priori* inconnu) que l'on met sous une forme particulière, de telle façon que son image dans l'automate universel est quasi-réversible. Pour ce faire, on doit introduire des automates d'apparence singulière.

5.1 Cordes

Si on quotiente chaque composante fortement connexe d'un automate, on obtient un automate orienté acyclique. On peut alors séparer les différents chemins (réussis) de ce graphe.

Une fois ces chemins séparés, comme on est particulièrement prudent et qu'on a une bonne mémoire, on peut remplacer chaque sommet par la composante fortement connexe dont il est le quotient en prenant soin que les états extrémaux des transitions soient corrects. C'est cette manipulation que nous allons détailler dans les propositions suivantes :

Définition 3.6 *Un automate $\mathcal{A} = \langle Q,A,E,I,T \rangle$ est une* ***corde***[2] *s'il est émondé, n'a qu'un seul état initial, un seul état final et si il y a au plus une transition arrivant sur chaque composante fortement connexe*[3] *et au plus une transition partant de chaque composante fortement connexe. Un automate est un* ***quipu***[4] *si c'est une union finie disjointe de cordes.*

Proposition 3.13 *Tout automate $\mathcal{A}$ émondé est équivalent à un quipu $\mathcal{B}$ tel qu'il existe un morphisme surjectif de $\mathcal{B}$ sur $\mathcal{A}$.*

Démonstration. La preuve est la même que pour le passage d'un automate quasi-réversible à un automate réversible. Au lieu de considérer les paires de transitions qui contredisent la réversibilité, on considère les paires de transitions qui contredisent le fait d'être une corde. En dupliquant ainsi l'automate pour chacune de ces paires et en supprimant dans chaque copie l'une des deux transition, on obtient finalement un quipu qui est équivalent à $\mathcal{A}$. Cette construction est illustrée par la figure 9. □

Remarque 3.5 La construction précédente a une complexité déplorable, puisque c'est une opération encore plus brutale que celle qui permet de transformer un automate quasi-réversible en réversible, cette dernière étant déjà exponentielle. Nous ne l'utiliserons que

2. à noeuds, évidemment.

3. au sens large

4. Instrument inca formé d'une collection de cordelettes sur lesquelles une succession de nœuds permettait, selon leur nombre et leur forme, de tenir une comptabilité.

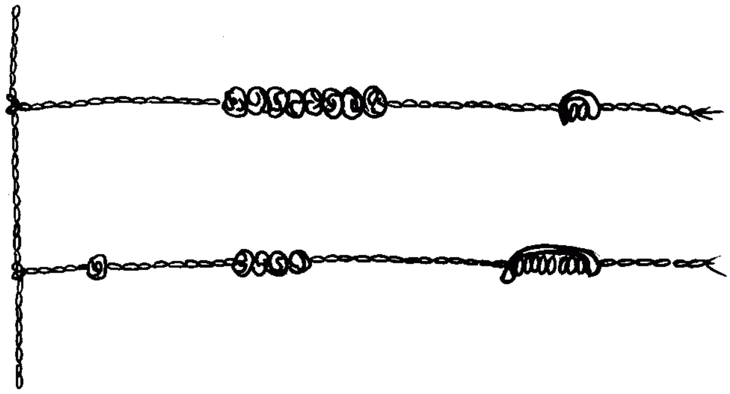


FIG. 8 – *Schéma d'un quipu inca.*

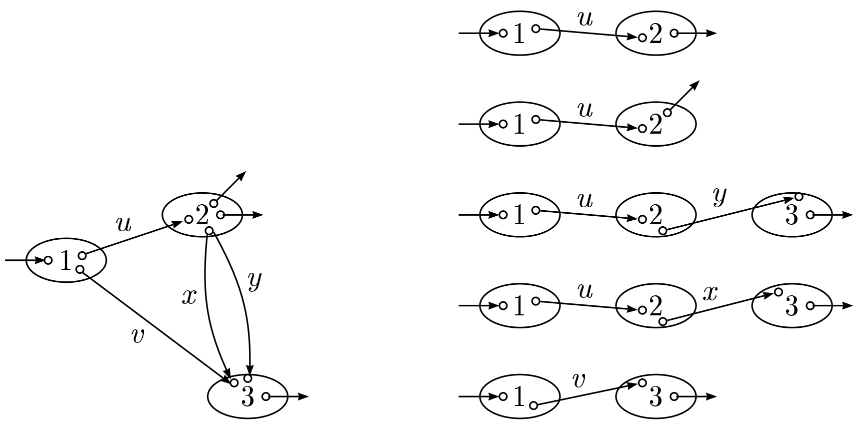


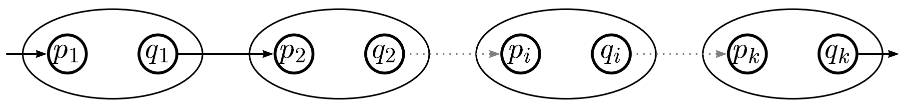
FIG. 9 – *D'un automate à un quipu.*

de façon théorique, pour obtenir un quipu ayant des propriétés similaires à l'automate de départ. Par exemple, si l'automate de départ est déterministe, chaque composante connexe du quipu l'est aussi.

— ∘ —

5.2 Automate quasi-réversible et automate universel

Nous avons montré dans le lemme 3.1 que tout automate quasi-réversible est équivalent à un quipu réversible. Ceci est vrai, en particulier, pour tout automate réversible. Nous allons utiliser dans cette partie cette forme particulière. Chaque corde d'un quipu a la forme suivante :



Si l'automate est en plus réversible, chacune de ces corde est déterministe et co-déterministe.

Noter que les états p_i et q_i jouent un rôle particulier : tout calcul de l'automate les emprunte obligatoirement.

DÉFINITION 3.7 *Un quipu est minimal si on ne peut supprimer aucune de ses cordes sans diminuer le langage reconnu.*

PROPOSITION 3.14 *Soit $\mathcal{L}$ un langage réversible et $\mathcal{A}$ un quipu réversible minimal qui reconnaît $\mathcal{L}$. Soit φ un morphisme de $\mathcal{A}$ dans l'automate universel $\mathcal{U}_\mathcal{L}$. Alors, l'image de chaque composante fortement connexe de $\mathcal{A}$ est une composante fortement connexe de $\mathcal{U}_\mathcal{L}$.*

Démonstration. On montre que, si ce n'est pas le cas, le quipu n'est pas minimal.

S'il y a une pelote $\mathcal{P}$ de $\mathcal{A}$ dont l'image par φ n'est pas une pelote de $\mathcal{U}_\mathcal{L}$, la pelote de $\mathcal{U}_\mathcal{L}$ qui contient $\mathcal{P}\varphi$ contient une transition e qui n'appartient pas à l'image de $\mathcal{P}$.

Soit $\mathcal{B}$ la corde qui contient $\mathcal{P}$. Tout mot reconnu par $\mathcal{B}$ peut se décomposer en $u.v$, avec u et v respectivement passé et futur d'un état r de la pelote $\mathcal{P}$.

Dans $\mathcal{P}\varphi$, il existe une boucle autour de $r\varphi$, étiquetée par un mot w, qui emprunte la transition e. Comme l'automate universel reconnaît $\mathcal{L}$ et que u (*resp.* v) appartient au passé (*resp.* au futur) de $r\varphi$, pour tout entier n, le mot $u.w^n.v$ appartient à $\mathcal{L}$. Comme l'automate $\mathcal{A}$ est réversible, il existe un entier k tel que, pour tout état p de $\mathcal{A}$, soit w^k étiquette une boucle autour de $\mathcal{A}$, soit il n'y a pas de chemin étiqueté par w^k qui part de p. Le mot $u.w^k.v$ est accepté par $\mathcal{A}$. S'il était accepté par $\mathcal{B}$, w^k étiquetterait une boucle autour de r dans $\mathcal{P}$ et, puisque les pelotes de l'automate universel sont déterministes, la transition e appartiendrait à $\mathcal{P}\varphi$. Donc il existe une autre corde $\mathcal{B}'$ qui accepte $u.w^k.v$, et, puisque w^k étiquette nécessairement une boucle, qui accepte aussi $u.v$. Tout mot de $\mathcal{B}$ est donc ainsi accepté par une autre corde, ce qui contredit la minimalité du quipu $\mathcal{A}$. □

PROPOSITION 3.15 *Soit $\mathcal{L}$ un langage réversible et $\mathcal{A}$ un quipu réversible minimal qui reconnaît $\mathcal{L}$. Soit φ un morphisme de $\mathcal{A}$ dans $\mathcal{U}_\mathcal{L}$. Alors, l'image de chaque corde de $\mathcal{A}$ dans l'automate universel est réversible.*

Démonstration. Par l'absurde, si ce n'est pas vrai, il existe deux transitions e_1 et e_2 d'une corde $\mathcal{B}$, qui ont la même étiquette, et dont les images ont soit le même état de départ soit le même état d'arrivée. On peut, sans perte de généralité, supposer que c'est la première configuration qui se présente.

Étant donnée la forme d'une corde, toute paire de transitions d'une corde est telle qu'il existe un chemin qui passe par ces deux transitions. Supposons que w est un mot qui étiquette un chemin dont la première transition est e_1 et qui se termine dans l'état de départ de e_2. L'image de ce chemin par φ est une boucle (puisque $e_1\varphi$ et $e_2\varphi$ ont le même état de départ. La transition $e_1\varphi$ appartient donc à une pelote, et comme les pelotes de $\mathcal{U}_\mathcal{L}$ sont réversibles, $e_2\varphi$ n'appartient à aucune pelote, donc e_2 non plus. Donc tout calcul de $\mathcal{B}$ emprunte e_2. Tout mot accepté par $\mathcal{B}$ se décompose en $u.v$, où v étiquette la fin d'un calcul dont la première transition est e_2.

La fin de la preuve est identique à celle de la proposition 3.14. Pour tout entier n, le mot $u.w^n.v$ appartient au langage. Pour k identique à celui de la preuve précédente, le mot $u.w^k.v$ n'étiquette pas un chemin dans $\mathcal{B}$, sinon e_1 et e_2 auraient le même état de départ, ce qui contredirait la réversibilité. Il existe donc une corde $\mathcal{B}'$ qui accepte $u.w^n.v$ et aussi $u.v$, ce qui contredit la minimalité de $\mathcal{A}$. □

— ∘ —

Le théorème principal de cette partie apparaît comme un corollaire des deux propositions précédentes.

THÉORÈME 3.4 *Il existe un sous-automate de l'automate universel d'un langage réversible, qui est quasi-réversible et qui reconnaît tout le langage.*

Démonstration. On montre que l'image d'un quipu réversible minimal, qui reconnaît le langage $\mathcal{L}$, dans $\mathcal{U}_\mathcal{L}$ est quasi-réversible, quel que soit le morphisme.

Soit $\mathcal{A}$ un quipu réversible minimal qui reconnaît $\mathcal{L}$ et φ un morphisme de $\mathcal{A}$ dans $\mathcal{L}$. L'image de chaque corde de $\mathcal{A}$ est réversible. Supposons $\mathcal{A}\varphi$ non quasi-réversible. Il existe alors deux transitions e_1 et e_2 qui appartiennent respectivement à deux cordes $\mathcal{B}_1$ et $\mathcal{B}_2$ de $\mathcal{A}$ et telles que $e_1\varphi$ et $e_2\varphi$ contredisent la quasi-réversibilité. L'une de celles-ci (supposons qu'il s'agit de $e_1\varphi$) appartient à une pelote, et l'autre non (les pelotes sont réversibles). L'état p commun à ces deux transitions fait partie d'une pelote qui contient $e_1\varphi$. L'état p est l'image par φ d'une des extrémités q de e_2. L'état q appartient à une composante fortement connexe dont l'image est la pelote de p tout entière (proposition 3.14). Donc $e_1\varphi$ appartient à $\mathcal{B}_2\varphi$, ce qui contredit la réversibilité de $\mathcal{B}_2\varphi$ et la proposition 3.15. □

Ce théorème nous dit que l'automate universel d'un langage réversible contient un sous-automate quasi-réversible qui reconnaît le langage. En fait, la preuve est plus précise et indique qu'il existe un tel sous-automate dont les composantes fortement connexes sont exactement celles de l'automate universel. On peut donc définir un sous-automate quasi-réversible maximal de l'automate universel qui reconnaît le langage.

PROPOSITION 3.16 *Soit $\mathcal{U}_\mathcal{L}$ l'automate universel d'un langage réversible $\mathcal{L}$. Soit $\mathcal{R}_\mathcal{L}$ l'automate obtenu à partir de $\mathcal{U}_\mathcal{L}$ en appliquant l'algorithme suivant :*

1. *Calcul des composantes fortement connexes de $\mathcal{U}_\mathcal{L}$ et marquage des transitions qui appartiennent à une pelote.*
2. *Suppression des transitions qui contredisent la quasi-réversibilité et qui n'appartiennent à aucune pelote.*

Alors, $\mathcal{R}_\mathcal{L}$ est un automate quasi-réversible qui reconnaît le langage $\mathcal{L}$.

Démonstration. Comme les pelotes de $\mathcal{U}_\mathcal{L}$ sont réversibles, $\mathcal{R}_\mathcal{L}$ est quasi-réversible. De plus, d'après la preuve du théorème 3.4, l'image de tout quipu réversible minimal qui reconnaît $\mathcal{L}$ est un sous-automate de $\mathcal{R}_\mathcal{L}$ qui reconnaît donc $\mathcal{L}$. □

Cet automate quasi-réversible est en quelque sorte universel pour les automates quasi-réversibles :

PROPOSITION 3.17 *Soit $\mathcal{A}$ un automate quasi-réversible qui reconnaît un langage $\mathcal{L}$ avec* *un nombre minimal d'états. Alors $\mathcal{A}$ est un sous-automate de l'automate $\mathcal{R}_{\mathcal{L}}$ défini ci-dessus.*

Démonstration. Soit φ un morphisme de $\mathcal{A}$ dans l'automate universel $\mathcal{U}_{\mathcal{L}}$. Soit $\mathcal{B}$ la décomposition en quipu réversible (rendu minimal) de $\mathcal{A}$ et μ le morphisme de $\mathcal{B}$ sur $\mathcal{A}$ qui en découle (morphisme surjectif à cause de la minimalité de $\mathcal{A}$). Alors $\mu.\varphi$ est un morphisme de $\mathcal{B}$ dans $\mathcal{U}_{\mathcal{L}}$ et, plus précisément, comme on l'a montré, un morphisme de $\mathcal{B}$ dans $\mathcal{R}_{\mathcal{L}}$. Donc μ est un morphisme de $\mathcal{A}$ dans $\mathcal{R}_{\mathcal{L}}$; ce morphisme est injectif à cause de la minimalité de $\mathcal{A}$. □

— ∘ —

EXEMPLE 16 Considérons le langage $\mathcal{L}_{r_3}$ défini par l'automate minimal de la figure 10. On peut vérifier que le langage $\mathcal{L}_{r_3}$ est un langage réversible. L'automate universel de $\mathcal{L}_{r_3}$

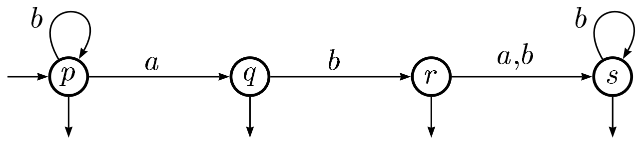


FIG. 10 – *L'automate minimal de $\mathcal{L}_{r_3}$.*

est représenté figure 11. En supprimant toutes les transitions hors des composantes forte-

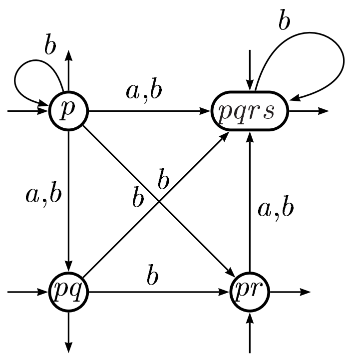


FIG. 11 – *L'automate universel de $\mathcal{L}_{r_3}$.*

ment connexes qui sont en contradiction avec la quasi réversibilité, on obtient l'automate de la figure 12.

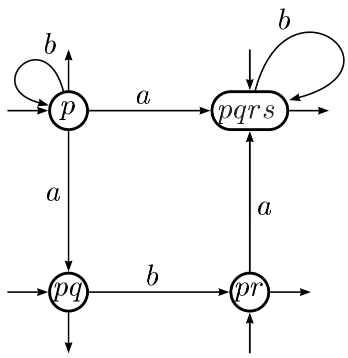


FIG. 12 – *Un automate quasi-réversible pour* $\mathcal{L}_{r_3}$.

Chapitre 4

Hauteur d'étoile

Tenter, sans force et sans armure,
D'atteindre l'inaccessible étoile.

J. Brel, *La quête*

Le problème de la hauteur d'étoile est l'un des plus anciens et des plus délicats posés dans la théorie des langages rationnels. En 1963, L. C. Eggan [22] établit l'égalité entre l'*enlacement* d'un automate[1] et la hauteur d'étoile d'une expression qu'on peut calculer à partir de ce graphe. Il posa alors deux questions. Premièrement, existe-t-il des langages intrinsèquement compliqués vis-à-vis de cette grandeur, c'est-à-dire tels que la hauteur d'étoile de toute expression qui représente ce langage est grande ? Deuxièmement, peut-on calculer la hauteur d'étoile d'un langage, c'est-à-dire la hauteur d'étoile minimale des expressions rationnelles qui représentent ce langage ?

La réponse à la première question vint rapidement. F. Dejean et M.-P. Schützenberger [21] répondirent positivement en 1966. La seconde question s'avéra plus ardue. Après des réponses partielles de R. McNaughton [47], R. Cohen [17] et K. Hashiguchi et N. Honda [31], K. Hashiguchi est parvenu en 1988, en utilisant notamment son résultat sur le *limitedness problem* [32], à montrer qu'on peut, du moins de manière théorique, décider de la hauteur d'étoile d'un langage rationnel [33].

Le résultat d'Hashiguchi n'apporte cependant pas pleinement satisfaction. D'une part, la complexité de l'algorithme le rend inapplicable même pour des données petites (*cf.* annexe B pour plus de détails sur l'algorithme d'Hashiguchi et une estimation du nombre d'expressions à traiter). D'autre part, l'algorithme ne tient pas compte finalement des propriétés du langage, puisqu'il consiste en l'énumération de toutes les expressions rationnelles de longueur bornée. Enfin l'algorithme lui-même (et pas seulement sa preuve) est difficile à exprimer.

On se propose dans ce chapitre de montrer le résultat suivant :

Théorème *L'automate universel d'un langage réversible contient un automate quasi-réversible équivalent d'enlacement minimum.*

1. « transition graph », à l'époque.

Dans la première partie, nous rappelons les définitions de hauteur d'étoile d'un langage rationnel et d'enlacement d'un automate, ainsi que le théorème d'Eggan qui permet de lier l'un à l'autre. D'autre part, nous présentons aussi une famille de morphismes de graphes (ou d'automates), dits *morphismes conformes*, introduits par R. McNaughton [47], qui ont la propriété de préserver l'enlacement.

Afin d'introduire les outils que nous utilisons, nous donnons ensuite une nouvelle présentation des résultats de R. McNaughton [47] sur la hauteur d'étoile des langages à groupe. On montre qu'il existe des pelotes de l'automate universel d'un tel langage dont l'union forme un automate à groupe équivalent d'enlacement minimum. La technique introduite ici s'apparente à celle du chapitre précédent; on considère un automate (inconnu) d'enlacement minimum et on montre que son image dans l'automate universel permet d'y trouver un sous-automate équivalent d'enlacement minimum lui aussi. Ce résultat, ainsi que la nouvelle preuve du théorème d'Eggan qui figurent dans ce chapitre ont été présentés à ICLWC [42].

Nous montrons ensuite le résultat principal de ce chapitre. Pour cela, nous utilisons la notion de corde introduite au chapitre précédent, ce qui nous permet de décomposer le langage et de pouvoir appliquer une méthode similaire à celle pratiquée pour les automates à groupe. Ce résultat, prouvé en collaboration avec J. Sakarovitch, est ici présenté de manière sensiblement différente à la preuve qui a été soumise par ailleurs ([43]), puisqu'elle tire partie des résultats du chapitre précédent. L'avantage que nous avons sur nos prédécesseurs est la connaissance du cadre dans lequel l'automate d'enlacement minimum doit se trouver.

Enfin, nous présentons la conjecture dans le cas général et expliquons pourquoi nos techniques se révèlent insuffisantes pour la résoudre.

— ◦ —

1 Hauteur d'étoile et degré d'enlacement

1.1 Hauteur d'étoile d'un langage rationnel

Expressions rationnelles. Comme la hauteur d'étoile est définie à partir des expressions rationnelles, il convient de les définir correctement et de ne pas confondre l'expression et l'ensemble qu'elle représente.

Définition 4.1 *Soit A un alphabet, et $\mathsf{0}$, $\mathsf{1}$, $+$, $\cdot$ et $\star$ des symboles qui n'appartiennent pas à A. L'ensemble* $\mathsf{RExp}A$ *des* ***expressions rationnelles*** *sur un alphabet A est défini inductivement comme suit :*

i) $\mathsf{0}$, $\mathsf{1}$ *et tout élément de A est une expression rationnelle (atomique).*

ii) *Si* E *et* F *sont des expressions rationnelles,* $\mathsf{E}+\mathsf{F}$ *et* $\mathsf{E}\cdot\mathsf{F}$ *aussi.*

iii) *Si* E *est une expression rationnelle,* $\mathsf{E}^\star$ *aussi.*

On ne fait pas apparaître dans cette définition les parenthèses qui encadrent implicitement chaque expression. On suppose qu'à partir d'une expression, on peut retrouver la façon dont elle a été construite.

Les expressions rationnelles peuvent être interprétées; elles représentent en effet des ensembles ou langages (qui sont, par définition, rationnels !).

DÉFINITION 4.2 *Soit* E *une expression rationnelle, le langage dénoté par* E*, et noté* $|\mathsf{E}|$ *est défini inductivement par :*

i) $|\mathsf{E}| = \emptyset$ *si* $\mathsf{E} = 0$*;* $|\mathsf{E}| = \{1_{A^*}\}$ *si* $\mathsf{E} = 1$*;* $|\mathsf{E}| = \{a\}$ *si* $\mathsf{E} = a$*;*

ii) $|\mathsf{E}| = |\mathsf{F}| \cup |\mathsf{G}|$ *si* $\mathsf{E} = \mathsf{F}+\mathsf{G}$*;* $|\mathsf{E}| = |\mathsf{F}|.|\mathsf{G}|$ *si* $\mathsf{E} = \mathsf{F} \cdot \mathsf{G}$*;*

iii) $|\mathsf{E}| = |\mathsf{F}|^*$*, si* $\mathsf{E} = \mathsf{F}^\star$.

L'opérateur $\star$, on le voit, joue un rôle remarquable. Sans lui, en effet, les langages rationnels ne seraient que des langages finis. On peut donc envisager de mesurer la complexité d'un langage rationnel au nombre de fois où l'on fait appel à $\star$ pour décrire le langage; c'est le sens de la définition suivante :

DÉFINITION 4.3 *La* ***hauteur d'étoile d'une expression rationnelle*** E *est définie inductivement par :*

i) $\mathsf{h}[\mathsf{E}] = 0$ *si* $\mathsf{E} = 0$, 1 *ou une lettre de* A*;*

ii) $\mathsf{h}[\mathsf{E}] = \max\{\mathsf{h}[\mathsf{F}],\mathsf{h}[\mathsf{G}]\}$*, si* $\mathsf{E} = \mathsf{F}+\mathsf{G}$ *ou* $\mathsf{E} = \mathsf{F} \cdot \mathsf{G}$*;*

iii) $\mathsf{h}[\mathsf{E}] = 1 + \mathsf{h}[\mathsf{F}]$*, si* $\mathsf{E} = \mathsf{F}^\star$.

— ∘ —

Définition de la hauteur d'étoile. La hauteur d'étoile d'une expression rationnelle est donc, on le voit, le nombre d'étoiles emboîtées dans l'expression. Il reste encore un pas à franchir pour que cette grandeur permette de mesurer la complexité d'un langage rationnel. En effet, deux expressions de hauteurs d'étoiles différentes peuvent représenter le même langage. Il est par exemple connu que l'expression $a^\star \cdot (b \cdot a^\star)^\star$ et l'expression $(a+b)^\star$ représentent le même langage. La complexité intrinsèque d'un langage correspond à la complexité de l'expression la plus simple qui permet d'exprimer ce langage :

DÉFINITION 4.4 *La* ***hauteur d'étoile d'un langage rationnel*** $\mathcal{L}$ *est :*

$$\mathsf{h}(\mathcal{L}) = \min\{\mathsf{h}[\mathsf{E}] \mid \mathsf{E} \in \mathsf{RExp}A,\, |\mathsf{E}| = \mathcal{L}\}$$

— ∘ —

1.2 Enlacement d'un graphe orienté

La pertinence de la hauteur d'étoile comme mesure de la complexité d'un langage est renforcée par le fait que cette grandeur ne mesure pas seulement la complexité des expressions qui mesurent le langage mais aussi celle des automates qui le reconnaissent, ou plus exactement, celle de leur graphe sous-jacent. Ce résultat, que nous développerons plus loin, est dû à Eggan [22] qui posa le premier les questions relatives à la hauteur d'étoile.

Définition 4.5 *Soit $\mathcal{G} = \langle Q, E \rangle$ un graphe orienté. L'**enlacement** de $\mathcal{G}$, $\mathsf{enl}(\mathcal{G})$, est défini récursivement par :*

- $\mathsf{enl}(\mathcal{G}) = 0$ *si $\mathcal{G}$ est acyclique.*
- *Si $\mathcal{G}$ n'est pas une pelote,* $\mathsf{enl}(\mathcal{G}) = \mathsf{max}\{\mathsf{enl}(\mathcal{P}) \mid \mathcal{P}$ *pelote de* $\mathcal{G}\}$.
- *Si $\mathcal{G}$ est une pelote,* $\mathsf{enl}(\mathcal{G}) = 1 + \mathsf{min}\{\mathsf{enl}(\mathcal{G} \setminus \{p\}) \mid p$ *sommet de* $\mathcal{G}\}$.

On voit que la définition de l'enlacement d'un graphe fait intervenir une alternance de minima et de maxima. On peut scinder la définition de l'enlacement en deux temps. D'une part, on effectue le calcul en suivant un ordre choisi par avance (sans avoir recours au minimum) et ensuite, on retient la valeur minimale par rapport à tous les ordres possibles.

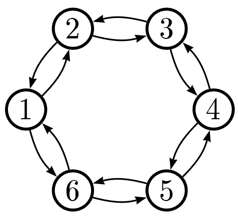
Étant donné un ordre total ω sur un ensemble, tout sous-ensemble est naturellement ordonné par ω. Dans ce qui suit, $\overline{\omega}$ désigne le maximum selon ω du sous-ensemble dans lequel on se trouve.

Définition 4.6 *Soit $\mathcal{G} = \langle Q, E \rangle$ un graphe orienté et ω un ordre total sur Q. L'enlacement de $\mathcal{G}$ selon ω, $\mathsf{enl}_\omega(\mathcal{G})$, est défini récursivement par :*

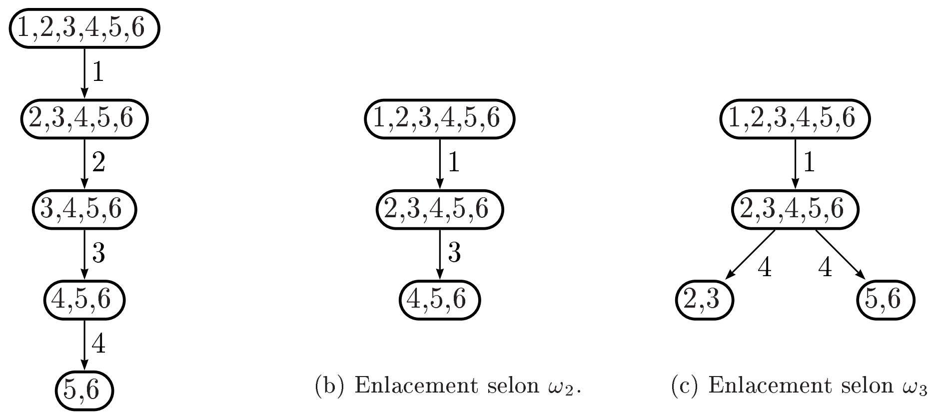
- $\mathsf{enl}_\omega(\mathcal{G}) = 0$ *si $\mathcal{G}$ est acyclique.*
- *Si $\mathcal{G}$ n'est pas une pelote,* $\mathsf{enl}_\omega(\mathcal{G}) = \mathsf{max}\{\mathsf{enl}_\omega(\mathcal{P}) \mid \mathcal{P}$ *pelote de* $\mathcal{G}\}$.
- *Si $\mathcal{G}$ est une pelote,* $\mathsf{enl}_\omega(\mathcal{G}) = 1 + \mathsf{enl}_\omega(\mathcal{G} \setminus \{\overline{\omega}\})$.

On voit que le calcul de l'enlacement d'un graphe consiste à le décomposer, d'une part en enlevant des sommets et d'autre part en séparant les pelotes. Le processus de calcul de l'enlacement peut être représenté par un arbre dont la hauteur est l'enlacement. Si le graphe contient plusieurs pelotes, on a une forêt d'arbres.

Exemple 17 On veut calculer l'enlacement du graphe suivant :



On peut fixer différents ordres sur les sommets. Soit $\omega_1 = (1,2,3,4,5,6)$, $\omega_2 = (1,3,5,2,4,6)$ et $\omega_3 = (1,4,2,5,3,6)$. On suppose que l'élément le plus grand est ici le premier de la liste. Le calcul de l'enlacement s'effectue comme indiqué sur la figure 1. Chaque nœud représente une pelote de l'automate. On passe d'un nœud à ses fils en supprimant le plus grand des sommets de la pelote. On voit que l'enlacement dépend fortement de l'ordre.



(a) Enlacement selon ω_1.

(b) Enlacement selon ω_2.

(c) Enlacement selon ω_3.

FIG. 1 – *Calcul d'enlacements d'un graphe.*

La proposition suivante assure que l'on peut effectivement calculer l'enlacement comme le minimum des enlacements selon tous les ordres possibles.

PROPOSITION 4.1 *Soit $\mathcal{G} = \langle Q,E \rangle$ un graphe orienté. L'enlacement de $\mathcal{G}$ est égal au minimum des enlacements de $\mathcal{G}$ selon chaque ordre total sur Q.*

Démonstration. La preuve est par récurrence sur le nombre de sommets. Si un graphe ne compte qu'un sommet, l'enlacement coïncide avec l'enlacement selon le seul ordre possible. Soit $\mathcal{G} = \langle Q,E \rangle$ un graphe. Si $\mathcal{G}$ est acyclique, alors, pour tout ordre ω sur Q, $\mathsf{enl}_\omega(\mathcal{G}) = \mathsf{enl}(\mathcal{G}) = 0$.

Si $\mathcal{G}$ n'est pas une pelote, par hypothèse de récurrence, il existe un ordre ω sur chaque pelote $\mathcal{P}$ de $\mathcal{G}$ tel que $\mathsf{enl}_\omega(\mathcal{P}) = \mathsf{enl}(\mathcal{P})$. En concaténant ces ordres, on obtient un ordre total sur Q (au besoin, on fixe un ordre sur les sommets qui n'apparetiennent à aucune pelote et on suppose qu'ils sont inférieurs aux autres). L'enlacement de $\mathcal{G}$ selon cet ordre est bien égal à $\mathsf{enl}(\mathcal{G})$

Si $\mathcal{G}$ est une pelote, par définition, il existe p dans Q tel que $\mathsf{enl}(\mathcal{G}) = 1 + \mathsf{enl}(\mathcal{G} \setminus \{p\})$. Par hypothèse de récurrence, il existe un ordre ω sur $Q \setminus \{p\}$ tel que $\mathsf{enl}(\mathcal{G} \setminus \{p\}) = \mathsf{enl}_\omega(\mathcal{G} \setminus \{p\})$. Si on étend cet ordre sur Q en supposant que p est maximum, on obtient que $\mathsf{enl}_\omega(\mathcal{G}) = 1 + \mathsf{enl}(\mathcal{G})$.

Réciproquement, supposons par l'absurde qu'il existe un graphe $\mathcal{G}$, avec un nombre minimal de sommets et un ordre ω sur les sommets de $\mathcal{G}$ tel que $\mathsf{enl}_\omega(\mathcal{G}) < \mathsf{enl}(\mathcal{G})$. Si $\mathcal{G}$ n'était pas une pelote, l'inégalité serait vérifiée sur une des pelote de $\mathcal{G}$, ce qui contredirait la minimalité de $\mathcal{G}$. Donc $\mathcal{G}$ est une pelote et on a :

$$\begin{aligned}\mathsf{enl}(\mathcal{G} \setminus \{\varpi\}) &\geqslant \mathsf{min}\{\mathsf{enl}(\mathcal{G} \setminus \{p\}) \mid p \in Q\} = \mathsf{enl}(\mathcal{G}) - 1 \\ &> \mathsf{enl}_\omega(\mathcal{G}) - 1 = \mathsf{enl}_\omega(\mathcal{G} \setminus \{\varpi\}),\end{aligned}$$

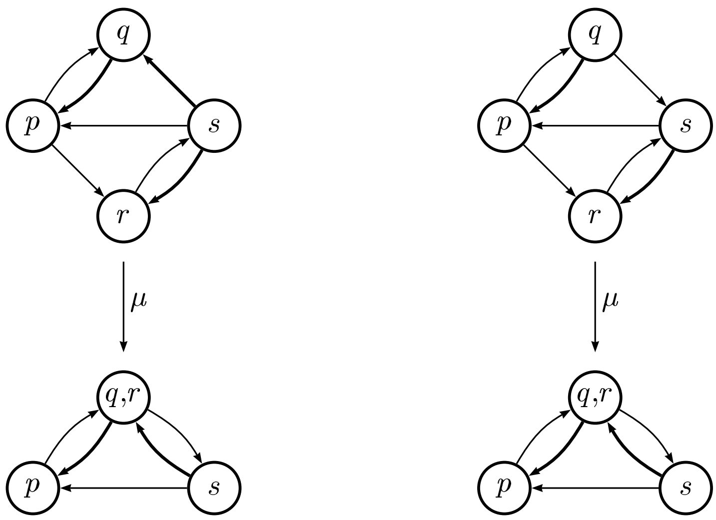
ce qui contredit la minimalité de $\mathcal{G}$. $\square$

Contrairement au calcul de la hauteur d'étoile d'une expression, trouver un algorithme simple du calcul de l'enlacement d'un graphe n'est pas évident. On peut toutefois trouver des moyens plus subtils que la simple énumération des permutations de l'ensemble des sommets. En représentant le calcul de la hauteur d'étoile suivant un ordre comme un arbre, on se rend compte que ce calcul peut s'effectuer non pas de la racine vers les feuilles (par suppression successive des sommets), mais des feuilles vers la racine (par ajouts successifs de sommets). Un nœud de l'arbre est créé chaque fois que la prise en compte d'un sommet fait apparaître une nouvelle pelote. On peut ainsi, en connaissant l'enlacement de chaque sous-graphe de taille k, calculer l'enlacement de chaque sous-graphe de taille $k+1$. Cette méthode demeure quand même exponentielle.

Nous verrons ensuite que l'on peut ramener le calcul de la hauteur d'étoile à celui de l'enlacement d'un graphe. Les morphismes de graphes (ou d'automates) qui respectent l'enlacement seront donc des outils précieux. La famille de morphismes présentée ici a été introduite par R. McNaughton [47] sous le nom anglais de *pathwise morphisms*.

DÉFINITION 4.7 *Soit $\mathcal{G}$ et $\mathcal{H}$ deux graphes orientés. Un morphisme de graphes μ de $\mathcal{G}$ dans $\mathcal{H}$ est un* ***morphisme conforme*** *s'il est surjectif et que, pour tout chemin $\mathcal{D}$ de $\mathcal{H}$, il existe un chemin $\mathcal{C}$ de $\mathcal{G}$ dont $\mathcal{D}$ est l'image.*

EXEMPLE 18 Le morphisme de la figure 2a) est un morphisme conforme. (Il s'agit en fait d'un *co-revêtement*[2].) Le morphisme de la figure 2b) n'est pas conforme, en effet, l'image inverse du chemin passant par les états s, $\{q,r\}$ puis p du graphe image ne contient pas de chemin.



(a) Morphisme conforme. (b) Morphisme non conforme.

FIG. 2 – *Deux morphismes de graphes.*

2. Formellement, si (p,q) est un arc du graphe image, pour tout s dans l'image inverse de q, il existe r dans l'image inverse de p tels que (r,s) est un arc du graphe de départ.

La proposition suivante accompagne évidemment la définition faite par R. McNaughton. Nous reproduisons ici la même démonstration, à une exception près que nous signalerons.

PROPOSITION 4.2 *Soit $\mathcal{G}$ et $\mathcal{H}$ deux graphes orientés. S'il existe un morphisme conforme μ de $\mathcal{G}$ sur $\mathcal{H}$, l'enlacement de $\mathcal{G}$ est supérieur à celui de $\mathcal{H}$. On dira que μ* ***préserve***[3] *l'enlacement.*

Cette proposition est issue du lemme suivant :

LEMME 4.3 *Soit $\mathcal{G}$ et $\mathcal{H}$ deux graphes orientés. S'il existe un morphisme conforme μ de $\mathcal{G}$ sur $\mathcal{H}$ et si $\mathcal{Q}$ est une pelote de $\mathcal{H}$, il existe une pelote $\mathcal{P}$ de $\mathcal{G}$ telle que $\mu_{|\mathcal{P}}$ est un morphisme conforme sur $\mathcal{Q}$.*

Démonstration. Faisons de $\mathcal{Q}$ un automate, en étiquetant chacun de ses arcs par une lettre distincte et en supposant chaque état initial et final. De même, $\mathcal{R} = \mathcal{Q}\mu^{-1}$ est un automate dans lequel tout état est initial et final, et les images inverses d'un arc de $\mathcal{Q}$ portent la même étiquette que celui-ci.

Le langage reconnu par $\mathcal{Q}$ représente l'ensemble des chemins dans cette pelote. μ est conforme, donc $\mathcal{R}$ reconnaît le même langage. Deux automates de taille n et m reconnaissent le même langage s'ils reconnaissent les mêmes mots de longueur inférieure à $2^{\mathsf{max}(m,n)}$ ([23]). Posons n le nombre d'états de $\mathcal{R}$ et considérons un chemin dans $\mathcal{Q}$ passant par tous les chemins de longueur 2^n et faisant une boucle. Soit w l'étiquette de ce chemin. w^n est reconnu par $\mathcal{Q}$ et donc par $\mathcal{R}$. Soit q_0 l'état de départ et q_n l'état atteint après avoir lu w^n. Par le lemme des tiroirs, deux de ces états sont les mêmes, donc il existe une pelote $\mathcal{P}$ reconnaissant le mot w, donc tout le langage, ce qui revient à dire que $\mu_{|\mathcal{P}}$ est un morphisme conforme sur $\mathcal{Q}$. □

La preuve de ce lemme diffère légèrement de celle qu'on trouve dans [47]. En effet, on peut ne pas utiliser la borne « 2^{m+n} » et considérer une suite de mots w_k qui étiquette un chemin contenant tous les chemins de longueur k. Pour chaque w_k, on obtient ainsi une pelote $\mathcal{P}_k$; lorsque k tend vers l'infini, une pelote $\mathcal{P}$ apparaît infiniment souvent dans la suite $(\mathcal{P}_k)_{k\in\mathbb{N}}$; elle vérifie le lemme.

Montrons maintenant la proposition en suivant mot à mot (ou presque) la preuve de R. McNaughton.

Démonstration de la proposition 4.2. Supposons qu'elle est fausse. Alors, parmi tous les graphes ne vérifiant pas la proposition, il existe $\mathcal{G}$ d'enlacement minimal c. Soit $d > c$ l'enlacement de $\mathcal{H}$. Si $c = 0$, la longueur des chemins de $\mathcal{G}$ est bornée et pas celle de $\mathcal{H}$. C'est impossible, donc $c > 0$.

Il existe une pelote $\mathcal{Q}$ de $\mathcal{H}$ de degré d, et d'après le lemme, il existe une pelote de $\mathcal{G}$ dont $\mathcal{Q}$ est l'image. Cette pelote $\mathcal{P}$ est au plus de degré c. Elle ne peut être de degré inférieur par minimalité de c. Il y a donc un état q de $\mathcal{P}$ tel que $\mathsf{enl}(\mathcal{P}\smallsetminus\{q\}) = c-1$. D'où $\mathsf{enl}(\mathcal{Q}\smallsetminus\{q\mu\}) \geqslant d-1$ et $\mathsf{enl}(\mathcal{P}\smallsetminus\{q\mu\mu^{-1}\}) \leqslant c-1$. On a donc construit un morphisme

3. plutôt que « conserve », puisque l'enlacement peut décroître.

conforme entre $\mathcal{P}\setminus\{q\mu\mu^{-1}\}$ et $\mathcal{Q}\setminus\{q\mu\}$ qui contredit la minimalité de c. Donc la proposition est vraie. □

— ∘ —

Maintenant que nous disposons d'un outil suffisament puissant pour manipuler l'enlacement de graphes, il faut établir le lien entre cette notion et la hauteur d'étoile d'un langage rationnel.

Définition 4.8 *L'**enlacement d'un automate** est égal à l'enlacement de son graphe sous-jacent. De même, un morphisme d'automate est conforme s'il est surjectif et que chaque chemin de l'image peut se relever en un chemin.*

La proposition 4.2 s'applique évidemment aux automates. Une illustration de l'utilisation de cette proposition est donnée par le résultat suivant :

Proposition 4.4 *Soit $\mathcal{A}$ un automate déterministe émondé reconnaissant le langage $\mathcal{L}$. Il existe un morphisme conforme de $\mathcal{A}$ sur l'automate minimal de $\mathcal{L}$. Par conséquent, l'automate minimal d'un langage est celui qui, parmi tous les automates déterministes reconnaissant ce langage, a un enlacement minimum.*

Démonstration. Soit $\mathcal{A}_{\mathcal{L}} = \langle Q,A,E,\{i\},T \rangle$ l'automate minimal de $\mathcal{L}$. L'application qui, à tout état p de $\mathcal{A}$, associe l'état $i \cdot \mathsf{Past}_{\mathcal{A}}(p)$ est un morphisme. Tout chemin $\mathcal{C}$ de $\mathcal{A}_{\mathcal{L}}$ peut se prolonger en un calcul étiqueté par un mot u. Ce mot u est accepté par l'automate $\mathcal{A}$; il existe donc un calcul de $\mathcal{A}$ étiqueté par u dont l'image dans $\mathcal{A}_{\mathcal{L}}$ ne peut être que $\mathcal{C}$ puisque $\mathcal{A}_{\mathcal{L}}$ est déterministe. Le morphisme est donc conforme. La seconde partie de la proposition découle de la proposition 4.2. □

— ∘ —

1.3 Enlacement et hauteur d'étoile

L'enlacement, on l'a vu, est relatif au graphe sous-jacent à un automate; la hauteur d'étoile, elle, correspond à une expression rationnelle. Le théorème d'Eggan [22] permet d'établir le lien entre ces deux grandeurs :

Théorème 4.1 *La hauteur d'étoile d'un langage rationnel est égale à l'enlacement minimal des automates qui reconnaissent le langage.*

Pour cela, on utilise des automates qui sont non plus étiquetés par des lettres, mais par des expressions rationnelles :

Définition 4.9 *Un quintuplet $\mathcal{A} = \langle Q,\mathsf{RExp}A,E,I,T \rangle$ est un **automate généralisé** si Q est un ensemble fini d'états, $\mathsf{RExp}A$ l'ensemble des expressions rationnelles sur l'alphabet A, E un sous-ensemble de $Q \times \mathsf{RExp}A \times Q$ et I et T des sous-ensembles de Q.*

Un automate $\mathcal{A}$ est donc un automate généralisé particulier, puisque les seules expressions qui étiquettent ses transitions sont des lettres.

Nous alons étendre la notion d'enlacement aux automates généralisés. On définit l'indice « Eggan »[4] i_{E} d'un automate généralisé. Cette méthode pour prouver le théorème d'Eggan [22] a été explicitée dans une communication faite en collaboration avec J. Sakarovitch [42].

Définition 4.10 *Soit* $\mathcal{A} = \langle Q, A^*, E, I, T \rangle$ *un automate généralisé et* ω *un ordre total sur* Q. *L'**indice*** i_{E} ***d'un automate généralisé*** $\mathcal{A}$ *selon* ω, *est défini récursivement par :*

– *Si* $\mathcal{A}$ *est acyclique :*

$$\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A}) = \mathsf{max}\{\mathsf{h}[|e|] \mid e \in E\}.$$

– *Si* $\mathcal{A}$ *n'est pas une pelote :*

$$\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A}) = \mathsf{max}(\{\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{P}) \mid \mathcal{P} \text{ pelote de } \mathcal{A}\} \\ \cup\{\mathsf{h}[|e|] \mid e \text{ n'appartient pas à une pelote}\}).$$

– *Si* $\mathcal{A}$ *est une pelote,*

$$\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A}) = 1 + \mathsf{max}(\{\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A} \setminus \{\overline{\omega}\})\} \\ \cup\{\mathsf{h}[|e|] \mid e \text{ adjacente à } \overline{\omega}\}).$$

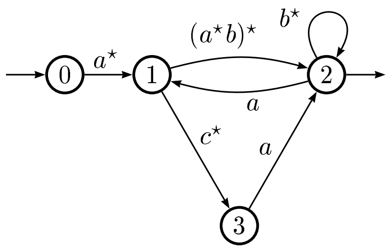
Les deux types extrêmes d'automates généralisés représentant un langage sont d'une part l'automate classique étiqueté par des lettres et d'autre part un automate réduit à un état initial et un état final dont l'unique transition (entre ces deux états) est étiquetée par une expression représentant le langage.

On voit que l'indice i_{E} d'un automate classique est exactement égal à l'enlacement selon le même ordre et que l'indice i_{E} d'un automate du second type est exactement la hauteur d'étoile de l'expression qui l'étiquette. On va montrer qu'on peut passer du premier type au second et que l'indice i_{E} reste invariant au cours de cette transformation.

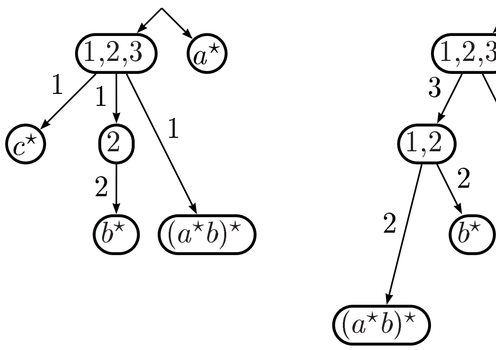
De même que le calcul de l'enlacement d'un graphe, le calcul de l'indice i_{E} peut se représenter par un arbre. La racine de cet arbre est l'automate lui-même, Cependant, dans ce cas, on a deux sortes de nœuds. D'une part, comme précédemment les nœuds qui correspondent à des pelotes, après suppressions successives d'états. D'autres part, des feuilles qui correspondent aux expressions rationnelles qui, soit étiquettent des transitions adjacentes à l'état qu'on supprime, soit étiquettent des transitions qui se retrouvent en dehors des pelotes. Les transitions dans l'arbre qui relient ces feuilles à leur père ont une longueur qui n'est pas obligatoirement égale à 1 mais à la hauteur d'étoile de l'expression qu'elles représentent (c'est la raison pour laquelle, on ne représente pas les expressions de hauteur d'étoile 0 et qu'on retrouve dans ce cas les arbres présentés pour l'enlacement).

Exemple 19 Considérons l'automate généralisé suivant :

4. Du nom du théorème qu'il permet de prouver.



Calculons l'indice i_{E} pour les ordres $\omega_1 = (0,1,2,3)$ et $\omega_2 = (3,2,1,0)$. Les arbres correspondants sont représentés figure 3. Dans le premier cas, l'indice i_{E} est égal à 3, dans le second, il est égal à 4.



(a) Indice i_{E} selon ω_1. (b) Indice i_{E} selon ω_2.

FIG. 3 – *Calcul de l'indice* i_{E}.

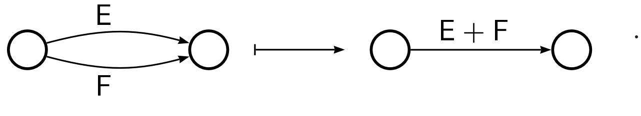
— ∘ —

1.4 Du calcul d'une expression au théorème d'Eggan

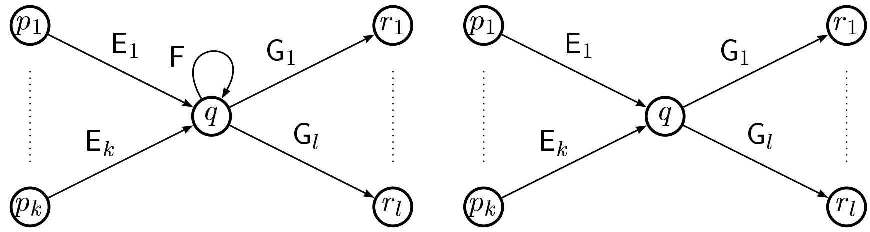
Rappelons une méthode classique, dite méthode d'élimination, pour obtenir une expression rationnelle à partir d'un automate.

Soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate et ω un ordre total sur Q, ensemble des états de $\mathcal{A}$. On normalise $\mathcal{A}$, c'est-à-dire qu'on rajoute deux états i (initial) et t (terminal), des transitions spontanées (c'est-à-dire étiquetées par 1) entre i et chaque état de I d'une part, et entre chaque état de T et t d'autre part.
On obtient l'automate $\mathcal{A} = \langle\, Q \cup \{i,t\},A,E',i,t \,\rangle$.

On va ensuite éliminer les états de $\mathcal{A}_0$ dans l'ordre ω, en modifiant les transitions de sorte qu'à chaque étape, le langage représenté par l'automate reste le même. Pour cela, on considère que les étiquettes de l'automate sont des expressions rationnelles (au départ, réduites à des lettres). On profite donc de cette généralisation pour fixer une règle qui permet de n'avoir qu'une seule transition entre un couple donné d'états :



Lorsqu'on supprime un état q dans une des configurations suivantes,



on crée respectivement les transitions suivantes :

$$p_i \xrightarrow{\mathsf{E}_i \cdot \mathsf{F}^\star \cdot \mathsf{G}_j} r_j \qquad p_i \xrightarrow{\mathsf{E}_i \cdot \mathsf{G}_j} r_j$$

Tout calcul passant par p est étiqueté par un mot qui se factorise en $u.v.w$, où u étiquette un chemin de i à p_i, w un chemin de r_j à t et v un chemin de p_i à r_j dont les seuls états intérieurs sont p. Donc w appartient bien au langage dénoté par $\mathsf{E}_i \cdot \mathsf{F}^\star \cdot \mathsf{G}_j$ et, après suppression de p, l'automate reconnaît bien au moins tous les mots qu'il reconnaissait avant. Réciproquement, tout mot du langage dénoté par $\mathsf{E}_i \cdot \mathsf{F}^\star \cdot \mathsf{G}_j$ (ou par $\mathsf{E}_i \cdot \mathsf{G}_j$, ce qui est encore plus simple) peut se décomposer en $u.v^k.w$, avec $u \in |\mathsf{E}_i|$, $v \in |\mathsf{F}|$ et $w \in |\mathsf{G}_j|$. Avant suppression de p, il existe donc un chemin de p_i à r_j, dont le seul état intérieur est p et dont l'étiquette contient $u.v^k.w$.

Lorsqu'on a supprimé tous les états de Q, on est donc dans la configuration suivante :

$$\longrightarrow i \xrightarrow{\quad\mathsf{E}\quad} t \longrightarrow$$

Comme à chaque étape le langage dénoté est toujours le même, l'expression E représente le langage.

Proposition 4.5 *L'expression obtenue à partir d'un automate émondé $\mathcal{A}$ en suivant la méthode d'élimination des états selon l'ordre ω représente le langage reconnu par l'automate $\mathcal{A}$. La hauteur d'étoile de cette expression est égale à l'enlacement de $\mathcal{A}$ selon ω.*[5]

On suit la démonstration faite dans [42].

Démonstration. La preuve est par récurrence sur le nombre d'états n de $\mathcal{A}$. (En fait l'automate a $n+2$ états, puisqu'il a un état initial et un état final qui lui sont ajoutés.) On montre que l'indice i_{E} d'un automate généralisé de taille n selon ω est égal à la hauteur d'étoile de l'expression obtenue en éliminant les états dans l'ordre ω.

Si $n = 0$, l'automate ne comporte au plus qu'une transition entre l'état initial et l'état terminal. C'est la configuration finale de l'algorithme d'élimination. La hauteur d'étoile de cette expression est justement l'index i_{E} de l'automate.

Supposons la propriété vraie pour tout entier strictement inférieur à n et $\mathcal{A}$ un automate à $n(+2)$ états.

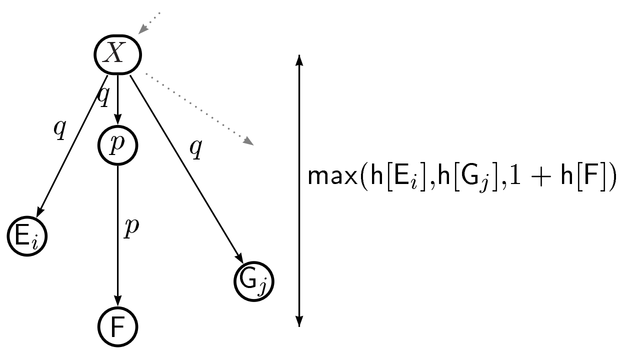
5. Il faut souligner que le calcul de l'enlacement de $\mathcal{A}$ selon ω consiste à traiter les états dans l'ordre décroissant selon ω, alors que la méthode d'élimination le fait dans l'ordre croissant.

Si l'automate $\mathcal{A}$ (abstraction faite des états i et t) n'est pas une pelote, l'expression rationnelle obtenue par l'algorithme d'élimination est une somme ou un produit d'expressions qui, soit étiquettent des transitions en dehors des pelotes, soit correspondent à une pelote. La hauteur d'étoile de l'expression est le maximum des hauteurs d'étoile de toutes ces composantes. Par hypothèses de récurrence, la hauteur d'étoile des expressions issues de pelotes est égale à l'indice i_{E} de ces pelotes; la hauteur d'étoile de l'automate est donc égale à son indice i_{E}.

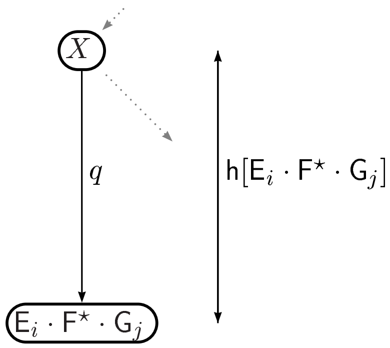
Si l'automate $\mathcal{A}$ (abstraction faite des états i et t) est une pelote, on pose p le plus petit état de $\mathcal{A}$ (selon ω). Soit $\mathcal{A}'$ l'automate obtenu après suppression de p dans l'algorithme d'élimination. Remarquons que, si $\mathcal{A}$ est une pelote, $\mathcal{A}'$ aussi.
L'algorithme de calcul de l'indice i_{E} va se dérouler de la même manière dans $\mathcal{A}$ et $\mathcal{A}'$ jusqu'à la dernière étape (c'est-à-dire l'examen de p), car les transitions qui relient leurs états sont les mêmes.

Les étiquettes des transitions de $\mathcal{A}'$ se divisent en deux parties :

- d'une part, les transitions identiques à celles de $\mathcal{A}$;
- d'autre part, des transitions issues de la suppression de p par l'algorithme d'élimination et qui sont, soit de la forme $\mathsf{E} \cdot \mathsf{G}$ si p n'a pas de boucle dans $\mathcal{A}$, soit de la forme $\mathsf{E} \cdot \mathsf{F}^{\star} \cdot \mathsf{G}$, sinon. Dans le premiers cas, les enlacements de $\mathcal{A}$ et $\mathcal{A}'$ sont évidemment égaux; dans le second cas, examinons l'arbre qui représente le calcul de l'indice i_{E}. Soit X l'ensemble d'états qui étiquette le père de p dans l'arbre correspondant à $\mathcal{A}$ et q le plus grand élément de X. La suppression de q dans le calcul conduit à une fragmentation de la pelote en différentes pelotes dont $\{p\}$. Localement, l'arbre à donc la forme suivante :



Dans $\mathcal{A}'$, comme p n'existe plus, la suppression de q entraîne que les transitions étiquetées par les expressions $\mathsf{E}_i \cdot \mathsf{F}^{\star} \cdot \mathsf{G}_j$ sont soit supprimées, soit se retrouvent en dehors de toute pelote, l'arbre a donc la forme suivante :



Les deux arbres correspondants aux deux calculs ont donc la même hauteur, et l'indice i_{E} de $\mathcal{A}'$ est égal à celui de $\mathcal{A}$.

Ceci permet de conclure la démonstration. □

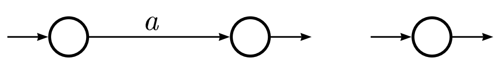
Démonstration du théorème 4.1. D'après la proposition précédente, la hauteur d'étoile d'un langage est inférieure ou égale à l'enlacement minimal des automates qui reconnaissent le langage. En effet, étant donné un automate d'enlacement minimal, on peut calculer une expression dont la hauteur d'étoile est égale à l'enlacement de l'automate.

On montre maintenant l'inégalité dans l'autre sens, c'est-à-dire qu'à partir d'une expression, on peut construire un automate qui décrive le même langage et dont l'enlacement est égal à la hauteur d'étoile de l'expression. Pour cela, on construit un automate en suivant les règles de formation d'une expression rationnelle. On peut noter qu'à part pour le langage vide représentable par un automate vide dont l'enlacement est bien égal à la hauteur d'étoile du langage, dans toute expression rationnelle, on peut supprimer la constante $\mathsf{0}$ en utilisant les règles suivantes :

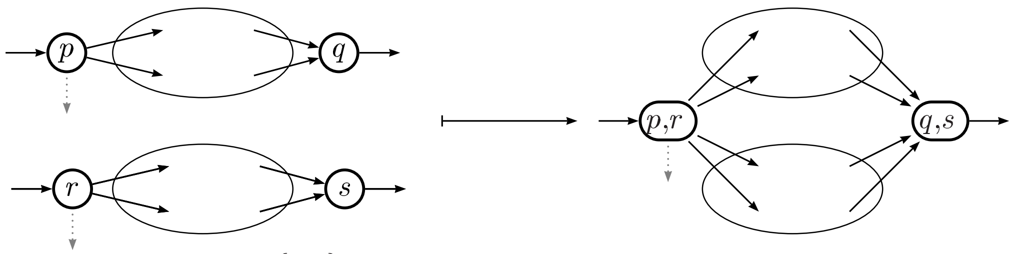
$$(\mathsf{0}\cdot\mathsf{E}) \mapsto \mathsf{0}, \quad (\mathsf{0}+\mathsf{E}) \mapsto \mathsf{E}, \quad \text{et } (\mathsf{0})^\star \mapsto \mathsf{1}.$$

Si l'application de ces règles ne conduit pas à une suppression de $\mathsf{0}$, c'est que le langage est vide. On considère donc les expressions rationnelles sans $\mathsf{0}$. On construit récursivement un automate qui décrit le même langage que l'expression en assurant que l'enlacement du premier est égal à la hauteur de la seconde. On construit des automates quasi-normalisés, dans le sens où il n'y a pas de transition spontanée et que, éventuellement, l'état initial peut être final (en plus de l'état final « unique »).

i) Si l'expression est a ou 1, le langage est reconnu respectivement par un des deux automates suivants d'enlacement 0.

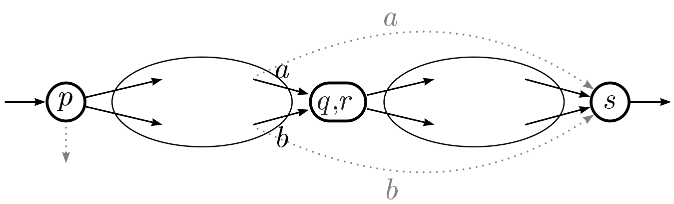


ii) Si E et F sont deux expressions pour lesquelles il existe deux automates $\mathcal{A}$ et $\mathcal{B}$ qui décrivent le même langage, et dont l'enlacement respectif est égal à la hauteur d'étoile respective de chacune des expressions, alors $\mathsf{E}+\mathsf{F}$ représente le langage reconnu par l'automte suivant, dont l'enlacement est le maximum des enlacements de $\mathcal{A}$ et $\mathcal{B}$, donc la hauteur d'étoile de $\mathsf{E}+\mathsf{F}$:



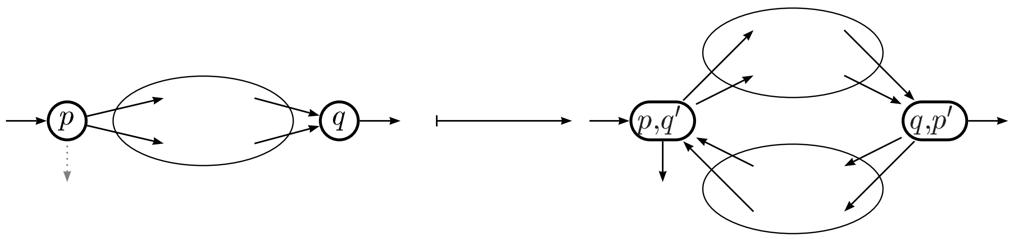
Si p ou r est initial, l'état $\{p,r\}$ est initial.

iii) Pour le produit, la construction est similaire :



Si p est final dans le premier automate, il l'est dans le résultat; si r est final, on crée dans le résultat, pour toute transition qui arrive en $\{q,r\}$, une transition étiquetée par la même lettre qui arrive en s.

iv) Pour l'étoile, à partir d'un automate, on crée deux copies et on réunit l'état initial de la première avec l'état final de la seconde et réciproquement :



Si on enlève l'état initial ou l'état final, on obtient un automate dont l'enlacement est égal à celui de départ, l'enlacement est donc inférieur à la hauteur d'étoile de E plus 1. Si on enlève un état interne, il appartient à l'une des deux copies; l'autre copie n'est pas modifiée, l'enlacement diminue donc au plus d'1. □

— ∘ —

2 Hauteur d'étoile des langages à groupe

De même que pour l'étude de l'automate universel, l'algorithme présenté pour les automates à groupe, qui est une adaptation de celui de R. McNaughton [47], se veut un avant-goût de celui qui est présenté pour les automates réversibles.

Soit $\mathcal{L}$ un langage à groupe et $\mathcal{A}$ son automate minimal. Cet automate est une pelote, son monoïde de transition est le groupe syntaxique G dont l'unique idempotent est 1_G.

Dans tout automate $\mathcal{B}$ qui accepte $\mathcal{L}$, on peut trouver un état r qui se comporte dans $\mathcal{B}$ sensiblement de la même façon que l'état initial dans $\mathcal{A}$.

LEMME 4.6 *Soit $\mathcal{L}$ un langage dont le monoïde syntaxique est un groupe G. Son automate minimal est $\mathcal{A} = \langle Q,A,E,i,T \rangle$ et soit $\mathcal{B}$ un automate équivalent. Soit g un élément du groupe syntaxique G correspondant à une partie du langage, H_g son image inverse dans A^* (H_g est inclus dans $\mathcal{L}$) et H_{1_G} l'image inverse de 1_G. Il existe un état r de $\mathcal{B}$ tel que :*

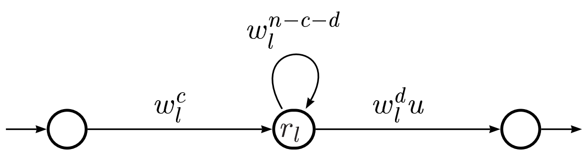
i) *$H_{1_G} \cap \mathsf{Past}_{\mathcal{B}}(r) \neq \emptyset$ et $H_g \cap \mathsf{Fut}_{\mathcal{B}}(r) \neq \emptyset$.*

ii) *Pour tout mot v de H_{1_G}* [6]*, il existe u dans H_{1_G} et w dans A^* tels que $u.v.w$ étiquette une boucle autour de r.*

Démonstration. Pour tout entier l positif, on définit B_l l'ensemble des mots de longueur inférieure à l qui étiquettent une boucle autour de i dans $\mathcal{A}$. Il existe un entier k (l'ordre du groupe syntaxique G) tel que pour tout mot u de B_l, le mot u^{k} appartient à H_{1_G}. On pose :

$$w_l = \prod_{u \in B_l} u^{\mathsf{k}}.$$

Soit n le nombre d'états de $\mathcal{B}$. Pour tout u appartenant à H_g, le mot $(w_l)^n.u$ appartient aussi à H_g, donc à $\mathcal{L}$. Il est accepté par $\mathcal{B}$. Ce calcul peut se décomposer de la façon suivante:



L'état r_l respecte i) et ii) pour v de longueur inférieure à l. Lorsqu'on fait tendre l vers l'infini, on obtient une suite d'états $(r_l)_{l\in\mathbb{N}}$ dans laquelle un état r de $\mathcal{B}$ apparaît infiniment souvent. Cet état vérifie les conditions i) et ii). □

THÉORÈME 4.2 *Si le monoïde syntaxique d'un langage $\mathcal{L}$ est un groupe, l'automate universel contient un sous-automate reconnaissant $\mathcal{L}$ et d'enlacement minimal. Plus précisément, il existe des pelotes de l'automate universel dont l'union forme un automate qui reconnaît $\mathcal{L}$ avec un enlacement minimum.*

Démonstration. Soit $\mathcal{B}$ un automate d'enlacement minimal reconnaissant $\mathcal{L}$. Soit g un élément du groupe syntaxique qui correspond à une partie H_g de $\mathcal{L}$. Il existe un état r qui vérifie les deux conditions du lemme. Soit φ un morphisme de $\mathcal{B}$ dans $\mathcal{U}_{\mathcal{L}}$ et $s = r\varphi$.

$H_{1_G} \cap \mathsf{Past}_{\mathcal{B}}(r) \neq \emptyset$, donc, comme $\mathsf{Past}_{\mathcal{B}}(r) \subseteq \mathsf{Past}_{\mathcal{U}_{\mathcal{L}}}(s)$, et que le passé de s est reconnu par G, $H_{1_G} \subseteq \mathsf{Past}_{\mathcal{U}_{\mathcal{L}}}(s)$, donc s est initial. De même, H_g est inclus dans le futur de s. Considérons la pelote $\mathcal{P}$ de $\mathcal{U}_{\mathcal{L}}$ qui contient s. Il existe un mot v de A^* tel que, pour tout mot u de H_g, le mot $u.v$ appartient à H_{1_G}. Donc il existe une boucle autour de s étiquetée par $u.v$. Soit t l'état de cette boucle atteint à partir de s en lisant u. Le mot u appartient

6. qui étiquette donc une boucle autour de i (état initial de $\mathcal{A}$)

à $\mathsf{Fut}_{\mathcal{U}_{\mathcal{L}}}(s)$, donc le mot vide appartient à $\mathsf{Fut}_{\mathcal{U}_{\mathcal{L}}}(t)$, qui est terminal. La pelote $\mathcal{P}$ accepte donc (au moins) tous les mots de H_g.

Montrons que $\mathcal{P}$ est d'enlacement inférieur à $\mathcal{B}$. Tout chemin de $\mathcal{P}$ est contenu dans une boucle autour de s qui, quitte à tourner plusieurs fois, est étiquetée par un mot v de H_{1_G}. Il existe donc u dans H_{1_G} et w dans A^* tels que $u.v.w$ étiquette une boucle autour de r dans $\mathcal{B}$:

$$r \xrightarrow{\quad u \quad} p \xrightarrow{\quad v \quad} q \xrightarrow{\quad w \quad} r$$

L'image de cette boucle par φ est une boucle autour de s étiquetée par $u.v.w$. Comme u est dans H_{1_G}, il étiquette une boucle autour de s, de même que v :

$$s \xrightarrow{\quad u \quad} s \xrightarrow{\quad v \quad} s \xrightarrow{\quad w \quad} s$$

Cette boucle est celle qui contient le chemin de départ, puisque $\mathcal{P}$ est déterministe. Tout chemin de $\mathcal{P}$ a donc une image inverse dans $\mathcal{B}$. L'enlacement de $\mathcal{P}$ est donc inférieur ou égal à celui de $\mathcal{B}$.

On peut faire ce raisonnement pour tout élément de G image d'une partie de $\mathcal{L}$. On obtient ainsi une union de pelotes d'enlacement inférieur à celui de $\mathcal{B}$. Elles forment donc un automate d'enlacement minimal qui reconnaît $\mathcal{L}$. □

— ∘ —

Corollaire 4.7 *Soit $\mathcal{A}$ l'automate minimal d'un langage à groupe. Si $\mathcal{A}$ a un seul état final, il est d'enlacement minimal.*

Démonstration. D'après le corollaire 3.7, dans ce cas, l'automate universel est isomorphe à l'automate minimal. La seule pelote de l'automate universel est donc $\mathcal{A}$ lui-même qui est donc d'enlacement minimal. □

Une conséquence triviale de ce corollaire est le résultat de F. Dejean et M.-P. Schützenberger [21] selon lequel, pour tout entier n, il existe des langages rationnels de hauteur d'étoile n. Il suffit de considérer les langages à groupe $\{u \in \{a,b\}^* \mid |u|_a = |u|_b \mod 2^n\}$, dont l'automate minimal a un seul état final et est d'enlacement n.

L'algorithme qui permet de trouver un sous-automate d'enlacement minimum ne nécessite pas une énumération de tous les sous-automates de l'automate universel. Il suffit de calculer l'enlacement des pelotes et de considérer les automates successifs formés par l'adjonction des pelotes en commençant par celles d'enlacement minimum, jusqu'à ce qu'on obtienne un automate qui reconnaisse le langage. De plus, si l'automate universel est construit à partir de l'automate minimal, il est facile de savoir quelle est la part du langage reconnue par chaque pelote.

Proposition 4.8 *Soit $\mathcal{A} = \langle\, Q,A,E,i,T \,\rangle$ l'automate minimal d'un langage à groupe $\mathcal{L}$. Soit $\mathcal{P}$ une pelote de l'automate universel de $\mathcal{L}$ construit à partir de $\mathcal{A}$. Les états de $\mathcal{P}$*

sont indicés par des sous-ensembles de Q. Le passé d'un état X dans $\mathcal{P}$ est égal à l'union des passés des états p dans $\mathcal{A}$, pour p dans X :

$$\mathsf{Past}_{\mathcal{P}}(X) = \bigcup_{p \in X} \mathsf{Past}_{\mathcal{A}}(p).$$

Démonstration.

$$u \in \mathsf{Past}_{\mathcal{P}}(X) \Leftrightarrow Y = u \cdot X \text{ initial} \Leftrightarrow i \cdot u \in X \Leftrightarrow u \in \bigcup_{p \in X} \mathsf{Past}_{\mathcal{A}}(p)$$

□

Le langage accepté par une pelote étant l'union des passés des états terminaux de la pelote, il est très simple de savoir si une union de pelotes réalise ou non tout le langage.

Exemple 15.2 L'enlacement de l'automate minimal de $\mathcal{L}_{g_2}$ (figure 3 page 83) est égal à 3. Dans cet exemple, l'automate minimal est isomorphe au graphe de Cayley du groupe. Chaque état correspond donc à un élément particulier du groupe et son passé à l'image inverse de cet élément par le morphisme canonique de A^* dans le groupe. On notera par exemple H_t l'image inverse de l'élément du groupe correspondant à l'état t. L'examen de l'automate universel donne le tableau suivant, dans lequel chaque pelote est désignée par un de ses états :

Pelote	Enlacement	langage reconnu
$\{p\}$	3	$H_q \cup H_s \cup H_t \cup H_u$
$\{p,s\}$	2	$H_q \cup H_t$
$\{p,t\}$	2	$H_q \cup H_u$
$\{p,u\}$	2	$H_q \cup H_s$
$\{p,q,r\}$	2	$H_s \cup H_t \cup H_u$
$\{p,q,r,s\}$	3	$H_q \cup H_s \cup H_t \cup H_u$

On voit qu'il suffit de prendre, par exemple, les pelotes des états $\{p,q,r\}$ et $\{p,s\}$ pour former un automate d'enlacement 2 qui reconnaît le langage (figure 4). D'après le théorème précédent, cet automate est d'enlacement minimal pour le langage. La hauteur d'étoile du langage $\mathcal{L}_{g_2}$ est donc 2.

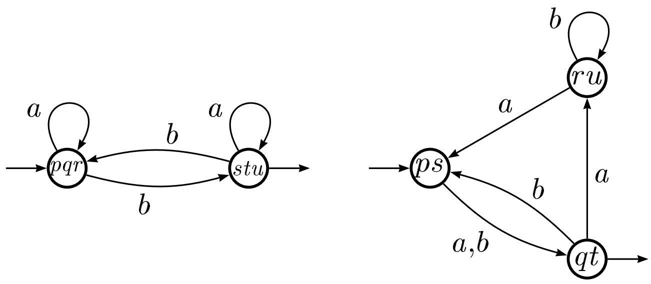


Fig. 4 – *Un automate d'enlacement minimum pour $\mathcal{L}_{g_2}$*

— ◦ —

3 Hauteur d'étoile des langages réversibles

Les langages réversibles étant la généralisation naturelle des langages à groupe, il paraît naturel d'essayer d'y étendre la méthode de R. McNaughton.

Les langages considérés par R. Cohen [17] et K. Hashiguchi et N. Honda [31] font partie d'une classe plus restreinte que celle des langages réversibles, celle des langages reset-free. Un langage est **reset-free** si son automate minimal est réversible. On voit que le problème de l'appartenance à cette classe est trivialement réglé de même que la construction d'un automate réversible pour un de ces langages.

R. Cohen montre que si l'automate minimal d'un tel langage n'a qu'un état minimal, l'automate minimal est d'enlacement minimal pour le langage. C'est la généralisation du corollaire 4.7 au langages reset-free. K. Hashiguchi et N. Honda étendent quant à eux la méthode de R. McNaughton. Il faut souligner ici une différence profonde entre les algorithmes de R. McNaughton, N. Hashiguchi et N. Honda et ceux présentés ici. Dans le premier cas, à partir d'un automate connu ayant les bonnes propriétés (de groupe ou de réversibilité), on construit des automates parmi lesquels on assure qu'on en trouvera d'enlacement minimum dont l'union reconnaît le langage. Dans le second cas, on construit l'automate universel à partir de l'automate minimal, dans lequel on cherche des sous-automates ayant les propriétés voulues. On peut montrer que cet automate universel respecte certaines propriétés à partir d'un automate qu'on ne sait *a priori* pas construire.

Avant de passer à la démonstration du résultat, on présente deux lemmes.

DÉFINITION 4.11 *On dit qu'un mot u est un* ***mot idempotent pour un automate*** $\mathcal{A}$ *si son image dans le monoïde de transition de* $\mathcal{A}$ *est un idempotent.*

LEMME 4.9 *Soit $\mathcal{A}$ un quipu réversible. Soit w un idempotent pour $\mathcal{A}$. Quels que soient les mots u et v de A^*, si $u.w.v$ est accepté par une branche de $\mathcal{A}$, $u.v$ est accepté par la même branche.*

Démonstration. Ceci vient du fait que dans un automate réversible, un idempotent ne peut étiqueter qu'une boucle. En effet, si w est un idempotent pour $\mathcal{A}$ qui étiquette un chemin de p à q, il étiquette aussi une boucle autour de q et comme l'automate est réversible, on obtient $p = q$. Par conséquent, de tout calcul de $\mathcal{A}$ étiqueté par $u.w.v$, avec w idempotent pour $\mathcal{A}$, on peut extraire un calcul étiqueté par $u.v$. □

Le monoïde de transition qui reconnaît un langage réversible n'est pas caractéristique d'un tel langage, puisqu'il faut, en plus, respecter certaines propriétés de fermeture. De fait, contrairement à ce que nous avons fait pour les groupes, on ne peut pas effectuer une partition du langage réversible par rapport aux élément du monoïde syntaxique de manière à obtenir des langages réversibles. Par exemple, on ne peut pas séparer le langage a^*+b^* en deux langages réversibles disjoints unions d'images réciproques d'éléments de son monoïde syntaxique. En fait, on va décomposer le langage en un ensemble de langages réversibles (non nécessairement disjoints); chacun de ces langages correspondant à une corde d'un quipu réversible minimal. Chaque corde compte un certain nombre d'états-clés qui sont les états d'entrée et de sortie des différentes pelotes. Tout chemin réussi dans une corde emprunte tous ces états.

DÉFINITION 4.12 *Soit $\mathcal{D} = \langle Q,A,E,i,t \rangle$ une corde. Les pelotes de $\mathcal{D}$ sont totalement ordonnées :*

$$\mathcal{P} < \mathcal{P}' \Leftrightarrow \forall p \in \mathcal{P}, \forall p' \in \mathcal{P}, \exists u \in A^*,\ p \cdot u = p'.$$

Soit $\mathcal{P}_1 < \mathcal{P}_2 < \ldots < \mathcal{P}_k$ les pelotes de $\mathcal{D}$. Pour tout j de $[1;k]$, l'état p_j (resp. q_j) est le seul état de $\mathcal{P}_j$ qui soit état d'arrivée (resp. de départ) d'une transition extérieure à $\mathcal{P}_j$. Si l'état i appartient à $\mathcal{P}_1$, $p_1 = i$; de même, si l'état t appartient à $\mathcal{P}_k$, $q_k = t$. Le $2k$-uplet $(p_1,q_1,p_2,\ldots,p_k,q_k)$ est appelé jalon de $\mathcal{D}$. Tout mot u accepté par $\mathcal{D}$ se décompose de façon unique en $u = v_0.u_1.v_1.u_2 \ldots u_k.v_k$, avec

$$\begin{aligned} i \cdot v_0 &= p_1 & \qquad p_j \cdot u_j &= q_j,\ \forall j \in [1;k] \\ q_k \cdot v_k &= t & \qquad q_j \cdot v_j &= p_{j+1},\ \forall j \in [1;k+1] \end{aligned}$$

Le lemme suivant est l'exacte généralisation du lemme 4.6.

LEMME 4.10 *Soit $\mathcal{L}$ un langage réversible et $\mathcal{A}$ un quipu réversible minimal qui reconnaît le langage $\mathcal{L}$ et $\mathcal{B}$ un automate équivalent. Soit $\mathcal{D}$ une corde de $\mathcal{A}$, $(p_1,q_1,\ldots,q_k)$ le jalon de $\mathcal{D}$ et u un mot accepté par $\mathcal{D}$, dont la décomposition selon $\mathcal{D}$ est $v_0,u_1,v_1,\ldots,v_k$. Il existe k idempotents pour $\mathcal{A}$, $w_1,\ldots,w_k$ et k états $r_1,\ldots,r_k$ de $\mathcal{B}$ tels que, pour tout i de $[1;k]$:*

i) *Le mot w_i étiquette une boucle autour de l'état p_i.*

ii) *Il existe des entiers h_i et l_i tels que*

$$\begin{aligned} v_0.w_1^{h_1} &\in \mathsf{Past}_{\mathcal{B}}(r_1), \\ w_k^{l_k}.u_k.v_k &\in \mathsf{Fut}_{\mathcal{B}}(r_k), \\ w_i^{l_i}.u_i.v_i.w_{i+1}^{h_{i+1}} &\in \mathsf{Trans}_{\mathcal{B}}(r_i,r_{i+1}),\ \text{pour } i < k. \end{aligned}$$

iii) *Si un mot y étiquette une boucle autour de p_i, il existe un mot x idempotent et un mot z tels que $x.y.z$ étiquette une boucle autour de r_i.*

Démonstration. La situation décrite dans le lemme est représentée figure 5.
Soit l un entier positif. Pour tout i, on pose $E_i = \{u \in A^* \mid p_i \cdot u = p_i \text{ et } |u| \leqslant l\}$. Il existe un entier k tel que pour tout mot u de A^*, le mot u^{k} est un idempotent pour $\mathcal{A}$. On forme le mot

$$w_i = \prod_{u \in E_i} u^{\mathsf{k}}$$

On choisit un ordre arbitraire sur les mots de E_i pour effectuer ce produit. Le mot w_i étiquette une boucle autour de p_i. Soit n le nombre d'états de $\mathcal{B}$. On considère le mot

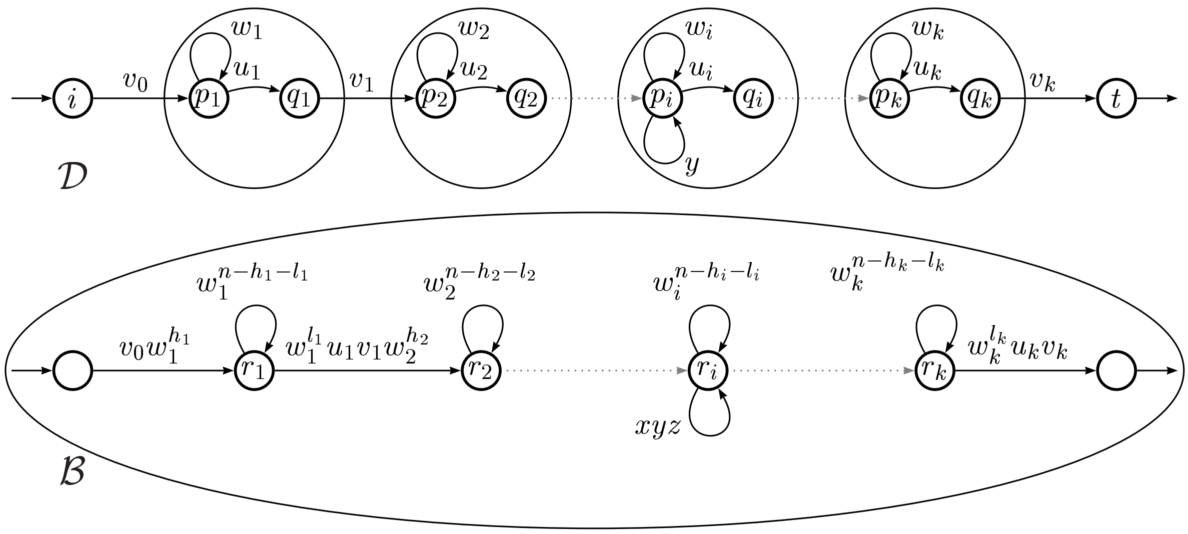
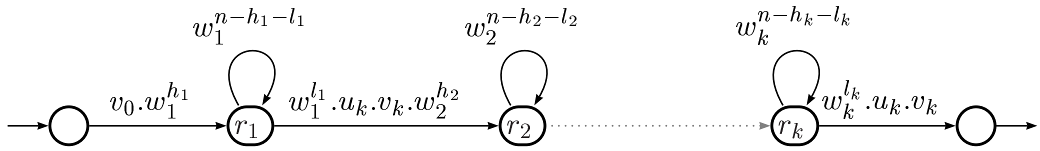


FIG. 5 – *Une corde, ses jalons, et sa trace dans un automate qui accepte le langage.*

$v_0.w_1^n.u_1.v_1 \dots w_k.u_k.v_k$, qui est accepté par $\mathcal{D}$, donc par $\mathcal{B}$. Par le lemme des tiroirs, un calcul de $\mathcal{B}$ étiqueté par ce mot peut se décomposer de la manière suivante :



Les états r_i ainsi définis vérifient i) et ii). Tout mot y de longueur inférieure à l qui fait une boucle autour de p_i, appartient à E_i. Donc $w_i = x_0.y.z$, avec x idempotent pour $\mathcal{A}$, et $w_i^{n-h_i-l_i} = w_i^{n-h_i-l_i-1}.x_0.y.z$. En posant $x = w_i^{n-h_i-l_i-1}.x_0$, on montre iii) pour les mots de longueur inférieure à l.

Pour tout l, on peut trouver des états $(r_1,r_2,\dots,r_k)$ qui vérifient i), ii) et iii) pour les mots de longueur inférieure à l. Quand on fait tendre l vers l'infini, le nombre de k-uplets d'états différents étant fini, il y en a un qui apparaît infiniment souvent. Celui-ci vérifie le lemme pour des mots de longueur arbitrairement grande. □

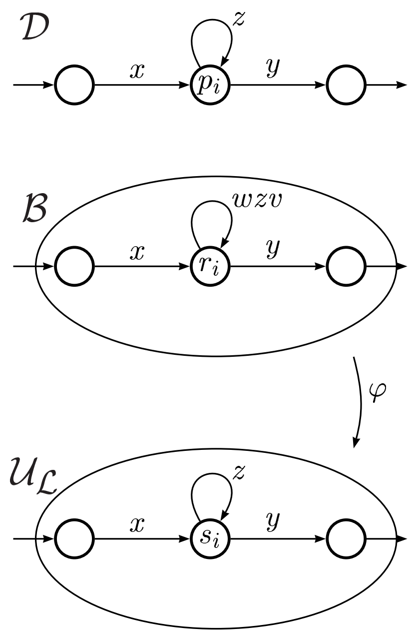
THÉORÈME 4.3 *Soit $\mathcal{L}$ un langage réversible. Il existe un sous-automate quasi-réversible de l'automate universel de $\mathcal{L}$ qui reconnaît $\mathcal{L}$ et qui est d'enlacement minimal.*

Démonstration. La preuve du théorème se fait en plusieurs étapes. D'abord, on pose les outils utilisés.

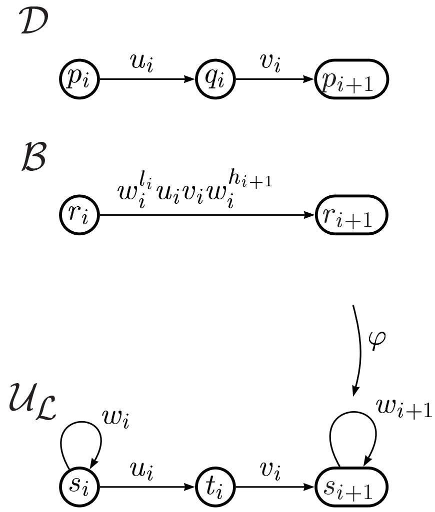
Soit $\mathcal{A}$ un quipu réversible minimal qui reconnaît le langage $\mathcal{L}$ et $\mathcal{B}$ un automate qui reconnaît ce même langage et qui est d'enlacement minimum. $\mathcal{B}$ a n états. On considère une corde $\mathcal{D}$ de $\mathcal{A}$ et u un mot accepté uniquement par cette corde. Soit $(p_1,q_1,p_2,q_2,\dots,q_k)$ le jalon de $\mathcal{D}$ et $(v_0,u_1,u_2,\dots,u_k,v_k)$ la décomposition de u par rapport à $\mathcal{D}$.
Soit $r_1,\dots,r_k$ et $w_1,\dots,w_k$ les états de $\mathcal{B}$ et les idempotents pour $\mathcal{A}$ donnés par le lemme 4.10 appliqué à $\mathcal{D}$ et u. La situation est donc celle schématisée figure 5.
Soit φ un morphisme de $\mathcal{B}$ dans $\mathcal{U}_{\mathcal{L}}$. On fixe un entier i de $[1;k]$.

On pose $s_i = r_i\varphi$ et $\mathcal{P}_i$ est la pelote de $\mathcal{U}_{\mathcal{L}}$ qui contient s_i. Il existe des entiers h_1, $\dots$,h_i (*resp.* $l_i,\dots$,l_k) tels que le mot $x = u_0.w_1^{h_1}.v_1.u_1\dots u_{i-1}.w_i^{h_i}$ (*resp.* le mot $y = w_i^{l_i}.v_i.u_i\dots w_k^{l_k}.v_k.u_k$) appartient au passé (*resp.* au futur) de r_i et donc de s_i.

On montre que φ est un morphisme conforme sur $\mathcal{P}_i$. Soit $\mathcal{C}$ un chemin de $\mathcal{P}_i$. On peut supposer, quitte à le prolonger, que $\mathcal{C}$ est une boucle autour de s_i et, quitte à tourner suffisament de fois autour de s_i, que $\mathcal{C}$ est étiqueté par un idempotent z pour $\mathcal{A}$.



Le mot $x.z.y$ appartient au langage. D'après le lemme 4.9, toute corde de $\mathcal{A}$ qui accepte ce mot accepte aussi le mot u, donc ce mot est accepté par la seule corde $\mathcal{D}$. Comme la corde est réversible, le début de calcul étiqueté par x mène en p_i et la fin de calcul étiquetée par y vient de p_i. Le mot z étiquette donc une boucle autour de p_i. Donc, d'après les propriétés de l'état r_i, il existe un mot w idempotent pour $\mathcal{A}$ et un mot v tels que $w.z.v$ étiquette une boucle autour de r_i. L'image de cette boucle par φ est une boucle autour de s_i. Comme w est un idempotent (dans le monoïde de transition donc *a fortiori* dans le monoïde syntaxique), d'après la proposition 3.12, il étiquette une boucle autour de s_i. Comme les pelotes sont déterministes, la boucle $\mathcal{C}$ autour de s_i étiquetée par z est bien image de la portion de boucle autour de r_i étiquetée par z. Le morphisme φ est donc conforme sur $\mathcal{P}_i$ dont l'enlacement est, par conséquence, inférieur à celui de $\mathcal{B}$.



On montre qu'on peut relier deux pelotes $\mathcal{P}_i$ et $\mathcal{P}_{i+1}$ sans former de nouvelle boucle. Le mot u_i peut être prolongé de manière à étiqueter une boucle autour de p_i dans $\mathcal{D}$, donc il existe un idempotent x pour $\mathcal{A}$ et un mot z tels que $x.u_i.z$ étiquette un mot autour de r_i, donc de s_i et comme x est un idempotent, il existe un chemin étiqueté par u_i dans $\mathcal{P}_i$ qui part de s_i. Soit t_i son état d'arrivée.
De même, en utilisant la proposition 3.12, les mots w_i et w_{i+1} étiquettent respectivement des boucles autour de s_i et s_{i+1}.

Il existe deux entiers l_i et h_{i+1} tels que $w_i^{l_i}.u_i.v_i.w_{i+1}^{h_{i+1}}$ appartienne à $\mathsf{Trans}_{\mathcal{B}}(r_i,r_{i+1})$ donc à $\mathsf{Trans}_{\mathcal{U}_{\mathcal{L}}}(s_i,s_{i+1})$. Posons $s_i = (L_i,R_i)$ et $t_i = (L'_i,R'_i)$. Par définition de l'automate universel :

$$L_i.w_i^{l_i}.u_i.v_i.w_{i+1}^{h_{i+1}} \subseteq L_{i+1}$$

Soit μ le morphisme syntaxique de $\mathcal{L}$. D'après la proposition 3.11, $(L_i.w_i^{l_i})\mu = L_i\mu$ et $(L_i.u_i)\mu = L'_i\mu$. Donc il existe un chemin étiqueté par $v_i.w_{i+1}^{h_{i+1}}$ entre t_i et s_{i+1}. On a donc :

$$R'_i \subseteq v_i.w_{i+1}^{h_{i+1}}.R_{i+1}$$

De même, $(w_{i+1}^{h_{i+1}}.R_{i+1})\mu = R_{i+1}\mu$, donc il existe un chemin étiqueté par v_i entre t_i et s_{i+1}. Ceci est vrai pour tout i de $[1; k-1]$. On montre de même qu'il existe un chemin d'un état initial à s_1 étiqueté par v_0 et un chemin de s_k à un état terminal étiqueté par v_k.

Si un des états intérieurs d'un de ces chemins étiqueté par v_i appartient à une pelote, on peut factoriser v_i en $x_i.y_i$ et il existe un idempotent w pour $\mathcal{A}$ tel que

$$v_0u_1v_1 \dots u_i.x_i.w.y_i.u_{i+1}.v_{i+1} \dots u_k.v_k \in \mathcal{L}.$$

Il ne peut être accepté que par $\mathcal{D}$ (lemme 4.9), ce qui contredit qu'il n'existe pas de boucle entre q_i et p_{i+1}.

On montre que le sous-automate de $\mathcal{U_L}$ ainsi formé accepte tous les mots de $\mathcal{D}$. Tout mot accepté par $\mathcal{D}$ est de la forme $v_0.u'_1.v_1 \dots u'_k.v_k$, avec $p_i \cdot u'_i = q_i$. Il suffit de montrer que u'_i appartient à $\mathsf{Trans}_{\mathcal{U_L}}(s_i,t_i)$. Il existe un mot w_i tel que $u'_i.w_i$ est un idempotent et $u'_i.w_i$ et $u_i.w_i$ étiquettent des boucles autour de p_i. De même que précédemment, ces mots étiquettent des boucles autour de s_i et comme la pelote est co-déterministe, le chemin partant de s_i étiqueté par u'_i arrive dans le même état que celui étiqueté par u_i, c'est-à-dire t_i.

On montre que ce sous-automate est réversible. Les pelotes de l'automate universel sont réversibles. Entre chaque pelote, l'automate ne comporte qu'un seul chemin, par construction. S'il existe une lettre a qui étiquette deux transitions arrivant en s_i, cette lettre est la dernière de v_i et on montre qu'il existe une boucle autour de p_i dont la dernière étiquette est a, ce qui contredit la réversibilité de $\mathcal{D}$. Donc le sous-automate est co-déterministe; il est de même déterministe.

Conclusion. Pour chaque corde $\mathcal{D}$ de $\mathcal{A}$, on peut montrer qu'il existe un sous-automate de l'automate universel d'enlacement inférieur à la hauteur d'étoile du langage, et qui accepte au moins le langage reconnu par cette corde. La superposition de tous ces automates dans l'automate universel forme un automate qui reconnaît le langage $\mathcal{L}$. De plus, on a vu que pour chaque état de chacun des sous-automates, si l'état appartient à une pelote, la pelote en intégralité appartient au sous-automate. Chaque pelote de la « superposition » est donc contenue dans un des sous-automates. L'enlacement de l'automate obtenu est donc égal au maximum des enlacement obtenus en considérant les branches séparément, c'est-à-dire inférieur ou égal à l'enlacement de $\mathcal{B}$ qui est minimum par hypothèse. Si deux transitions étiquetées par la même lettre arrivent dans un même état, une d'entre elle n'appartient à aucune pelote. Le sous-automate dont elle provient est réversible et contient les pelotes de $\mathcal{U}_{\mathcal{L}}$ qu'il intersecte en intégralité; la seconde n'appartient donc à aucune pelote non plus. De même pour les transitions partant d'un même état. L'automate est donc quasi-réversible. Ce raisonnement est similaire à celui de la preuve du théorème 3.4. Il existe donc un sous-automate quasi-réversible de l'automate universel qui est d'enlacement minimum pour le langage. □

Le résultat de R. Cohen [17] est alors un corollaire du théorème.

COROLLAIRE 4.11 *Soit $\mathcal{L}$ un langage réversible dont l'automate minimal est réversible et n'a qu'un état final. Alors, l'automate minimal de $\mathcal{L}$ est d'enlacement minimum.*

La raison est exactement la même que pour les groupes. On a en effet vu que dans ce cas, l'automate universel est isomorphe à l'automate minimal (proposition 3.10). Par minimalité, le seul sous-automate de l'automate universel qui accepte le langage est donc l'automate universel lui-même.

EXEMPLE 12.4 Cet algorithme, appliqué à l'automate universel de $\mathcal{L}_{r_1}$, permet d'obtenir l'automate présenté figure 6. Cet automate est quasi-réversible et contient un sous-

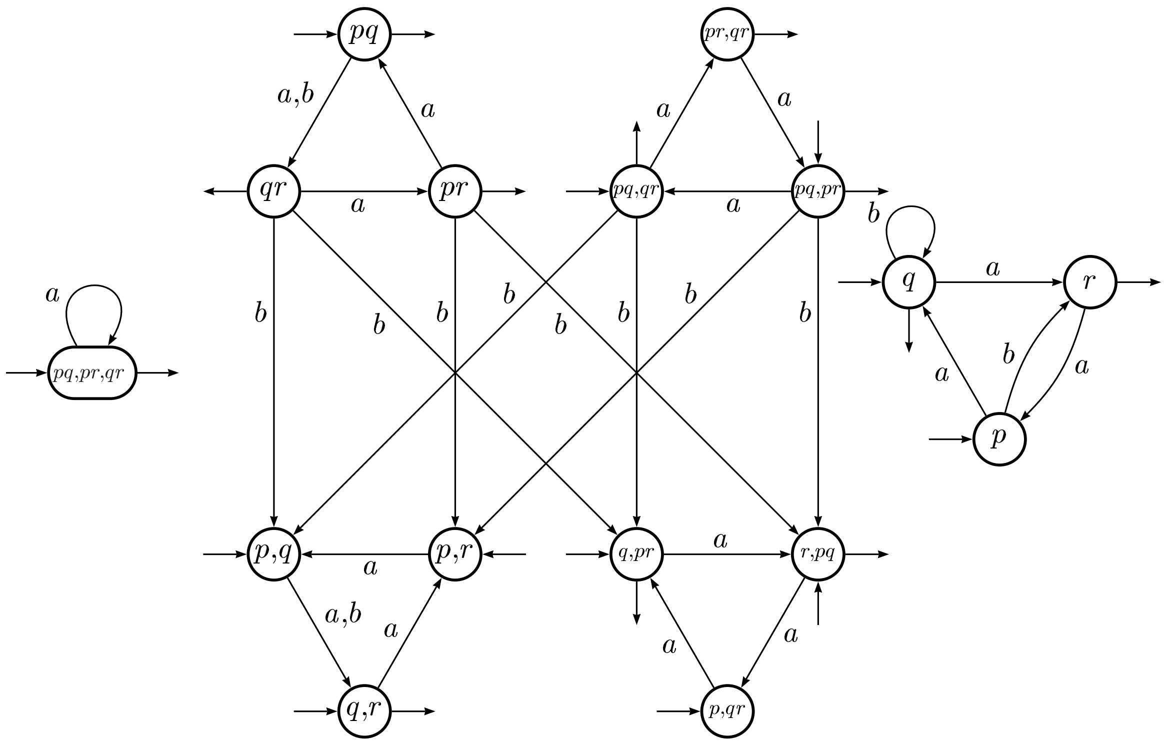


FIG. 6 – *Un automate quasi-réversible pour $\mathcal{L}_{r_1}$.*

automate d'enlacement minimal pour le langage. De plus, on sait qu'il existe un tel sous-automate qui recouvre les pelotes qu'il intersecte. On va donc considérer des sous-automates formés de composantes fortement connexes « entières ». Or, le mot b^4 de $\mathcal{L}_{r_1}$

n'est reconnu que par la composante formée des états p, q et r. L'automate d'enlacement minimal contient donc au moins cette pelote. Celle-ci reconnaît tout le langage; elle forme donc un automate quasi-réversible (ici, même, réversible) d'enlacement minimal (égal à 2) qui reconnaît le langage. La hauteur d'étoile du langage $\mathcal{L}_{r_1}$ est donc égale à 2.

— ∘ —

4 Automate universel et hauteur d'étoile

Rien n'empêche d'exprimer les résultats obtenus pour les langages réversibles ou les langages à groupe dans le cas général.

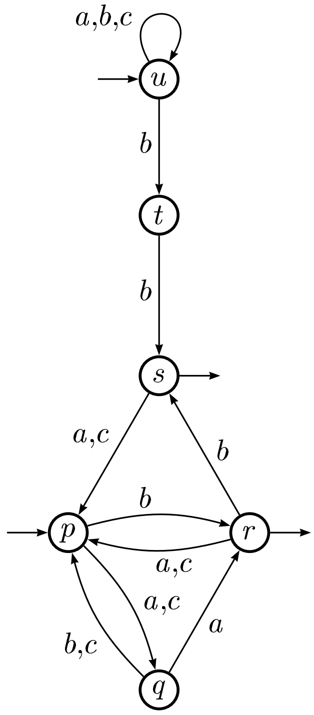
CONJECTURE 4.1 *Soit $\mathcal{L}$ un langage rationnel de A^* et $\mathcal{U}_{\mathcal{L}}$ l'automate universel de $\mathcal{L}$. L'automate $\mathcal{U}_{\mathcal{L}}$ contient un sous-automate qui reconnaît $\mathcal{L}$ avec un enlacement minimum.*

Jusqu'à présent, rien n'est venu contredire cette conjecture. Toutefois, il semble difficile d'adapter les techniques employées précédemment pour résoudre ce problème. L'exemple ci-dessous reflète les difficultés qu'on peut rencontrer.

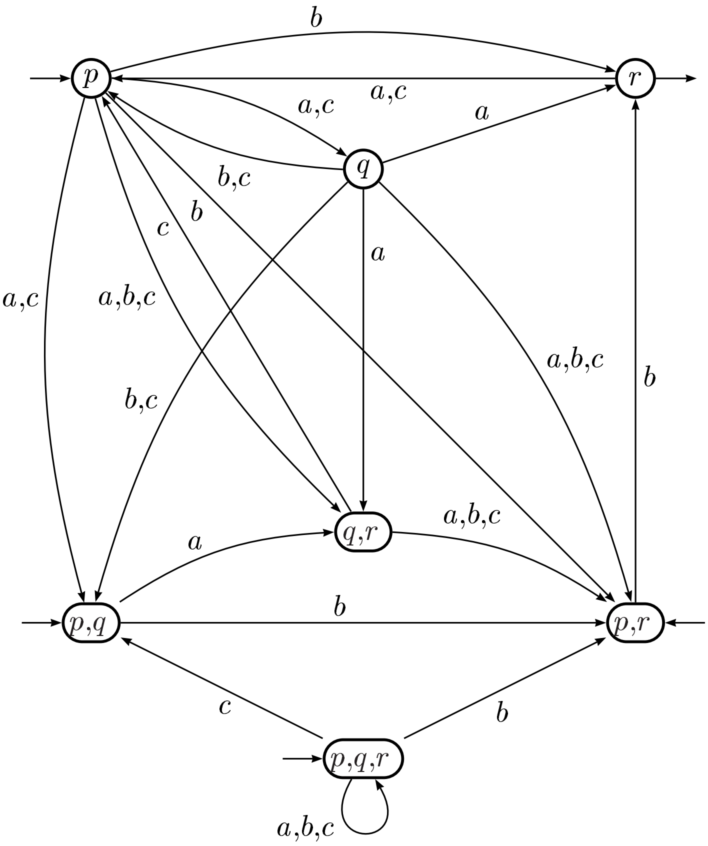
EXEMPLE 10.3 Si l'on regarde attentivement l'automate minimal de $\mathcal{L}_4$ (voir figure 11 page 60), on s'aperçoit que tout facteur bb mène dans l'état 3 et que le langage $\mathcal{L}_4$ est donc reconnu par l'automate $\mathcal{A}_4$ présenté figure 7 a), dont l'enlacement est égal à 1. La hauteur d'étoile de $\mathcal{L}_4$ est donc 1. L'automate universel de $\mathcal{L}_4$ a un seul état terminal. Les états r et s de $\mathcal{A}_4$ ont donc la même image dans $\mathcal{U}_{\mathcal{L}_4}$ L'image par n'importe quel morphisme de cet automate dans l'automate universel de $\mathcal{L}_4$ est donc d'enlacement 2. On peut trouver dans ce dernier un sous-automate d'enlacement 1 qui accepte tous les mots du langage : l'automate de la figure 7 b) est d'enlacement 1; si on supprime l'état p, la composante fortement connexe composée de tous les états hormi $\{p,q,r\}$ devient acyclique.

Cet exemple pointe la difficulté de prouver le résultat en général. Comment utiliser le fait qu'il existe un automate d'enlacement minimum pour prouver qu'il en existe un dans l'automate universel ? On voit ici que le premier peut n'avoir aucun rapport avec le second.

— ∘ —



(a) L'automate $\mathcal{A}_4$.



(b) Un sous automate de $\mathcal{U}_{\mathcal{L}_4}$.

FIG. 7 – *Deux automates d'enlacement minimum pour* $\mathcal{L}_4$.

Chapitre 5

Déterminisation des automates $(\mathsf{max}\,,+)$

Le semi-anneau tropical $\mathcal{M}$ a été introduit pour calculer le coût minimal de certaines opérations en recherche opérationnelle. Imaginons un automate dans lequel chaque transition est pondérée par un entier (le coût de la transition). Si un mot représente une suite d'opérations à effectuer, chaque calcul étiqueté par ce mot a un coût (la somme des coûts des transitions); on désire calculer le coût minimal pour lire un mot donné. La somme sera donc la multiplication du semi-anneau et le min son addition.

Ce semi-anneau a aussi été étudié pour résoudre de nombreux problèmes liés à la hauteur d'étoile d'un langage rationnel [32, 34, 41, 58, 60].

D'autres semi-anneaux voisins peuvent être considérés. On peut supposer que chaque transition représente au contraire un gain, on veut alors emprunter le parcours correspondant au gain maximum. L'addition du semi-anneau dans lequel on se place sera alors le max. De même, le support du semi-anneau n'est pas nécessairement restreint à $\mathbb{N}$.

Un problème naturel qui se pose pour les séries rationnelles à coefficients dans ces semi-anneaux est celui de leur séquentialité [1]. Si les séries sont réalisées par des automates non ambigus, on peut adapter les techniques développées par C. Choffrut [14] pour les transducteurs. Ceci a été fait par M. Mohri [49]. Toutefois, les séries réalisées par ce type d'automates sont très particulières, car, on le voit, l'opérateur max n'intervient pas dans le calcul d'un coefficient. Nous montrerons qu'il existe des séries rationnelles qui ne peuvent pas être réalisées par des automates non ambigus. Ceci remet en cause, en particulier, la caractérisation des séries séquentielles par la propriété de *divergence uniforme*.

En effet, dans le cadre des transducteurs, avant de décider de la séquentialité, on décide de la *fonctionnalité* de la série réalisé, ce qui, pour les automates $(\mathsf{max}\,,+)$ revient à dire que chaque calcul étiqueté par un mot donné a la même valeur (nous appellerons un tel automate *univoque*). Par l'intermédiaire du théorème de « cross-section » [23] ou par une construction directe sur l'automate univoque [54], on peut alors construire un automate non ambigu qui réalise la même série.

1. Le problème apparaît d'ailleurs naturel quel que soit le semi-anneau des coefficients.

En toute généralité, le problème s'avère beaucoup plus compliqué. Après avoir donné des propriétés des séries séquentielles et pointé celles qui sont caractéristiques et celles qui ne le sont pas, on montre dans ce chapitre comment décider de la séquentialité d'une série sur un alphabet unaire. Nous montrons que la complexité de cet algorithme ne dépend pas de la taille des coefficients.

L'algorithme de décision s'apparente dans une certaine mesure au théorème de cyclicité sur les matrices à coefficients dans les semi-anneaux (max ,+) ([26]), sauf que notre but nous permet de restreindre notre étude à la valeur propre maximale et que ce ce ne sont pas exactement les vecteurs propres qui nous intéressent. En effet, la matrice de transition de l'automate que l'on considère peut avoir plusieurs valeurs propres et la séquentialité de la série réalisée dépend non seulement de la matrice de transition, mais aussi des états initiaux ou terminaux de l'automate.

S. Gaubert [24] a montré que toutes les séries (max ,+) rationnelles sur une lettre peuvent être obtenue par *fusion*[2] (c'est-à-dire somme non ambiguë) de séries séquentielles. Ce résultat permet de transformer un automate en automate non ambigu puis d'appliquer le résultat générique sur les automates non ambigus. Le passage par un automate non ambigu peut toutefois se révéler fort coûteux; nous montrons qu'il n'est pas nécessaire pour décider de la séquentialité. Nous donnons, comme conséquence de l'algorithme de déterminisation, une construction (différente de celle de S. Gaubert) d'un automate non ambigu qui accepte une série rationnelle donnée et nous montrons que la taille minimale d'un tel automate peut être arbitrairement grande devant la taille de l'automate de départ.

Nous revenons ensuite sur les automates *univoques* pour expliciter l'adaptation de méthodes données par M.-P. Béal, O. Carton, C. Prieur, et J. Sakarovitch en [5] pour résoudre le problème dans ce cas particulier, ce qui n'apporte aucun résultat de décidabilité nouveau par rapport à [49], mais permet de décider sur ces automates sans recourir à une désambiguïsation qui peut se révéler exponentielle.

En guise de conclusion, nous faisons part d'un résultat de S. Gaubert qui met en rapport le problème de la limitation et celui de la déterminisation, et nous expliquons pourquoi les méthodes mises en œuvre dans ce chapitre sont difficilement extensibles à d'autres semi-anneaux.

1 Le semi-anneau tropical et sa famille

Définition 5.1 *Soit $\mathbb{G}$ un sous-groupe de $(\mathbb{R},+)$, $\mathbb{G}_+$ son intersection avec $\mathbb{R}_+$ et $\mathbb{G}_-$ son intersection avec $\mathbb{R}_-$. On pose $\mathbb{K} = \mathbb{G}$, $\mathbb{G}_+$ ou $\mathbb{G}_-$. Dans les deux derniers cas, on dit que $\mathbb{K}$ est respectivement positif ou négatif.*

$$\mathbb{K}_m = (\mathbb{K} \cup \{\infty\}, \mathsf{min}\,,+) \textit{ est un semi-anneau } (\mathsf{min}\,,+),$$
$$\mathbb{K}_{\mathsf{m}} = (\mathbb{K} \cup \{-\infty\}, \mathsf{max}\,,+) \textit{ est un semi-anneau } (\mathsf{max}\,,+).$$

Le ***semi-anneau tropical*** *est le semi-anneau* $\mathcal{M} = \mathbb{N}_m = (\mathbb{N} \cup \{\infty\}, \mathsf{min}\,,+)$.

2. *merge* en anglais.

Proposition 5.1 *Le semi-anneau $\mathbb{K}_m$ est isomorphe au semi-anneau $(-\mathbb{K})_{\mathsf{m}}$.*

Cette proposition nous permet de restreindre notre étude aux semi-anneaux $(\mathsf{max},+)$.

Nous allons donc étudier les séries rationnelles de $\mathbb{K}\langle\langle A^*\rangle\rangle$ (ou de $\mathbb{K}_{\mathsf{m}}\langle\langle A^*\rangle\rangle$ selon qu'on les considère comme des fonctions à valeur dans $\mathbb{K}$ ou des applications à valeurs dans $\mathbb{K}_{\mathsf{m}}$). Le support d'une telle série α est l'ensemble des mots u de A^* tels que $\langle\alpha,u\rangle \neq -\infty$.

D'autre part, les semi-anneaux $(\mathsf{max},+)$ ont ceci de particulier qu'une somme finie d'éléments est toujours égale à l'un des termes. Dans un automate $\mathcal{A}$ qui réalise une série α de $\mathbb{K}_{\mathsf{m}}\langle\langle A^*\rangle\rangle$, pour tout mot u, il existe donc un calcul étiqueté par u dont la valeur est exactement $\langle\alpha,u\rangle$. On appellera un tel calcul un **calcul victorieux**.

2 Caractérisation des séries séquentielles

2.1 Séries translatées

Le théorème 1.5 nous indique qu'une série rationnelle sur A à coefficients dans un semi-anneau $\mathbb{K}$ est séquentielle si et seulement si ses quotients appartiennent à un $\mathbb{K}$-cône finiment engendré. Nous allons essayer de définir un ensemble générateurs canoniques de ce cône. Si le pgcd est uniquement défini, chaque série α peut alors s'écrire de façon unique $k \otimes \alpha'$, où k est le pgcd des coefficients de α.

Si $\mathbb{K}$ est un sous-groupe de $\mathbb{G}$, le pgcd est loin d'être unique (on peut *factoriser* les éléments de $\mathbb{K}_m$ par n'importe quel élément de $\mathbb{K}$). Il nous faudra donc adapter notre définition. En revanche, si $\mathbb{K}$ est positif ou négatif, le pgcd est uniquement défini dans le semi-anneau $\mathbb{K}_{\mathsf{m}}$. On peut donc rechercher les générateurs *canoniques* du cône qui contient les quotients. Si $\mathbb{K}$ est positif, le pgcd de deux nombres a et b de $\mathbb{K}_{\mathsf{m}}$ différents de $-\infty$ est égal au min de ces deux nombres (car la multiplication du semi-anneau est la somme usuelle); s'il est négatif, il s'agit du max.

Nous définissons donc le pgcd des coefficients du quotient d'une série α par un mot u de la façon suivante :

Définition 5.2 *Soit α une série de $\mathbb{K}_{\mathsf{m}}\langle\langle A^*\rangle\rangle$. On définit $\mathring{\alpha}$ par :*

$$\langle\mathring{\alpha},u\rangle = \begin{cases} \mathsf{inf}\{\langle u^{-1}\alpha,v\rangle \mid v \in u^{-1}\,\mathsf{Supp}(\alpha)\}, \text{ si } \mathbb{K} \text{ est positif,} \\ \mathsf{sup}\{\langle u^{-1}\alpha,v\rangle \mid v \in u^{-1}\,\mathsf{Supp}(\alpha)\}, \text{ si } \mathbb{K} \text{ est négatif,} \\ \langle u^{-1}\alpha,v\rangle, \text{ avec } v \text{ minimal pour l'ordre radiciel si } \mathbb{K} \text{ est un groupe.} \end{cases}$$

On définit ainsi une série $\mathring{\alpha}$ dont le support est A^*.

Remarque 5.1 Supposons α rationnelle et $\mathbb{K}$ négatif. Soit n le nombre d'états d'un automate qui réalise α.[3] Pour tout u dans $\mathsf{Supp}(\alpha)$ de longueur supérieur à n, tout calcul victorieux étiqueté par u comporte une boucle étiquetée par v. On peut décomposer u en $x.v.y$. Il existe donc un chemin étiqueté par $x.y$ dont la valeur est supérieure à la valeur

3. D'après la remarque 1.14, chaque quotient de α peut etre réalisé par un automate à n états.

de tout calcul victorieux étiqueté par u, donc $\langle \alpha,u \rangle \leqslant \langle \alpha,x.y \rangle$. Si α est rationnelle, ses quotients aussi, le « sup » employé dans la définition ci-dessus peut donc être appliqué à un ensemble fini; c'est alors un « max ».

Si $\mathbb{K}$ est positif et α rationnelle, l'« inf » est évidemment aussi un « min ». Toutefois, ce minimum peut être réalisé par un mot de longueur non polynomiale dans la taille de l'automate. Le calcul effectif du inf s'en trouve compliqué. Ce cas est illustré par l'exemple suivant qui doit beaucoup à la construction de M. Chrobak dans [15].

Exemple 20 Soit k un entier et A un alphabet unaire. On définit, pour tout i dans $[1;k]$, un automate $\mathcal{A}_i$ à multiplicité sur $\mathbb{N}_{\mathsf{m}}$, qui est un circuit de n_i états $p_{i,1},p_{i,2},\ldots,p_{i,n_i}$:

$$E_{(p_{i,k},a,p_{i,l})} = \begin{cases} 0 & \text{si } l-k=1 \mod n_i \\ -\infty & \text{sinon.} \end{cases}$$

Pour tout i, seul l'état $p_{i,1}$ est initial :

$$I_{p_{i,k}} = \begin{cases} 0 & \text{si } k=1 \\ -\infty & \text{sinon.} \end{cases}$$

Pour tout $i > 1$, les états différents de p_{i,n_i} sont terminaux, avec multiplicité 1 ($T_{p_{i,k}} = 1$ si $k \neq n_i$ et $T_{p_{i,k}} = -\infty$).

$$T_{p_{i,k}} = \begin{cases} -\infty & \text{si } k=n_i \\ 1 & \text{sinon.} \end{cases}$$

Pour $i = 1$, l'état p_{1,n_1} est terminal, mais avec multiplicité 0 :

$$T_{p_{i,k}} = \begin{cases} 0 & \text{si } k=n_i \\ 1 & \text{sinon.} \end{cases}$$

On considère la série réalisée par l'union de ces automates. Son support est a^*. Le plus petit mot dont la valeur maximale sur les chemins est 0 est de longueur $\mathsf{ppcm}\{n_i \mid i \in [1;k]\}-1$. Si on suppose que la somme des n_i est égale à n fixé, quelle est la valeur maximale $F(n)$ de leur ppcm? Ce problème classique d'arithmétique est connu comme problème de Landau. Il a été montré ([63]) que $F(n) = O(e^{\sqrt{n \ln n}})$, ce qui conclut notre exemple.

La définition précédente nous permet de trouver un système générateur du cône des quotients :

Définition 5.3 *La* ***translatée*** *(à gauche) de la série α de $\mathbb{K}\langle\langle A^*\rangle\rangle$ par un mot u de A^*, notée α/u, est définie par :*

$$\forall v \in A^*,\ \langle \alpha/u,v \rangle = \langle u^{-1}\alpha,v \rangle - \langle \mathring{\alpha},u \rangle.$$

Noter qu'on utilise le signe « $-$ », c'est-à-dire la division du semi-anneau. Ceci est possible car $\langle \mathring{\alpha},u \rangle$ est justement défini comme un pgcd des coefficients de $u^{-1}\alpha$.

De même que pour les quotients, la définition des translatées induit la définition d'une action à droite de A^* sur les séries, puisque, pour tout mot u et toute lettre a, pour toute série α, $(\alpha/u)/a = \alpha/(u.a)$.

On va montrer qu'on obtient bien la caractérisation voulue, à savoir :

Proposition 5.2 *Une série de* $\mathbb{K}\mathrm{Rat}\, A^*$ *est séquentielle si et seulement si le nombre de ses translatées est fini.*

Lemme 5.3 *Soit u et u' deux mots de A^*. Si $u^{-1}\,\mathsf{Supp}(\alpha) = u'^{-1}\,\mathsf{Supp}(\alpha)$ et qu'il existe une constante $c_{u,u'}$ de $\mathbb{R}$ telle que, pour tout mot v de $u^{-1}\,\mathsf{Supp}(\alpha)$, $\langle \alpha,u.v\rangle - \langle \alpha,u'.v\rangle = c_{u,u'}$, alors $c_{u,u'} = \langle \mathring{\alpha},u\rangle - \langle \mathring{\alpha},u'\rangle$ et $\alpha/u = \alpha/u'$.*

Démonstration. Si $\mathbb{K}$ est un groupe, comme $u^{-1}\,\mathsf{Supp}(\alpha) = u'^{-1}\,\mathsf{Supp}(\alpha)$, il existe un mot v (élément minimal de cet ensemble) tel que $\langle \mathring{\alpha},u\rangle = \langle u^{-1}\alpha,v\rangle$ et $\langle \mathring{\alpha},u'\rangle = \langle u'^{-1}\alpha,v\rangle$. Donc $\langle \mathring{\alpha},u\rangle - \langle \mathring{\alpha},u'\rangle = \langle \alpha,u.v\rangle - \langle \alpha,u'.v\rangle = c_{u,u'}$ et, pour tout mot v de $u^{-1}\,\mathsf{Supp}(\alpha)$,

$$\langle \alpha/u,v\rangle = \langle \alpha,u.v\rangle - \langle \mathring{\alpha},u\rangle = \langle \alpha,u'.v\rangle + c_{u,u'} - \langle \mathring{\alpha},u'\rangle - c_{u,u'} = \langle \alpha/u,v\rangle.$$

Si $\mathbb{K}$ est positif, il existe un mot v dans $v \in u^{-1}\,\mathsf{Supp}(\alpha)$ tel que $\langle u^{-1}\alpha,v\rangle = \langle \mathring{\alpha},u\rangle$. Comme pour tout mot w de $u^{-1}\,\mathsf{Supp}(\alpha)$, la distance entre les coefficients de w dans $u^{-1}\alpha$ et $u'^{-1}\alpha$ est constante, le minimum de $u'^{-1}\alpha$ est aussi réalisé en v et $\langle \mathring{\alpha},u'\rangle = \langle u'^{-1}\alpha,v\rangle$. D'où $c_{u,u'} = \langle \mathring{\alpha},u\rangle - \langle \mathring{\alpha},u'\rangle$ et, pour tout mot v de $u^{-1}\,\mathsf{Supp}(\alpha)$,

$$\langle \alpha/u,v\rangle = \langle \alpha,u.v\rangle - \langle \mathring{\alpha},u\rangle = \langle \alpha,u'.v\rangle + c_{u,u'} - \langle \mathring{\alpha},u'\rangle - c_{u,u'} = \langle \alpha/u,v\rangle.$$

De même si $\mathbb{K}$ est négatif. □

Démonstration de la proposition 5.2. Soit α une série de $\mathbb{K}\mathrm{Rat}\, A^*$ dont le nombre de translatées est fini. Posons Q_α l'ensemble des translatées. On définit l'automate $\mathcal{A}_\alpha$ dont l'ensemble d'états est Q_α par sa représentation linéaire (λ,μ,ν) :

$$\begin{aligned} \lambda_\gamma &= \begin{cases} \langle \mathring{\alpha},1_{A^*}\rangle \text{ si } p = \alpha/1_{A^*}, \\ -\infty \text{ sinon,} \end{cases} \\ \nu_\gamma &= \langle p,1_{A^*}\rangle, \\ \forall a \in A,\ (a\mu)_{\gamma,\gamma'} &= \begin{cases} \langle \gamma,a\rangle - \langle \gamma',1_{A^*}\rangle \text{ si } \gamma' = \gamma/a, \\ -\infty \text{ sinon.} \end{cases} \end{aligned}$$

D'une part cet automate est déterministe. En effet, son seul état initial est $\alpha/1_{A^*}$, et si deux mots u et v donnent la même translatée,

$$\mathsf{Supp}(\alpha/(u.a)) = a^{-1}\,\mathsf{Supp}(\alpha/u) = a^{-1}\,\mathsf{Supp}(\alpha/v) = \mathsf{Supp}(\alpha/(v.a))$$

et, pour tout mot w de $\mathsf{Supp}(\alpha/(u.a))$,

$$\langle \alpha,u.a.w\rangle = \langle \mathring{\alpha},u\rangle + \langle \alpha/u,a.w\rangle = \langle \mathring{\alpha},u\rangle + \langle \alpha/v,a.w\rangle = \langle \alpha,v.a.w\rangle + \langle \mathring{\alpha},u\rangle - \langle \mathring{\alpha},v\rangle.$$

Donc, d'après le lemme précédent, $u.a$ et $v.a$ donnent la même translatée. D'autre part cet automate réalise la série α. En effet, pour tout mot $u = u_1 \dots u_k$ (on pose $u_0 = 1_{A^*}$), le calcul étiqueté par u a pour multiplicité :

$$
\begin{aligned}
&\langle \mathring{\alpha},1_{A^*}\rangle + \sum_{i=1}^{k}(\langle \alpha/(u_1 \dots u_{i-1}),u_i\rangle - \langle \alpha/(u_1 \dots u_i),u_{i-1}\rangle) + \langle \alpha/u,1_{A^*}\rangle \\
&\langle \mathring{\alpha},1_{A^*}\rangle + \sum_{i=1}^{k}(\langle \mathring{\alpha},(u_1 \dots u_i)\rangle - \langle \mathring{\alpha},u_1 \dots u_{i-1}\rangle) + \langle \alpha/u,1_{A^*}\rangle \\
&\qquad = \langle \mathring{\alpha},u\rangle + \langle \alpha/u,1_{A^*}\rangle = \langle \alpha,u\rangle
\end{aligned}
$$

Réciproquement, soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate déterministe qui réalise la série séquentielle α. Soit i l'état initial de $\mathcal{A}$. Soit u et v deux mots qui mènent dans le même état p de $\mathcal{A}$. Alors, $u^{-1}\,\mathsf{Supp}(\alpha) = v^{-1}\,\mathsf{Supp}(\alpha)$ (de même que pour un automate déterministe sans multiplicité). Et, pour tout mot w de $u^{-1}\,\mathsf{Supp}(\alpha)$,

$$
\begin{aligned}
\langle \alpha,uw\rangle - \langle \alpha,vw\rangle &= (I_i + i * u + (i \cdot u) * w + T_{(i \cdot uw)}) - (I_i + i * v + (i \cdot v) * w + T_{(i \cdot vw)}) \\
&= i * u - i * v
\end{aligned}
$$

Cette différence ne dépend pas de w. Donc, d'après le lemme précédent, les translatées de α par rapport à u et à v sont égales. Le nombre de translatées de α est donc au plus égal au nombre d'états de $\mathcal{A}$, donc fini. □

Remarque 5.2 On voit dans cette preuve que, si la série est séquentielle, il existe un automate minimal canonique dont les états sont les translatées de la série. Un algorithme de minimisation a été donné par M. Mohri [49].

— ∘ —

2.2 Le problème d'une caractérisation topologique

Dans le cadre de l'étude des transducteurs a été introduite la notion de fonction uniformément divergente (ou fonction « à variations bornées ») [14]. On peut adapter cette définition aux séries rationnelles d'un semi-anneau (max ,+).

Définition 5.4 *Soit α une série de $\mathbb{K}\mathrm{Rat}\,A^*$. La série α est* ***uniformément divergente*** *si, quels que soient les mots u et v du support de α, la distance entre $\langle\alpha,u\rangle$ et $\langle\alpha,v\rangle$ est bornée par une quantité qui dépend uniquement de la distance entre u et v. Formellement :*

$$
\begin{gathered}
\forall k \in \mathbb{N}, \exists N \in \mathbb{R}_+, \forall u,v \in \mathsf{Supp}(\alpha), \\
d_{\mathsf{pref}}(u,v) \leqslant k \Rightarrow |\langle\alpha,u\rangle - \langle\alpha,v\rangle| \leqslant N.
\end{gathered}
$$

Proposition 5.4 *Les séries séquentielles sont uniformément divergentes.*

Démonstration. Si α est séquentielle, le nombre de ses translatées est fini. Posons, pour tout k,

$$N_k = \max_{u \in A^*} \max_{\substack{v \in \mathsf{Supp}(\alpha/u) \\ |v| \leqslant k}} \{\langle \alpha/u, v\rangle\}.$$

La définition de N_k comporte des maxima qui portent en fait sur des ensembles finis, puisque le nombre de translatées est fini et que le nombre de mots de longueur inférieure à k aussi. La quantité N_k est donc bien définie et, elle aussi, finie.
Pour tout couple de mots u et v du support de α, tels que $d_{\mathsf{pref}}(u,v) \leqslant k$, posons $w = u \wedge v$; on peut écrire $u = w.u'$ et $v = w.v'$, avec $|u'| + |v'| \leqslant k$. On obtient :

$$|\langle \alpha, u\rangle - \langle \alpha, v\rangle| = |\langle \alpha/w, u'\rangle - \langle \alpha/w, v'\rangle| \leqslant 2N_k.$$

La série α est donc uniformément divergente. □

Dans le cas des transducteurs cette propriété est caractéristique des fonctions séquentielles [14]. Ce n'est malheureusement pas le cas dans le cadre des séries $(\mathsf{max},+)$, ce qui nous empêche d'adapter un certain nombre de techniques pour décider de la séquentialité, comme l'utilisation de la « twinning property » par exemple. Nous verrons dans quels cas (restreints) on peut étendre les résultats connus sur les transducteurs aux automates $(\mathsf{max},+)$.

Proposition 5.5 *Une série rationnelle uniformément divergente n'est pas nécessairement séquentielle.*

La preuve de cette proposition se présente sous la forme de l'exemple suivant.

Exemple 21 Soit α_1 la série à coefficients dans $\mathbb{N}_{\mathsf{m}}$ sur $\{a,b\}^*$ définie de la façon suivante : pour tout mot u de A^*, $\langle \alpha_1, u\rangle = \mathsf{max}(|u|_a, |u|_b)$. Cette série est bien rationnelle. Elle est réalisée par l'automate présenté figure 1. D'autre part, cette série est uniformément

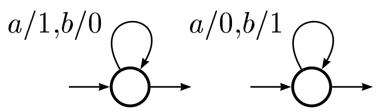


Fig. 1 – *Un automate non déterminisable.*

divergente, elle est même lipschitzienne. En effet, pour tout mot u et tout préfixe w de u ($u = w.u'$), $\langle \alpha_1, u\rangle$ appartient à $[\langle \alpha_1, w\rangle, \langle \alpha_1, w\rangle + |u'|]$; donc, pour tout couple de mots $u = w.u'$ et $v = w.v'$ de A^*, avec $w = u \wedge v$,

$$|\langle \alpha_1, u\rangle - \langle \alpha_1, v\rangle| \leqslant |u'| + |v'| \leqslant d_{\mathsf{pref}}(u,v)$$

Enfin, cette série n'est pas séquentielle. En effet, elle compte un nombre infini de translatées. Par exemple, pour tout couple d'entiers distincts n et m, les séries α/a^n et α/a^m sont distinctes : si $n < m$, $\langle \alpha/a^n, b^m\rangle = m - n$, alors que $\langle \alpha/a^m, b^m\rangle = 0$.

Cette remarque est à mettre en rapport avec le résultat transposé aux automates (max ,+) par M. Mohri [49] selon lequel toute série de $\mathbb{K}\mathrm{Rat}\, A^*$ réalisable par un automate non ambigu qui est uniformément divergente est séquentielle. D'où l'on déduit :

Proposition 5.6 *Il existe des séries rationnelles de* $\mathbb{K}\mathrm{Rat}\, A^*$ *qui ne peuvent pas être réalisées par un automate non ambigu.*

Nous reviendrons plus tard sur la déterminisation des séries réalisées par des automates non ambigus.

3 Décidabilité de la séquentialité dans le cas des alphabets unaires

Nous allons voir que pour un alphabet unaire, les séries rationnelles sont beaucoup plus simples que dans le cas général. Elles sont en effet toutes réalisables de façon non ambiguë, donc la propriété de divergence uniforme est caractéristique des séries séquentielles. Le but est d'obtenir un algorithme de décidabilité qui dépende uniquement de la structure de l'automate de départ, et non de la forme des coefficients.

Pour alléger les notations, nous identifions le monoïde libre engendré par une lettre à $\mathbb{N}$. Dans ce cadre, la représentation linéaire d'un automates dont les états forment un ensemble Q est le triplet (λ,μ,ν), où λ et ν sont des vecteurs de $\mathbb{K}_{\mathsf{m}}^{Q}$ et μ une matrice de $\mathbb{K}_{\mathsf{m}}^{Q\times Q}$. L'automate réalise la fonction α :

$$\forall n \in \mathbb{N},\ \langle \alpha,n\rangle = \lambda \otimes \mu^n \otimes \nu.$$

Définition 5.5 *Le* ***poids d'un circuit*** *d'un automate à multiplicité est le rapport de sa valeur sur sa longueur. On appelle* ***graphe critique*** *l'ensemble des états et des transitions qui appartiennent à des circuits de poids maximum.*

Dans ce qui suit, on utilisera, à propos d'un $\mathbb{K}$-automate $\mathcal{A}$ les notations suivantes :

- M est le maximum des valeurs absolues des coefficients de $\mathcal{A}$ (c'est-à-dire des coefficients des transitions, mais aussi des vecteurs initial et terminal).
- n est le nombre d'états de $\mathcal{A}$.
- ρ désigne le poids maximal des circuits de l'automate. En fait, il n'est pas difficile de remarquer que ρ est le poids maximal des circuits élémentaires de l'automate.
- δ est la différence minimale entre ρ et les poids des autres circuits élémentaires.
- $\mathcal{L}_\rho$ est l'ensemble des mots acceptés par $\mathcal{A}$ pour lesquels il existe un chemin qui emprunte un état du graphe critique.
- $\mathcal{L}_{\overline{\rho}}$ est le complément de $\mathcal{L}_\rho$ dans l'ensemble des mots acceptés par $\mathcal{A}$: $\mathcal{L}_{\overline{\rho}} = \mathcal{L}_{\mathcal{A}} \smallsetminus \mathcal{L}_\rho$.

Exemple 22.1 L'automate $\mathcal{A}_{\mathsf{m}_1}$ présenté figure 2 a cinq états ($n = 5$). Ses coefficients sont compris entre -2 et 2 ($M = 2$). Il comporte deux circuits élémentaires dont les poids respectifs sont 1 et 3/2 ($\rho = 3/2$, $\delta = 1/2$). Le graphe critique est formé des états s et t

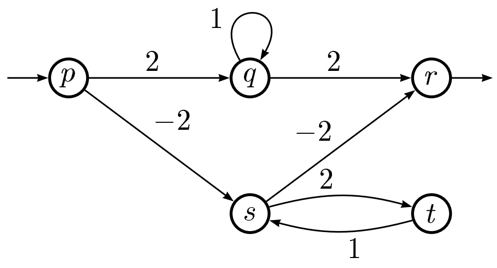


FIG. 2 – *L'automate* $\mathcal{A}_{\mathsf{m}_1}$.

ainsi que des deux transitions entre ces états. Les calculs de cet automate passant par s ou t sont ceux de longueur paire non nulle, donc $\mathcal{L}_\rho = \{2k \mid k > 0\}$. Comme il existe des calculs de n'importe quelle longueur supérieure à 2, $\mathcal{L}_{\overline{\rho}} = \{2k+1 \mid k > 0\}$.

Le théorème suivant donne un critère qui permettra de décider la séquentialité d'une série.

THÉORÈME 5.1 *Soit $\mathcal{A}$ un automate unaire à multiplicité dans un semi-anneau* (max ,+) *et ρ le poids maximal de ses circuits. Soit $\mathcal{L}_{\overline{\rho}}$ l'ensemble des mots acceptés par $\mathcal{A}$ qui n'étiquettent aucun chemin passant par un état du graphe critique. La série rationnelle réalisée par l'automate α est séquentielle si et seulement si $\mathcal{L}_{\overline{\rho}}$ est fini.*

Ce théorème est équivalent aux propositions 5.8 et 5.11 que nous allons prouver séparément.

Le résultat suivant montre que les coefficients d'une série rationnelle unaire sont bornés supérieurement par une progression arithmétique de raison ρ (poids maximal des circuits de l'automate qui réalise la série) et que certains coefficients restent à distance bornée de cette progression.

PROPOSITION 5.7 *Soit α une série rationnelle réalisée par un automate unaire $\mathcal{A}$. Soit n le nombre d'états de $\mathcal{A}$, ρ le poids maximum de ses circuits et M le maximum des valeurs absolues de ses coefficients.*
Alors, pour tout entier k de $\mathbb{N}$,

$$\langle \alpha, k\rangle \leqslant \rho k + M(n+2),$$
$$\exists (r,s) \in \mathbb{N}^2, \langle \alpha, rk+s\rangle \geqslant \rho(rk+s) - M(4n+2).$$

Démonstration. Montrons la première inégalité par récurrence sur k. Mais d'abord, remarquons que s'il n'existe pas de calcul de longueur k dans $\mathcal{A}$, alors $\langle \alpha, k\rangle = -\infty$ et l'inégalité est vérifiée. Si $k \leqslant n$, puisque tout calcul de longueur k a une valeur inférieure à $M(k+2)$, l'égalité est vraie. Pour tout $k > n$, considérons un calcul de valeur maximale de longueur k. Ce calcul comporte obligatoirement un circuit de longueur $c < k$ et de valeur au plus $c\rho$. Le reste du calcul (de longueur $k - c$) a, par hypothèse de récurrence, une valeur au plus égale à $\rho(k-c) + M(n+2)$; la valeur du calcul de longueur k est donc bien au plus $\rho k + M(n+2)$.

Construisons les entiers qui vérifient la seconde inégalité. Considérons un circuit de poids maximum; soit p un état de ce circuit et r sa longueur. Il existe deux chemins qui vont respectivement d'un état initial à p et de p à un état final et dont la longueur totale s est inférieure à $2n$. Leur valeur totale est donc supérieure à $-M(2n+2)$[4]. Pour tout entier k et pour tout mot de longueur $rk+s$, il existe donc un calcul de valeur supérieure à $\rho rk - M(2n+2)$. Cette quantité est elle-même supérieure à $\rho(rk+s) - M(4n+2)$. □

Proposition 5.8 *Avec les notations du théorème 5.1, si une série rationnelle α réalisée par un automate $\mathcal{A}$ est séquentielle, $\mathcal{L}_{\overline{\rho}}$ est fini.*

Démonstration. Soit s et r les entiers définis à la proposition 5.7. Si une série est séquentielle, elle est uniformément divergente, donc il existe un entier c tel que :

$$|i-j| \leqslant r \Rightarrow |\langle \alpha,i\rangle - \langle \alpha,j\rangle| \leqslant c.$$

Soit δ la différence entre ρ et les poids des autres circuits élémentaires. Pour tout l de $\mathcal{L}_{\overline{\rho}}$, $\langle \alpha,l\rangle \leqslant M(n+2) + (\rho-\delta)l$. Posons $k = \lceil (l-s)/r \rceil$. On a alors :

$$\begin{aligned}\langle \alpha,rk+s\rangle - \langle \alpha,l\rangle &> [\rho(rk+s) - M(4n+2)] - [M(n+2) + (\rho-\delta)l] \\ &> \delta l - M(5n+4).\end{aligned}$$

Si $\mathcal{L}_{\overline{\rho}}$ est infini, on peut prendre un élément l supérieur à $[M(5n+4)+c]/\delta$. En prenant k comme ci-dessus, on obtient que $|rk+s-l| < r$ et $|\langle \alpha,rk+s\rangle - \langle \alpha,l\rangle| > c$, ce qui contredit la divergence uniforme. □

On va maintenant prouver la proposition inverse. Pour cela, on utilise un petit lemme combinatoire :

Lemme 5.9 *Pour tout entier n de $\mathbb{N}$, pour tout k supérieur à n, quel que soit l'ensemble $(x_i)_{i\in[1;k]}$ d'éléments de $\mathbb{Z}/n\mathbb{Z}$, il existe un sous-ensemble J de $[1;k]$ tel que*

$$\sum_{i\in J} x_i = 0.$$

Démonstration. C'est en fait une application simple du lemme des tiroirs. Pour j appartenant à $[1;k]$, notons S_j la somme des j premiers x_i. Si l'une de ces sommes est nulle, on obtient le résultat, sinon, ces k sommes ne peuvent prendre que $n-1$ valeurs possibles; il en existe donc deux (S_i et S_j, avec $i<j$, par exemple) qui ont la même valeur. La somme des éléments de $i+1$ à j est donc nulle. □

Ce lemme s'applique aux mots dans un automate.

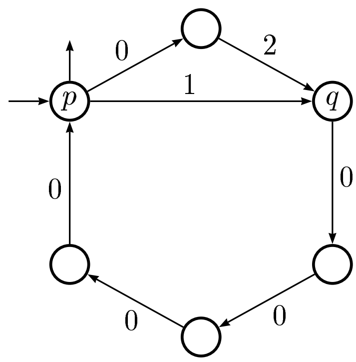
Lemme 5.10 *Soit $\mathcal{A}$ un automate unaire à n états. Pour tout k, s'il existe un calcul de longueur k qui passe par un état du graphe critique, il existe un calcul de longueur k qui contient des circuits de poids maximum dont la suppression donne un calcul de longueur inférieure à n^2.*

4. en prenant en compte les valeurs initiale et terminale.

Démonstration. Si k est inférieur à n^2, il vérifie le lemme. Supposons maintenant qu'il existe k strictement supérieur à n^2 qui contredise le lemme. On suppose k minimal : c'est l'entier le plus petit tel qu'il existe un calcul de longueur k qui passe par un état du graphe critique, mais tel qu'il n'existe aucun calcul qui contient un circuit de poids maximum (car en enlevant ce circuit, soit on obtiendrait un chemin de longueur inférieur à n^2, soit on contredirait la minimalité de k). Considérons un calcul $\mathcal{C}$ de longueur k passant par un état p qui appartient à un circuit de poids maximum de longueur l.

Comme k est supérieur à n^2, on peut découper le calcul $\mathcal{C}$ en n morceaux de longueurs n qui contiennent chacun au moins un circuit (lemme des tiroirs). D'après le lemme précédent, on peut choisir un certain nombre de ces circuits de façon à ce que la somme de leur longueur est un multiple de l. En supprimant ces circuits et en insérant suffisament de fois le circuit de poids maximal de longueur l autour de p, on obtient un chemin de longueur k qui contient un circuit de poids maximal et, en même temps, une contradiction. □

Exemple 23 La borne quadratique que l'on obtient dans cette proposition est proche d'une borne optimale. La famille d'automates suivante réalise presque cette borne :



Ici, $n = 6$. Il y a deux circuits élémentaires de poids respectifs $2/n$ et $1/(n-1)$. Les calculs de longueur kn^2+1 empruntent, pour tout k, au moins 5 fois la transition entre p et q, lorsqu'on a supprimé tous les circuits de longueur n, il reste donc au moins $n-1$ circuits de longueur $n-1$, soit $(n-1)^2$ transitions. On peut d'ailleurs remarquer qu'on peut emprunter de nombreuses fois les transitions du graphe critique sans parcourir de circuit de poids maximum.

Proposition 5.11 *Avec les notations du théorème 5.1, si une série rationnelle α réalisée par un automate $\mathcal{A}$ est telle que $\mathcal{L}_{\overline{\rho}}$ est fini, elle est séquentielle.*

Démonstration. On pose $N_0 = \mathsf{max}\,\mathcal{L}_{\overline{\rho}}$ et $N = \mathsf{max}(N_0, n^2 + 2M(n^2+2)/\delta))$. Soit B le ppcm des longueurs des circuits de poids maximum.

D'après la proposition 5.7, pour tout entier k strictement supérieur à N,

$$\langle \alpha, k\rangle > \rho(k - n^2) - M(n^2+2). \qquad (1)$$

On suppose qu'il existe un entier r supérieur à N tel qu'il existe un calcul victorieux de longueur k avec r transitions en dehors des circuits de poids maximum. On pose $s = k - r$.

Néanmoins, la partie du calcul hors des circuits de poids maximum contient elle aussi des circuits; la valeur de α en k est donc bornée :

$$\begin{aligned}\langle \alpha,k\rangle \leqslant & M(n+2)+(\rho-\delta)(r-n)+s\rho \\ < & M(n^2+2)+(\rho-\delta)(r-n^2)+s\rho \\ < & \rho(s+r-n^2)+\delta(n^2+2M(n^2+2)/\delta-r)-M(n^2+2) \\ < & \rho(k-n^2)-M(n^2+2)\end{aligned}$$

Ce qui est en contradiction avec l'inégalité (1). Donc, pour tout mot plus long que N, pour tout calcul victorieux, il y a moins de N transitions en dehors des circuits de poids maximum.

On montre maintenant que la série a une certaine périodicité, c'est-à-dire que, pour tout k supérieur à $N+Bn$, on a :

$$\langle \alpha,k+B\rangle = \langle \alpha,k\rangle + B\rho.$$

On montre cette égalité comme une double inégalité. D'une part, tout calcul victorieux de longueur k passe par un état p qui appartient à un circuit de poids maximum. En tournant suffisament autour de p, on obtient un calcul de longueur $k+B$ dont la valeur est $\langle \alpha,k\rangle + B\rho$.

Réciproquement, tout chemin victorieux de longueur $k+B$ contient des circuits de poids maximum de longueurs $l_1,..,l_k$ inférieures à n . On pose λ_i le nombre de circuits de longueurs l_i. Comme il y a au plus N transitions en dehors de ces circuits, on a :

$$\sum_i \lambda_i l_i \geqslant Bn,$$

Donc il existe i tel que $\lambda_i l_i \geqslant B$ et comme l_i divise B, il existe λ'_i tel que $\lambda'_i l_i = B$. En empruntant les circuits de longueur l_i seulement $\lambda_i - \lambda'_i$ fois, on obtient un calcul de longueur k dont la valeur est $\langle \alpha,k+B\rangle - B\rho$, ce qui montre l'inégalité dans l'autre sens. □

Proposition 5.12 *Une série α de $\mathbb{K}\mathrm{Rat}\,\mathbb{N}$ est séquentielle si, et seulement si, elle est uniformément divergente.*

Démonstration. Soit α une série uniformément divergente : Pour tout entier k de $\mathcal{L}_{\overline{\rho}}$,

$$\langle \alpha,k\rangle \leqslant M(n+2)+(\rho-\delta)k.$$

De la proposition 5.7, on obtient qu'il existe un entier d tel que, pour tout entier k de $\mathcal{L}_{\overline{\rho}}$, il existe un entier k' tel que $|k-k'| \leqslant d$ et

$$\langle \alpha,k'\rangle \geqslant \rho k' - M(4n+2).$$

Comme α est uniformément divergente, il existe η (dépendant uniquement de d) tel que :

$$\begin{aligned}\eta & \geqslant |\alpha,k'-\alpha,k| \\ & \geqslant \rho k' - M(4n+2) - M(n+2) - (\rho-\delta)k \\ & \geqslant \rho d - M(5n+4) + \delta k\end{aligned}$$

D'où

$$k \leqslant \frac{\eta + M(5n+4) - \rho d}{\delta}.$$

Donc $\mathcal{L}_{\overline{\rho}}$ est fini et α est séquentielle. □

Si l'automate est une composante fortement connexe, c'est-à-dire si la matrice de transition est *irréductible*, le langage $\mathcal{L}_{\overline{\rho}}$ est alors fini quels que soient les vecteurs initial et final. On se retrouve alors dans le cadre de l'application du théorème de « Perron-Frobenius $(\mathsf{max},+)$ » qui nous dit en effet que la matrice de transition a une seule valeur propre.

4 Algorithmes

4.1 Décidabilité

La décidabilité dépend uniquement de la finitude ou non du langage $\mathcal{L}_{\overline{\rho}}$. Il suffit donc d'identifier l'ensemble R des états appartenant à un circuit de poids maximum. On est ensuite ramené à un problème classique de théorie des automates.

Pour reconnaître $\mathcal{L}_{\overline{\rho}}$, on construit d'abord un automate qui reconnaît $\mathcal{L}_{\rho}$. On considère deux copies de l'automate $\mathcal{A}_0$ sous-jacent à l'automate à multiplicité $\mathcal{A}$.[5] La première copie conserve les états initiaux et les transitions, la seconde les états terminaux et les transitions. On ne peut passer de la première à la seconde que si l'on emprunte une transition qui permet de se rendre dans un état de R.

Formellement, la construction est la suivante. Soit $\mathcal{A}_0 = \langle Q,A,E,I,T \rangle$ l'automate sous-jacent de $\mathcal{A}$. On construit $\mathcal{A}_1 = \langle Q \times \{0;1\},A,F,J,U \rangle$ défini par :

$$\begin{aligned}
J =& \{(p,0) \mid p \in I\} \\
U =& \{(p,1) \mid p \in T\} \\
F =& \{((p,i),a,(q,i)) \mid i \in \{0;1\},(p,a,q) \in E\} \\
&\cup \{((p,0),a,(q,1)) \mid (p,a,q) \in E, q \in R\}.
\end{aligned}$$

On ne tient alors pas compte du mot vide, mais ce n'est pas nécessaire, puisque ceci ne change rien au caractère fini ou infini de $\mathcal{L}_{\overline{\rho}}$.

Après avoir émondé $\mathcal{A}_1$ qui reconnaît $\mathcal{L}_{\rho}$, on le déterminise et le complète, afin de calculer le complément de $\mathcal{L}_{\rho}$. On obtient l'automate $\mathcal{A}_2$ qui reconnaît $\overline{\mathcal{L}_{\rho}}$ Il suffit ensuite d'effectuer le produit de l'automate $\mathcal{A}_2$ par $\mathcal{A}_0$, pour obtenir un automate qui reconnaît $\mathcal{L}_{\overline{\rho}}$.

Cette construction est exponentielle en la taille de l'automate. L'étape de déterminisation a effectivement une complexité de $O(e^{\sqrt{n \ln n}})$ (M. Chrobak [15]). Toutefois, cet algorithme ne dépend aucunement de la taille des coefficients ou de leurs distances respectives.

Exemple 22.2 Soit α_{m_1} la série réalisée par $\mathcal{A}_{\mathsf{m}_1}$. On calcule dans un premier temps l'automate (booléen) qui accepte le langage correspondant aux calculs qui passent par des

5. Cette construction est celle de la preuve du « théorème des arcs colorés » qui montre que le langage des mots étiquetant les calculs qui empruntent un sous ensemble de transitions donné d'un automate est reconnaissable ([55]).

états du circuit de poids maximum. Le résultat est présenté figure 3. Seule la partie en traits pleins appartient à l'automate émondé. On doit maintenant calculer l'automate qui

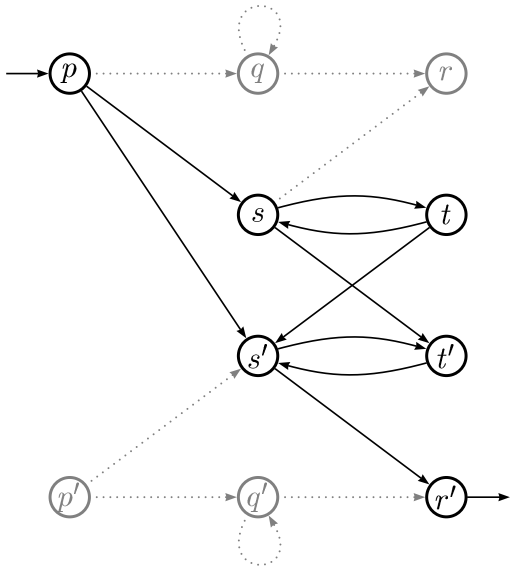


FIG. 3 – *L'automate qui reconnaît* $\mathcal{L}_\rho$.

accepte le complémentaire de ce langage. Pour cela, on déterminise l'automate puis on change les états non finals en états finals et inversement. On obtient l'automate représenté figure 4. Enfin, on calcule le produit avec l'automate sous-jacent de $\mathcal{A}_{\mathsf{m}_1}$ pour obtenir

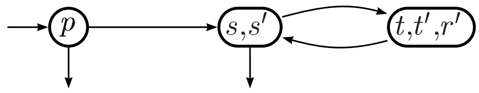


FIG. 4 – *L'automate minimal qui reconnaît* $A^* \setminus \mathcal{L}_\rho$.

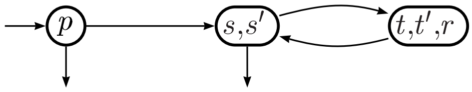
l'automate qui reconnaît $\mathcal{L}_{\overline{\rho}}$ (figure 5). L'automate produit émondé n'est pas acyclique, $\mathcal{L}_{\overline{\rho}}$ n'est donc pas fini et α_{m_1} n'est donc pas séquentielle.

4.2 Déterminisation

Nous décrivons ici un algorithme de déterminisation qui dérive de la construction des sous-ensembles et qui est similaire à l'algorithme de déterminisation des transducteurs. Nous allons voir cependant qu'il faut prendre quelques précautions.

Pour les automates sans multiplicité, à partir d'une représentation linéaire d'un automate, (λ,μ,ν), calculer l'automate déterminisé revient à calculer l'ensemble des $\lambda \otimes (u\mu)$, où u est un mot. Comme cet ensemble est inclus dans $\mathbb{B}^Q$, il est fini.

Lorsque le semi-anneau a une infinité d'éléments, l'ensemble $\lambda \otimes \mu^n$ n'est *a priori* pas fini, même si α est séquentielle, c'est le même phénomène qui fait que les quotients d'une série séquentielle sont en nombre infini. Pour contourner ce problème, on définit le



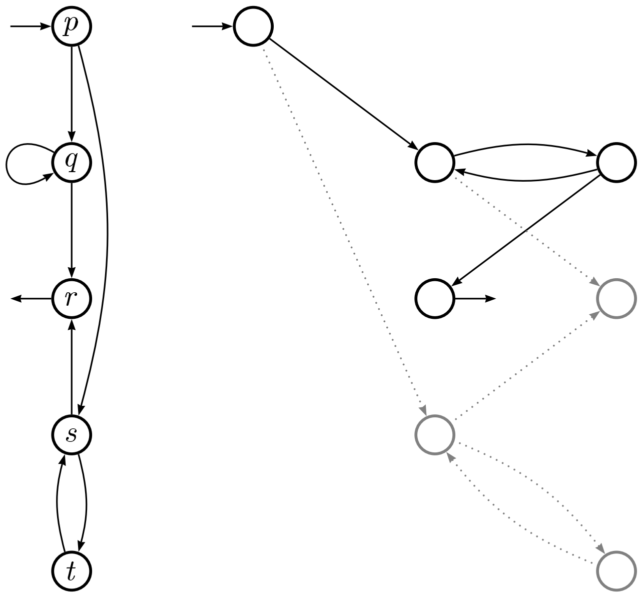


FIG. 5 – *L'automate qui reconnaît* $\mathcal{L}_{\overline{\rho}}$.

translaté d'un vecteur, de la même façon qu'on a défini la translatée d'une série.

DÉFINITION 5.6 *On suppose que Q est un ensemble fini ordonné. Pour tout vecteur γ de $\mathbb{K}_{\mathsf{m}}^{Q}$, on définit :*

$$\mathring{\gamma} = \begin{cases} \mathsf{min}\{\gamma_p \mid p \in \mathsf{Supp}(\gamma)\}, & \text{si } \mathbb{K} \text{ est positif,} \\ \mathsf{max}\{\gamma_p \mid p \in \mathsf{Supp}(\gamma)\}, & \text{si } \mathbb{K} \text{ est négatif,} \\ \gamma_p, \text{ avec } p = \mathsf{min}\,\mathsf{Supp}(\gamma), & \text{si } \mathbb{K} \text{ est un groupe.} \end{cases}$$

*Le **vecteur translaté** de γ est $\overline{\gamma} = \gamma - \mathring{\gamma}$.*

DÉFINITION 5.7 *Soit $\mathcal{A}$ un automate unaire dont les états forment l'ensemble Q et (λ,μ,ν) sa représentation linéaire. Soit $S = \{\overline{\lambda \otimes \mu^n} \mid n \in \mathbb{N}\}$. Soit $\mathcal{B}$ l'automate déterministe décrit par la représentation (ζ,η,ϑ), où ζ et ϑ sont des vecteurs de $\mathbb{K}_{\mathsf{m}}^{S}$ et η une matrice de $\mathbb{K}_{\mathsf{m}}^{S\times S}$ définis par :*

$$\zeta_p = \begin{cases} \mathring{\lambda} & \text{si } p = \overline{\lambda}, \\ -\infty & \text{sinon.} \end{cases}$$

$$\vartheta_p = \mathsf{max}\{p_i + \nu_i \mid i \in Q\},$$

$$\eta_{p,q} = \begin{cases} \mathring{q} & \text{si } q = \overline{p \otimes \mu}, \\ -\infty & \text{sinon.} \end{cases}$$

On peut montrer que cet automate réalise bien la même série que $\mathcal{A}$. Malheureusement, avec cette définition, l'ensemble Q peut ne pas être fini. Il faut donc modifier la définition de façon à obtenir un automate fini.

Pour cela, on réalise une partition de l'ensemble Q des états de l'automate $\mathcal{A}$ qu'on veut déterminiser. Soit R l'ensemble des états qui appartiennent à des circuits de poids maximum ρ. On scinde Q en deux parts :

- les éléments de R où leurs descendants : $\{p \mid \exists u \in a^*, p \in R \cdot u\}$;
- les autres éléments de Q qui forment un ensemble G.

Pour tout état p de la première catégorie, la fonction $n \mapsto (\lambda \otimes \mu^n)_p$ croît comme ρ^n, alors que les états de G prennent de plus en plus de « retard ». Si la fonction est séquentielle, tout chemin victorieux de $\mathcal{A}$ ne peut traverser les états de G que dans une première section de longueur bornée. On définit donc un filtre :

Définition 5.8 *Soit Q un ensemble fini et F un sous-ensemble de Q, pour tout vecteur γ de $\mathbb{K}_{\mathsf{m}}^Q$, on définit le vecteur $\gamma \setminus F$ de $\mathbb{K}_{\mathsf{m}}^Q$ par :*

$$(\gamma \setminus F)_p = \begin{cases} -\infty & \text{si } p \in F, \\ \gamma_p & \text{sinon.} \end{cases}$$

La définition de l'automate déterminisé est alors la suivante :

Définition 5.9 *Soit $\mathcal{A}$ un automate unaire dont les états forment l'ensemble Q de cardinal n et (λ,μ,ν) sa représentation linéaire. On calcule ρ et ρ_m les poids maximaux et minimaux des circuits élémentaires de $\mathcal{A}$, ainsi que M, valeur absolue maximale des coefficients de $\mathcal{A}$. On pose :*

$$\Delta = \rho(n^2 - 2n) - \rho_m(n^2 - n) + M(3n + 4).$$

On construit de façon incrémentale l'automate déterministe $\mathcal{D} = \langle R,\{a\},\mathbb{K},E,I,T \rangle$, où R est un sous-ensemble de $\mathbb{K}^Q$.

- *L'état initial est $\gamma_0 = \overline{\lambda}$ et $I_{\gamma_0} = \overset{\circ}{\lambda}$.*
- *Pour tout i, si i est inférieur à n^2 ou si $\mathsf{Supp}(\gamma_i \otimes \mu)$ est différent de $\mathsf{Supp}(\gamma_k)$ pour tout $k \leqslant i$, alors $\gamma_{i+1} = \overline{\gamma_i \otimes \mu}$ et $E_{(\gamma_i,a,\gamma_{i+1})} = \gamma_i \overset{\circ}{\otimes} \mu$. Sinon, on considère $\gamma = \gamma_i \otimes \mu$ et on calcule l'ensemble F de la façon suivante :*

$$F = \{p \in G \mid \mathsf{min}\{\gamma'_q - \gamma'_p \mid q \in Q \setminus G\} \geqslant \Delta\}$$

On pose alors $\gamma_{i+1} = \overline{\gamma \setminus F}$ et $E_{(\gamma_i,a,\gamma_{i+1})} = \overset{\circ}{\overbrace{\gamma \setminus F}}$.
- *Pour tout état γ_i, $T_{\gamma_i} = \gamma_i \otimes \nu$.*

Proposition 5.13 *Soit $\mathcal{A}$ un automate déterminisable. L'automate $\mathcal{D}$ défini dans la proposition précédente est un automate fini déterministe qui réalise la même série que $\mathcal{A}$.*

Démonstration. L'automate est fini car, ultimement, l'ensemble F est égal à l'ensemble G. Pour tout n supérieur à la borne N donnée dans la preuve de la proposition 5.11, les seules coordonnées non nulles du vecteur γ obtenu après lecture de a^n n'appartiennent pas à G, et, pour tout état p de $Q \setminus G$, pour tout k supérieur à $N + Bn$ (avec $n = \mathsf{Card}(Q)$), $(\lambda \otimes \mu^{k+B})_p = (\lambda \otimes \mu^k)_p + B\rho$. Donc l'automate contient au plus $N + Bn$ états.

Montrons qu'en supprimant des états comme nous le faisons, nous conservons les calculs victorieux qui correspondent à chaque mot. D'une part, comme l'automate est déterminisable, si on considère le déterminisé de l'automate sous-jacent, tout état final de la boucle du déterminisé contient un état de $Q \setminus G$. Supposons qu'on se trouve dans un état γ_i et que l'élément q de Q respecte les conditions pour être supprimé. Pour tout mot v dans le futur de q, tout chemin étiqueté par v qui va de q à un état final a pour valeur maximale $\rho(|v| - n) + (n + 1)M$. La valeur du calcul d'un mot $u.v$ (avec $|u| = i$) qui emprunterait q après la lecture de u est donc au plus $(\gamma_i)_q + \rho(|v| - n) + (n + 1)M$.

Tout élément p de $Q \setminus G$ qui appartient à γ_i satisfait $(\gamma_i)_p \leqslant \rho(i - n) + (n + 1)M$. Et nous avons vu qu'il existe un tel élément au futur duquel appartient v. D'autre part, comme i est par hypothèse supérieur à n^2, nous avons vu que le mot $u.v$ étiquette un chemin qui a au plus n^2 arcs en dehors des circuits de poids maximal. Sa valeur est donc au moins $\rho(|u.v| - n^2) + \rho_m(n^2 - n) - (n + 2)M$. On a donc :

$$\begin{aligned}
(\gamma_i)_q + \rho(|v| - n) + (n + 1)M &\leqslant \rho(|u| - n) + (n + 1)M - \Delta + \rho(|v| - n) + (n + 1)M \\
&\leqslant \rho(|u.v| - 2n) + (2n + 2)M - \Delta \\
&\leqslant \rho(|u.v| - 2n) + (2n + 2)M \\
&\qquad - (\rho(n^2 - 2n) - \rho_m(n^2 - n) + M(3n + 4)) \\
&\leqslant \rho(|u.v| - n^2) + \rho_m(n^2 - n) - (n + 2)M
\end{aligned}$$

Le calcul victorieux pour $u.v$ ne passe donc pas par q après la lecture de u. Le mot $u.v$ sera donc accepté avec la même valeur qu'on enlève ou pas l'état p.

L'automate $\mathcal{D}$ réalise donc la même série que $\mathcal{A}$. En effet, pour tout vecteur γ, on a $\overline{\gamma \otimes \mu} = \overline{\overline{\gamma} \otimes \mu}$, donc, si (ζ,η,ϑ) est la représentation linéaire de $\mathcal{D}$, on a :

$$\begin{aligned}
\forall k, \zeta \otimes \eta^k \otimes \vartheta =& \mathring{\lambda} + \overline{\lambda} \mathring{\otimes} \mu + \overbrace{\overline{\lambda \otimes \mu} \otimes \mu}^{\circ} + \ldots + \overbrace{\overline{\lambda \otimes \mu^{k-2}} \otimes \mu}^{\circ} \\
& + \overbrace{\overline{\lambda \otimes \mu^{k-1}} \otimes \mu}^{\circ} + \mathsf{max}\{\overline{\lambda \otimes \mu^k}_i + \nu_i \mid i \in Q\} \\
=& \mathsf{max}\{\mathring{\lambda} + \lambda \mathring{\otimes} \mu + \ldots + \overbrace{\overline{\lambda \otimes \mu^{k-2}} \otimes \mu}^{\circ} \\
& + \overbrace{\overline{\lambda \otimes \mu^{k-1}} \otimes \mu}^{\circ} + (\overline{\overline{\lambda \otimes \mu^{k-1}} \otimes \mu})_i + \nu_i \mid i \in Q\} \\
=& \mathsf{max}\{\mathring{\lambda} + \lambda \mathring{\otimes} \mu + \ldots + \overbrace{\overline{\lambda \otimes \mu^{k-2}} \otimes \mu}^{\circ} + (\overline{\lambda \otimes \mu^{k-1}} \otimes \mu)_i + \nu_i \mid i \in Q\}
\end{aligned}$$

$$
\begin{aligned}
&= \max\{\mathring{\lambda} + \lambda \mathring{\otimes} \mu + \ldots + \overbrace{\lambda \otimes \mu^{k-2} \otimes \mu}^{\circ} + (\overline{\lambda \otimes \mu^{k-1}})_i \\
&\qquad + (\mu \otimes \nu)_i \mid i \in Q\} \\
&= \max\{\mathring{\lambda} + \overline{\lambda}_i + (\mu^k \otimes \nu)_i \mid i \in Q\} \\
&= \max\{\lambda_i + (\mu^k \otimes \nu)_i \mid i \in Q\} \\
&= \lambda \otimes \mu^k \otimes \nu
\end{aligned}
$$

□

REMARQUE 5.3 Si $\mathbb{K}$ est positif ou négatif, on peut donner pour Δ une valeur plus petite :

$$
\begin{aligned}
\mathbb{K} \geqslant 0,\ \Delta =& \rho(n^2 - 2n) - \rho_m(n^2 - n) + M(2n+2), \\
\mathbb{K} \leqslant 0,\ \Delta =& \rho(n^2 - 2n) - \rho_m(n^2 - n) + M(n+2).
\end{aligned}
$$

D'autre part, chaque fois qu'on a une raison pour décider qu'un état a pris trop de retard, on peut le supprimer, ce qui peut, dans bien des cas, diminuer la taille de l'automate obtenu.

5 Non-ambiguïté des séries rationnelles sur un alphabet à une lettre

Nous avons signalé qu'on peut décider de la séquentialité d'une série réalisée par un automate non ambigu. De fait, pour les alphabets unaires (le monoïde libre est alors isomorphe à $\mathbb{N}$), on a la propriété suivante, due à S. Gaubert [24] :

PROPOSITION 5.14 *Toute série de* $\mathbb{K}_{\mathsf{m}}\mathrm{Rat}\,\mathbb{N}$ *peut être réalisée par un automate non ambigu.*

La preuve de cette proposition consiste à montrer que l'ensemble des séries qui sont des sommes finies disjointes de séries (séquentielles) de la forme :

$$
\langle \alpha, l \rangle = \begin{cases} x + ky & \text{si } \exists r,s,\ l = rk + s \\ -\infty & \text{sinon.} \end{cases}
$$

est fermé sous les opérations rationnelles, c'est-à-dire qu'on peut appliquer la somme, le produit ou l'étoile de façon non ambiguë à de telles séries. Cette méthode pour obtenir une représentation non ambigué risque en pratique de mener à une explosion combinatoire.

Nous donnons ici une méthode de construction d'un automate non ambigu pour une série rationnelle de $\mathbb{K}_{\mathsf{m}}\mathrm{Rat}\,\mathbb{N}$ quelconque, inspirée par l'algorithme de déterminisation.

DÉFINITION 5.10 *Soit* $\mathcal{A} = \langle\, Q,A,\mathbb{K}_m,E,I,T\,\rangle$ *un automate avec multiplicité et* $\mathcal{B} = \langle\, R,A,F,J,U\,\rangle$ *un automate sans multiplicité sur le même alphabet. Le* ***produit avec***

multiplicité *de $\mathcal{A}$ par $\mathcal{B}$ est l'automate correspondant à la représentation linéaire* (λ,μ,ν) *sur $Q \times R$, avec :*

$$\forall (i,j) \in Q \times R,\ \lambda_{i,j} = \begin{cases} I_i & \text{si } j \in J, \\ -\infty & \text{sinon;} \end{cases}$$

$$\nu_{i,j} = \begin{cases} T_i & \text{si } j \in U, \\ -\infty & \text{sinon;} \end{cases}$$

$$\forall a \in A, \forall (i,j),(i',j') \in Q \times R,\ (a\mu)_{(i,j),(i',j')} = \begin{cases} E_{(i,a,i')} & \text{si } (j,a,j') \in F, \\ -\infty & \text{sinon;} \end{cases}$$

Cet outil nous permet d'énoncer l'algorithme.

Soit α une série rationnelle réalisée par un automate émondé $\mathcal{A}$.

i) Si $\mathcal{A}$ est acyclique, le langage est fini et on peut facilement donner un automate non ambigu équivalent.

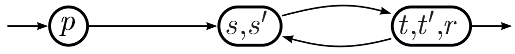
ii) Sinon, soit $\mathcal{L}_\rho$ l'ensemble des mots acceptés par $\mathcal{A}$ pour lesquels il existe un chemin qui emprunte un état d'un circuit de poids maximum ρ. On calcule le produit avec multiplicité de $\mathcal{A}$ par l'automate qui reconnaît $\mathcal{L}_\rho$. On obtient un automate qui réalise une série séquentielle qui coïncide avec α sur son support et qu'on peut déterminiser afin d'obtenir un automate non ambigu.

iii) On calcule à présent le produit avec multiplicité (émondé) de $\mathcal{A}$ avec l'automate qui reconnaît le complémentaire de $\mathcal{L}_\rho$ dans A^*. On obtient un automate $\mathcal{A}'$ qui réalise une série dont le support est disjoint de $\mathcal{L}_\rho$ et qui coïncide sur son support avec α. On applique l'algorithme à $\mathcal{A}'$.

Cet algorithme termine car l'ensemble des poids des circuits de $\mathcal{A}'$ est inclus dans celui des poids des circuits de $\mathcal{A}$, et cette inclusion est stricte, puisque qu'aucun circuit de $\mathcal{A}'$ n'a un poids égal ρ. Cet ensemble décroît donc jusqu'à ce qu'on obtienne un automate déterminisable, ce qui marque la fin de l'algorithme.

Il faut souligner que, contrairement au cas des transducteurs fonctionnels, la taille minimal d'un automate non ambigu équivalent à un automate $\mathcal{A}$ ne dépend pas uniquement de la structure de l'automate sous-jacent à $\mathcal{A}$, mais aussi des coefficients.

Exemple 22.3 Dans un premier temps, comme on l'a fait dans l'exemple 22.2, on calcule un automate qui reconnaît $\mathcal{L}_\rho$. En pratique, il peut être préférable de prendre l'automate minimal qui reconnaît ce langage. On calcule l'automate produit comme indiqué sur la figure 6. On calcule ensuite le déterminisé de l'automate obtenu. Dans ce cas précis, on sait quand on peut « abandonner » les états q ou q'. Le résultat est présenté figure 7. Ensuite on examine l'automate réalisant α_{m_1} sur le complémentaire de $\mathcal{L}_\rho$. Il est obtenu comme indiqué sur la figure 8. Comme cet automate n'a qu'un circuit, il réalise une série séquentielle et, de plus, aucun état ne peut prendre de retard important; on peut donc procéder à une déterminisation « classique » présentée figure 9. L'union des automates des figures 7 et 9 est un automate non ambigu qui réalise α_{m_1}.



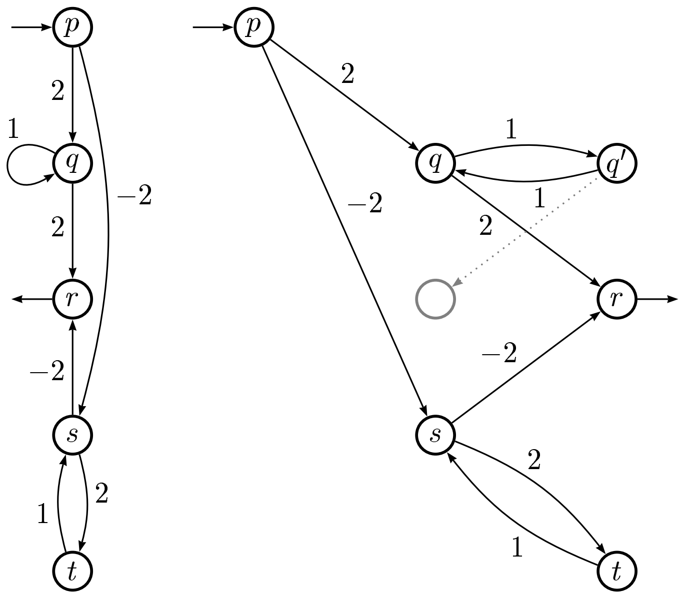


FIG. 6 – *Un automate qui réalise* α_{m_1} *sur* $\mathcal{L}_\rho$.

Calculons la taille minimale d'un automate non ambigu qui réalise α_{m_1}. Cette série vérifie :

$$\forall k > 0,\ \langle \alpha,2k\rangle = \mathsf{max}(3k-7,2k+2) \qquad \langle \alpha,2k+1\rangle = 2k+3.$$

Considérons un automate non ambigu qui réalise cette série. Examinons le calcul réussi de longueur 16. Sa valeur est 18. S'il existe un circuit de longueur l auquel appartient un des dix-sept états de ce calcul, (on peut supposer l pair, quitte à emprunter plusieurs fois le circuit), il existe un calcul de longueur $16 + l$ dont la valeur est 18 plus la valeur M_l de ce circuit. En empruntant deux fois le circuit, on obtient un calcul de longueur $16 + 2l$ dont la valeur est $18 + 2M_l$, et comme la différence entre les coefficients de deux entiers pairs supérieurs à 18 qui diffèrent de l dans α est $3l/2$, on a $M_l = 3l/2$ et la valeur du calcul de longueur $16 + l$ est $18 + 3l/2$, alors que $\langle \alpha_{\mathsf{m}_1},16 + l\rangle = 17 + 3l/2$. Donc aucun des dix-sept états du calcul réussi de longueur 16 n'appartient à un circuit.

D'autre part, les valeurs des chemins de longueur paire et impaire ne croissent pas de la même façon, il existe donc au moins deux circuits disjoints de longueur au moins 2, soit quatre états de plus. Le plus petit automate non ambigu qui réalise α_{m_1} a donc au moins vingt-et-un états. Cette borne est atteinte. Le plus petit automate non ambigu qui réalise α_{m_1} est représenté figure 10. Il est assez facile de voir qu'en changeant seulement les coefficients de $\mathcal{A}_{\mathsf{m}_1}$, on peut réaliser des séries pour lesquelles la taille du plus petit automate non ambigu est arbitrairement grande.

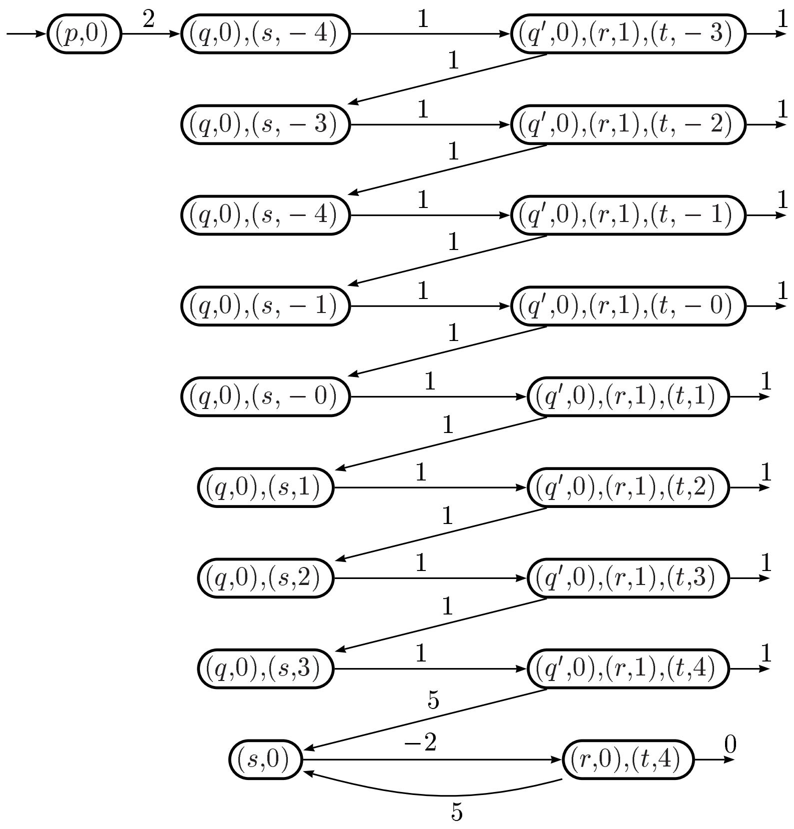


FIG. 7 – *Un automate déterministe qui réalise* α_{m_1} *sur* $\mathcal{L}_\rho$.

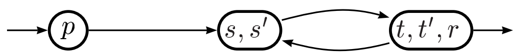
6 Automates univoques

Si l'alphabet a plusieurs lettres, le problème de la séquentialité et de la déterminisation reste ouvert. Toutefois, si l'opérateur max n'est pas utilisé, c'est-à-dire si chaque chemin réussi de l'automate considéré, étiqueté par un mot donné, a la même valeur on peut apporter une réponse. Ce cas correspond à celui des transducteurs fonctionnels, et nous allons voir qu'on peut étendre les techniques développées pour ceux-ci.

DÉFINITION 5.11 *Soit $\mathcal{A}$ un automate à multiplicité sur $\mathbb{K}$. On dit que $\mathcal{A}$ est* ***univoque*** *si, pour tout mot u, tous les chemins réussis de $\mathcal{A}$ étiquetés par u a la même valeur. Autrement dit, tout calcul est victorieux.*

REMARQUE 5.4 Les automates non ambigus sont évidemment univoques. D'ailleurs, on peut construire un automate non ambigu équivalent à un automate univoque donné[6]. De plus, la série réalisée par un tel automate est rationnelle non seulement sur $\mathbb{K}_{\mathsf{m}}$, mais aussi sur $\mathbb{K}_m$.

6. Cette construction est similaire à celle qu'on peut effectuer sur les transducteurs fonctionnels, mais nous allons voir qu'elle n'est pas nécessaire pour décider de la séquentialité.



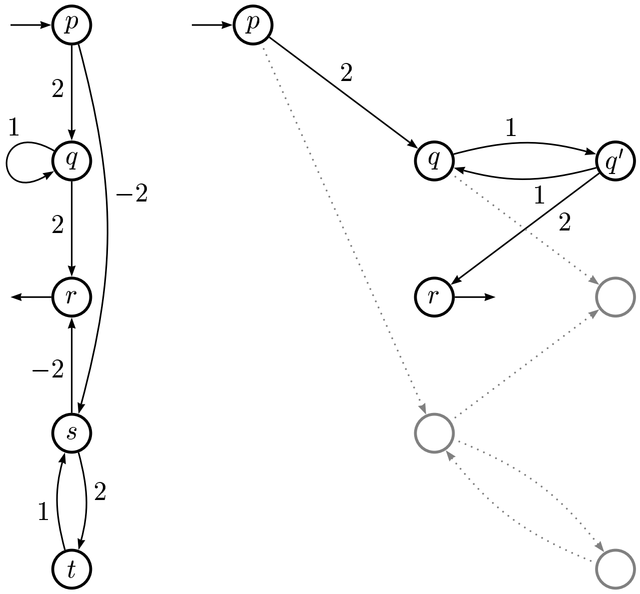


FIG. 8 – *Un automate qui réalise* α_{m_1} *sur* $\mathcal{L}_{\overline{\rho}}$*.*

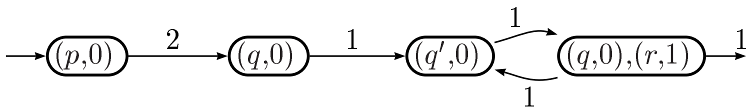


FIG. 9 – *Un automate déterministe qui réalise* α_{m_1} *sur* $\mathcal{L}_{\overline{\rho}}$*.*

Nous reprenons ici les algorithmes présentés en [5] pour les transducteurs, adaptés dans le cadre des automates (max ,+).

PROPOSITION 5.15 *On peut décider si un automate* (max ,+) *est univoque.*

Démonstration. Soit $\mathcal{A} = (Q,A,\mathbb{K},E,I,T)$ un automate et $\mathbb{G}$ le sous-groupe additif de $\mathbb{R}$ à partir duquel est construit $\mathbb{K}$. On construit l'automate

$$\mathcal{B} = \langle\, Q \times Q, A, (\mathcal{P}(\mathbb{G}), \cup, +), F, J, U\,\rangle$$

dont l'automate sous-jacent est le carré de l'automate sous-jacent de $\mathcal{A}$ et tel que :

$$\begin{aligned} J_{(p,q)} &= I_p - I_q \\ U_{(p,q)} &= T_p - T_q \\ F_{((p,q),a,(p',q'))} &= E_{(p,a,p')} - E_{(q,a,q')} \end{aligned}$$

Les valeurs de cet automate représentent l'avance (ou le retard) que l'on prend en suivant l'un ou l'autre des deux chemins de $\mathcal{A}$ qui correspondent à un chemin de $\mathcal{B}$. Soit (λ,μ,ν)

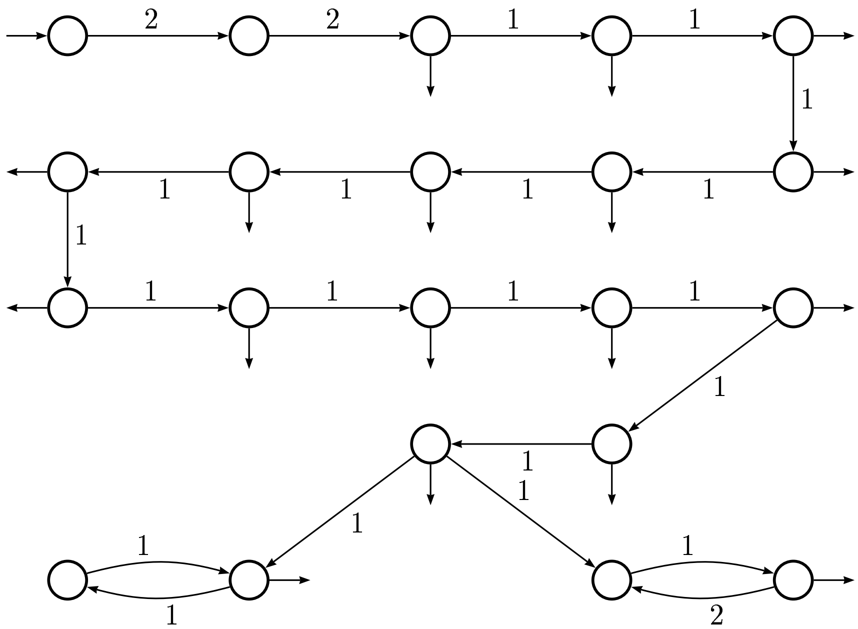


FIG. 10 – *Un automate non ambigu pour* α_{m_1}.

la représentation linéaire de $\mathcal{B}$. S'il existe un mot u tel que $\lambda \otimes u\mu \otimes \nu$ est différent de $\{0\}$, alors il existe un chemin réussi dans $\mathcal{B}$ étiqueté par u dont la valeur est non nulle. Les deux chemins réussis de $\mathcal{A}$ qui correspondent à ce chemin ont donc une valeur différente et $\mathcal{A}$ n'est pas univoque. Réciproquement, si $\lambda \otimes u\mu \otimes \nu = \{0\}$ pour tout mot u, tous les chemins étiquetés par un même mot ont la même valeur et l'automate $\mathcal{A}$ est univoque.

Comment vérifier que cette condition est satisfaite ? Soit (p,q) un état de $\mathcal{B}$ et u et v deux mots du passé de (p,q). Si $(\lambda \otimes u\mu)_{(p,q)} \neq (\lambda \otimes v\mu)_{(p,q)}$, alors, pour tout mot w dans le futur de (p,q), $\lambda \otimes (u.w) \otimes \mu\nu$ ou $\lambda \otimes (v.w)\mu \otimes \nu$ est différent de $\{0\}$. Si $\mathcal{A}$ est univoque, on peut donc associer à tout état (p,q) la valeur $(p,q)\chi = (\lambda \otimes u\mu)_{(p,q)}$, pour u dans le passé de (p,q). La première partie de l'algorithme consiste donc en un parcours de l'automate à partir des états initiaux (pour lesquels $(p,q)\chi = J_{(p,q)}$) afin d'attribuer à chaque état sa valeur. S'il existe une contradiction, l'automate $\mathcal{A}$ n'est pas univoque.

La seconde partie de l'algorithme consiste à vérifier que pour tout état final (p,q), $(p,q)\chi + T_{(p,q)} = 0$. L'automate $\mathcal{A}$ est alors univoque si et seulement si cette condition est satisfaite. □

On retrouve dans ce cadre les caractérisations habituelles :

PROPOSITION 5.16 *Soit α une série réalisée par un automate univoque. Alors α est séquentielle si et seulement si elle est à variations bornées.*

LEMME 5.17 *Soit $\mathcal{A}$ un automate avec n états. Soit $\mathcal{C}_1$ et $\mathcal{C}_2$ deux chemins étiquetés par le même mot u de longueur supérieure à n^2. Alors il existe une factorisation de u en $x.v.y$ telle que la portion de chemin étiquetée par v est une boucle dans $\mathcal{C}_1$ et $\mathcal{C}_2$.*

Démonstration. Il suffit de considérer l'automate $\mathcal{B}$, carré de $\mathcal{A}$. Il existe un chemin qui est le produit de $\mathcal{C}_1$ et de $\mathcal{C}_2$. Comme $\mathcal{B}$ a seulement n^2 états, ce chemin comporte une boucle qui correspond à une boucle dans $\mathcal{C}_1$ et une boucle dans $\mathcal{C}_2$. $\square$

Démonstration de la proposition. Soit $\mathcal{A}$ un automate univoque et émondé et (λ,μ,ν) la représentation correspondante. On pose n le nombre d'états de l'automate. On suppose que la série réalisé par $\mathcal{A}$ est uniformément bornée. Il existe donc k tel que :

$$\forall u,v \in A^*,\ d_{\mathsf{pref}}(u,v) \leqslant 2n \Longrightarrow |\langle\alpha,u\rangle - \langle\alpha,v\rangle| \leqslant k$$

On veut montrer que α est séquentielle, ce qui revient à montrer qu'elle a un nombre fini de translatées. Remarquons que si, pour deux mots u et v,

$$\overline{\lambda \otimes u\mu} = \overline{\lambda \otimes v\mu},$$

alors les translatées de α selon u et v sont égales. On va donc montrer que si la série est uniformément divergente et réalisée de façon univoque, l'ensemble $\{\overline{\lambda \otimes u\mu} \mid u \in A^*\}$ est fini.

Pour tout couple (p,q) de Q^2, on considère l'ensemble $C_{p,q}$ des mots u tels que $(\overline{\lambda \otimes u\mu})_p$ et $(\overline{\lambda \otimes u\mu})_q$ sont différents de $-\infty$. Pour tout u appartenant à $C_{p,q}$ de longueur supérieure à n^2, on considère un chemin $\mathcal{C}_p$ (*resp.* $\mathcal{C}_q$) d'un état initial i_1 à l'état p (*resp.* de i_2 initial à q) étiqueté par u. Il existe une factorisation de u en $x.v.y$ telle que v étiquette une boucle dans chacun des chemins $\mathcal{C}_p$ et $\mathcal{C}_q$. Si ces deux boucles ont une valeur qui diffère de $r \neq 0$, en prenant h suffisament grand, on obtient que $(\overline{\lambda \otimes (x.v^h.y)\mu})_p$ et $(\overline{\lambda \otimes (x.v^h.y)\mu})_q$ diffèrent de plus de $k + (2n+2)M$, ce qui est en contradiction avec l'hypothèse de divergence uniforme. Donc les deux boucles ont même valeur et $(\overline{\lambda \otimes u\mu})_p - (\overline{\lambda \otimes u\mu})_q = (\overline{\lambda \otimes (x.y)\mu})_p - (\overline{\lambda \otimes (x.y)\mu})_q$. Donc pour tout u de $C_{p,q}$, il existe v dans $C_{p,q}$ de longueur inférieure à n^2 telle que la différence entre les coordonnées p et q est la même dans $\overline{\lambda \otimes u\mu}$ et $\overline{\lambda \otimes v\mu}$. Il y a donc un nombre fini de différences possibles entre ces deux coordonnées. Ceci est vrai pour tout couple; et comme, de plus, une des coordonnée de $\overline{\lambda \otimes u\mu}$ est nulle pour tout u, l'ensemble $\{\overline{\lambda \otimes u\mu} \mid u \in A^*\}$ est fini. $\square$

L'argument employé dans cette preuve n'est ni plus ni moins que la « twinning property » (*cf.* [14]). C'est elle que l'on va tester pour décider la séquentialité.

PROPOSITION 5.18 *Soit $\mathcal{A}$ un automate univoque. On peut décider si $\mathcal{A}$ est déterminisable.*

Démonstration. Pour décider si un automate émondé $\mathcal{A}$ est déterminisable, on considère à nouveau l'automate $\mathcal{B}$ défini dans la preuve de la proposition 5.15. Cette fois, en revanche, on ne considère plus la partie émondée, mais la partie accessible de $\mathcal{B}$. On teste que la valeur de chaque circuit élémentaire de $\mathcal{B}$ est nulle. La fonction réalisée par $\mathcal{A}$ est séquentielle si et seulement si cette condition est satisfaite. Noter que sur la partie émondée de l'automate $\mathcal{B}$, cette condition est une conséquence du caractère univoque de $\mathcal{A}$.

Supposons qu'un circuit de $\mathcal{B}$ a une valeur non nulle. Soit (p,q) un état de ce circuit, u un mot du passé de (p,q) et v un mot qui étiquette le circuit (de valeur non nulle) autour de p.

Alors, si (λ,μ,ν) est la représentation linéaire de $\mathcal{A}$, pour k suffisament grand, la différence entre $(\lambda \otimes (u.v^k)\mu)_p$ et $(\lambda \otimes (u.v^k)\mu)_q$ peut être arbitrairement grande, ce qui, on l'a vu, contredit la divergence uniforme et donc la séquentialité de la série réalisée par $\mathcal{A}$.

Réciproquement, si tous les circuits de $\mathcal{B}$ ont une valeur nulle, quel que soit le mot u, si la longueur de u est supérieure à n^2 et que les coordonnés p et q de $\lambda \otimes u\mu$ sont non nulles, il existe un chemin d'un état initial de $\mathcal{B}$ à (p,q). Ce chemin comprend une boucle étiquetée par v (on pose x et y tels que $u = x.v.y$). Alors la distance entre les coordonnées p et q de $\lambda \otimes u\mu$ est la même qu'entre les coordonnées p et q de $\lambda \otimes (x.y)\mu$. La distance entre les coordonnées des vecteurs $\lambda \otimes u\mu$, pour u dans A^* est donc bornées par celle des coordonnées des vecteurs correspondant aux mots de longueur inférieure à n^2; la série réalisée est donc uniformément divergente et donc séquentielle. □

La déterminisation se fait alors tout simplement en calculant les vecteurs $\{\overline{\lambda \otimes u\mu} \mid u \in A^*\}$, de même que la construction pour les automates sans multiplicité.

PROPOSITION 5.19 *Soit $\mathcal{A}$ un automate univoque déterminisable et (λ,μ,ν) sa représentation linéaire. Soit $\mathcal{D}$ l'automate, dont les états sont $\{\overline{\lambda \otimes u\mu} \mid u \in A^*\}$, et qui est défini par sa représentation linéaire (ζ,η,ϑ) :*

$$\begin{aligned}
\zeta_p &= \begin{cases} \overset{\circ}{\lambda} & \text{si } p = \overline{\lambda}, \\ -\infty & \text{sinon,} \end{cases} \\
\vartheta_p &= p \otimes \nu, \\
\forall a \in A,\ (a\eta)_{p,q} &= \begin{cases} p \overset{\circ}{\otimes} a\mu & \text{si } q = \overline{p \otimes a\mu}, \\ -\infty & \text{sinon.} \end{cases}
\end{aligned}$$

Alors, l'automate $\mathcal{D}$ est un automate fini déterministe équivalent à $\mathcal{A}$. Il s'agit du ***déterminisé*** *de $\mathcal{A}$.*

Cet automate est clairement déterministe. Il est fini (on a montré que l'ensemble $\{\overline{\lambda \otimes u\mu} \mid u \in A^*\}$ est fini). Il réalise la même série que $\mathcal{A}$; on peut le montrer en suivant exactement la même preuve que celle de la proposition 5.13.

— ∘ —

7 Le cas général

Le cas général paraît beaucoup plus difficile à traiter. S. Gaubert a montré[7] que décider la séquentialité permet de répondre au problème de la limitation. Nous présentons ici ce problème ainsi que le résultat de S. Gaubert sans donner de preuve.

Soit α une série rationnelle de $\mathbb{K}_{\mathsf{m}}\mathrm{Rat}\,A^*$ donnée par un automate ou une représentation linéaire. On veut savoir si la série α est **limitée**, c'est-à-dire si l'ensemble $\{\langle\alpha,u\rangle \mid u \in A^*\}$ est fini[8].

7. Communication personnelle basée sur des résultats de [25].

8. On peut montrer (*cf.* [25]) que cette condition est équivalente à dire que les coefficients de α sont bornés.

Ce problème difficile (qui intervient dans l'approche de K. Hashiguchi du problème de la hauteur d'étoile) a été résolu sur $\mathbb{N}_m$ par K. Hashiguchi [32, 34] et étudié par I. Simon [58, 60] et H. Leung [41]. Il a été étendu à $\mathbb{R}_{\mathsf{m}}$ par S. Gaubert [25].

PROPOSITION 5.20 *Soit α une série rationnelle de $\mathbb{K}_{\mathsf{m}}\mathrm{Rat}\,A^*$. Soit b une lettre n'appartenant pas à A on définit la série β de $\mathbb{K}_{\mathsf{m}}\mathrm{Rat}\,(A \cup \{b\})^*$ par :*

$$\begin{aligned} \forall u \in A^*, \langle \beta,u\rangle &= \langle \alpha,u\rangle \\ \langle \beta,u.b\rangle &= 0 \\ \forall u \notin A^*(1+b), \langle \beta,u\rangle &= -\infty \end{aligned}$$

α est limitée si et seulement si β est séquentielle.

Ceci montre que décider la séquentialité est « plus difficile » que décider la limitation.

— ◦ —

8 Problème de la généralisation à d'autres semi-anneaux

Tout ce que nous avons fait dans ce chapitre s'appuie sur la notion de translatée. Celle-ci n'est fondée que si le semi-anneau considéré est principal. Par exemple, on ne peut pas étendre cette notion à $(\mathcal{P}(B^*), \cup, .)$. Le problème de la séquentialisation des transducteurs (même sur un alphabet unaire en entrée) reste posé pour les transducteurs non fonctionnels.

— ◦ —

Chapitre 6

Dérivation d'expressions rationnelles avec multiplicité

« C'est à la fois très simple et très compliqué ... »
Hergé, *Tintin au pays de l'or noir*

Dans le quatrième chapitre, au détour de la preuve du théorème d'Eggan, nous avons donné un algorithme qui permet de transformer une expression rationnelle en automate. Cet algorithme n'est pas des plus efficaces : le nombre d'états de l'automate engendré double chaque fois que l'on rencontre une étoile.

Pour une expression donnée, on peut construire des automates beaucoup plus petits. Si n est le nombre de lettres qui apparaissent dans l'expression, on peut construire un automate avec $n + 1$ états qui reconnaît le langage. (Cette borne est optimale, il suffit de considérer une expression qui représente un mot de longueur n). On peut noter que ce n'est pas une borne optimale sur la taille de l'automate, puisque celui-ci peut alors avoir un nombre quadratique de transitions. J. Hromkovič, S. Seibert et T. Wilke [37] ont montré qu'on peut construire un automate avec seulement $n \log^2(n)$ transitions (voir aussi [28]).

Une première méthode pour construire un automate à $n + 1$ états consiste à utiliser un algorithme dit « de position » dans lequel chaque état correspond à une occurence de lettre dans l'expression. En comptant un état initial qui ne correspond à aucune lettre, on obtient ainsi un automate qui a exactement $n + 1$ états. Cette méthode a été initiée par Glushkov [27]; un algorithme efficace a été donné par Berry et Sethi [6]. Une preuve formelle de cet algorithme ainsi que ses relations avec les langages *locaux* ont été présentés par Berstel et Pin [7].

Une autre méthode est issue d'une approche algébrique. On a vu dans le chapitre introductif que l'ensemble des quotients à gauche d'un ensemble rationnel de A^* est fini : ce sont les états de l'automate minimal. Peut-on définir formellement une opération de *dérivation* sur les expressions rationnelles telle que la dérivation d'une expression E par une lettre soit une expression qui dénote le quotient du langage que représente E par la lettre ? Ceci a été fait par Brzozowski [9]. Il montre, de plus, que l'ensemble des dérivées d'une expression par tous les mots de A^* est fini, sous certaines conditions d'équivalences entre expressions (associativité, commutativité, idempotence). On peut donc construire un automate déterministe fini sur le modèle de l'automate minimal, qui reconnaît le langage. Ce ne sont plus les quotients qui étiquettent les états de l'automate, mais les dérivées de l'expression. Évidemment, on peut obtenir deux expressions différentes qui représentent le même langage, cet automate n'est donc pas nécessairement minimal. D'autre part, comme il s'agit d'un automate déterministe, il n'est pas étonnant que le nombre de ses états soit exponentiel par rapport à la taille de l'expression rationnelle qui, elle, peut être non-déterministe.

Antimirov [3] a repris l'idée de Brzozowski. Si $\mathcal{A}$ est un automate et $\mathcal{D}$ le déterminisé de $\mathcal{A}$, pour tout état X de $\mathcal{D}$, le futur de X est égal à l'union des futurs des éléments de X dans $\mathcal{A}$. Or, les futurs des états de $\mathcal{D}$ sont les quotients du langage. Les quotients du langages appartiennent donc à l'ensemble obtenu par clôture selon l'union, à partir de l'ensemble des futurs états de $\mathcal{A}$. Si on a l'esprit retors, et qu'on identifie l'union avec une somme booléenne, on peut dire que les quotients appartiennent au $\mathbb{B}$-semi-module engendré par les futurs de $\mathcal{A}$. L'algorithme d'Antimirov consiste, non plus à trouver des expressions qui représentent les quotients, mais des expressions qui représentent des générateurs de ce $\mathbb{B}$-semi-module. Il obtient ainsi un automate (non déterministe) qui a au plus $n+1$ états.

Les relations entre l'automate des positions donné par la première méthode et l'automate des dérivées d'Antimirov ont été étudiées par Champarnaud et Ziadi [13].

Les séries rationnelles sur un semi-anneau $\mathbb{K}$ peuvent elles aussi être dénotées par des expressions rationnelles avec des coefficients pris dans $\mathbb{K}$. Il paraît alors naturel d'adapter les algorithmes connus pour obtenir une méthode permettant de construire un automate avec multiplicité à partir d'une expression avec multiplicité. Caron et Flouret [11] ont ainsi donné un algorithme de position qui permet de construire un automate à $n+1$ états pour une expression comptant n lettres.

Généraliser le résultat de Brzozowski semble difficile. En effet, on sait que les quotients d'une série rationnelle peuvent être en nombre infini (considérer par exemple la série $a^* + (2a)^*$ sur $\mathbb{N}$, dont les quotients sont $\{a^* + 2^n(2a)^* \mid n \in \mathbb{N}\}$ qui est un ensemble infini de séries). En revanche, les quotients d'une série $\mathbb{K}$-rationnelle appartiennent à un $\mathbb{K}$-semi-module finiment engendré. Nous allons donc adapter la méthode d'Antimirov, de façon à calculer, à partir d'une expression rationnelle avec multiplicité, des expressions qui représentent des générateurs du $\mathbb{K}$-semi-module auquel appartiennent les quotients de la série. Nous verrons, que, de même que dans le cas booléen (c'est-à-dire dans le cas des langages), ceci permet de donner un automate à multiplicité avec au plus $n+1$ états.

— ∘ —

1 Expressions rationnelles

La définition des **expressions rationnelles sur A à multiplicité dans un semi-anneau** $\mathbb{K}$ se fait par récurrence.

DÉFINITION 6.1 *Soit* $\{0,1,+,\cdot,\star\}$ *un ensemble de symboles et* A *un alphabet.*

i) $\mathsf{0}$, $\mathsf{1}$, *et* a, *pour tout* a *appartenant à* A, *sont des expressions rationnelles (atomiques).*

ii) *Si* E *est une expression rationnelle, et* k *un élément de* $\mathbb{K}$, *alors* $(k\,\mathsf{E})$ *et* $(\mathsf{E}\,k)$ *sont des expressions rationnelles.*

iii) *Si* E *et* F *sont des expressions rationnelles, alors* $(\mathsf{E}+\mathsf{F})$, $(\mathsf{E}\cdot\mathsf{F})$ *et* $(\mathsf{E}^{\star})$ *aussi.*

Les symboles $\mathsf{0}$ et $\mathsf{1}$ sont donc des constantes, $+$ et $\cdot$ sont des opérateurs binaires; $\star$ est un opérateur unaire.
On note $\mathbb{K}\mathsf{RExp}A$ l'ensemble des expressions rationnelles sur A à multiplicité dans $\mathbb{K}$.

Exemple 24.1 Les semi-anneaux de coefficients qu'on manipule le plus souvent dans le cadre des séries formelles sont commutatifs : entiers, corps commutatifs, semi-anneaux « max ,+ »,*etc.* Toutefois, même si ce n'est pas la façon habituelle de considérer ces objets, les transductions rationnelles sur $A^* \times B^*$ ne sont rien d'autre que des séries rationnelles de $\mathcal{P}(B^*)\mathrm{Rat}\,A^*$. Dans cet exemple, $A = \{a,b\}$ et $B = \{\mathrm{x},\mathrm{y}\}$. On définit l'expression E_4 :

$$\mathsf{E}_4 = (\mathrm{y}((((\mathrm{x}(((a\cdot((\mathrm{y}b)^{\star}))\cdot)\mathrm{y}))+(((\mathrm{y}(b\cdot((\mathrm{x}a)^{\star})))\cdot b)\mathrm{x}))\mathrm{x})^{\star}))$$

On le voit, l'utilisation systématique de parenthèses, si elle permet d'éviter tout risque d'ambiguïté, rend, en revanche, l'expression difficilement lisible. C'est pourquoi on s'autorise à supprimer les parenthèses qui n'ôtent aucune ambiguïté. Pour cela, on fixe comme convention que tout produit ou somme de plusieurs facteurs est parenthésé par défaut à gauche et on utilise implicitement un certain nombre de règles qui seront détaillées plus tard. On obtient :

$$\mathsf{E}_4 = \mathrm{y}\Big(\mathrm{x}\big(a\cdot(\mathrm{y}\,b)^{\star}\cdot a\big)\,\mathrm{y}+\Big(\big(\mathrm{y}\,(b\cdot(a\,\mathrm{x})^{\star})\big)\cdot b\Big)\,\mathrm{x}\Big)^{\star}$$

La complexité d'une expression rationnelle peut être évaluée de différentes façons. La longueur (littérale) de l'expression est le paramètre qui permet d'exprimer la taille de l'automate que nous construirons à la section 4. Toutefois, la plupart des preuves que nous allons effectuer sont par récurrence sur la profondeur des expressions rationnelles.

Définition 6.2 *La* ***longueur d'une expression*** E, *notée* $\ell(\mathsf{E})$ *est le nombre de lettres qui apparaissent dans l'expression :*

$$\begin{aligned} \ell(\mathsf{0}) = \ell(\mathsf{1}) &= 0,\\ \forall a\in A \qquad \ell(a) &= 1,\\ \ell((k\,\mathsf{E})) = \ell((\mathsf{E}\,k)) = \ell((\mathsf{E}^{\star})) &= \ell(\mathsf{E}),\\ \ell((\mathsf{E}\cdot\mathsf{F})) = \ell((\mathsf{E}+\mathsf{F})) &= \ell(\mathsf{E}) + \ell(\mathsf{F}). \end{aligned}$$

La ***profondeur d'une expression*** E, *notée* $\mathsf{d}(\mathsf{E})$ *est définie récursivement par :*

$$\begin{aligned} \mathsf{d}(\mathsf{0}) = \mathsf{d}(\mathsf{1}) &= 0,\\ \forall a\in A \qquad \mathsf{d}(a) &= 0,\\ \mathsf{d}((k\,\mathsf{E})) = \mathsf{d}((\mathsf{E}\,k)) = \mathsf{d}((\mathsf{E}^{\star})) &= 1 + \mathsf{d}(\mathsf{E}),\\ \mathsf{d}((\mathsf{E}\cdot\mathsf{F})) = \mathsf{d}((\mathsf{E}+\mathsf{F})) &= 1 + \mathsf{max}(\mathsf{d}(\mathsf{E}),\mathsf{d}(\mathsf{F})). \end{aligned}$$

Définition 6.3 *Le* ***terme constant d'une expression rationnelle***[1] *est défini par récurrence sur la profondeur de l'expression :*

$$\begin{gathered}
\mathsf{c}(0) = 0_{\mathbb{K}}\,, \quad \mathsf{c}(1) = 1_{\mathbb{K}}, \\
\forall a \in A \qquad \mathsf{c}(a) = 0_{\mathbb{K}}, \\
\mathsf{c}((k\,\mathsf{E})) = k \otimes \mathsf{c}(\mathsf{E})\,, \quad \mathsf{c}((\mathsf{E}\,k)) = \mathsf{c}(\mathsf{E}) \otimes k, \\
\mathsf{c}((\mathsf{E}+\mathsf{F})) = \mathsf{c}(\mathsf{E}) \oplus \mathsf{c}(\mathsf{F})\,, \quad \mathsf{c}((\mathsf{E}\cdot\mathsf{F})) = \mathsf{c}(\mathsf{E}) \otimes \mathsf{c}(\mathsf{F}) \\
\textit{et} \quad \mathsf{c}((\mathsf{E}^{\star})) = \mathsf{c}(\mathsf{E})^{*} \quad \textit{si le second membre est défini dans } \mathbb{K}.
\end{gathered}$$

Une expression rationnelle E est une formule. Une telle formule peut être une **expression rationnelle valide** et dénoter une série.

Définition 6.4 *Une expression rationnelle est* ***valide*** *si son terme constant est défini. La* ***série dénotée par une expression valide*** E*, qu'on désigne par* $|\mathsf{E}|$*, est elle aussi définie par récurrence sur la profondeur de l'expression* E *:*

$$\begin{gathered}
|0| = 0_{\mathbb{K}}\,, \quad |1| = 1_{A^*}\,, \quad |a| = a\,, \quad \textit{pour tout } a \textit{ dans } A\,, \\
|(k\,\mathsf{E})| = k \otimes |\mathsf{E}|\,, \quad |(\mathsf{E}\,k)| = |\mathsf{E}| \otimes k\,, \\
|(\mathsf{E}+\mathsf{F})| = |\mathsf{E}| \oplus |\mathsf{F}|\,, \quad |(\mathsf{E}\cdot\mathsf{F})| = |\mathsf{E}| \otimes |\mathsf{F}|\,, \\
\textit{et} \quad |(\mathsf{E}^{\star})| = |\mathsf{E}|^{*}.
\end{gathered}$$

Deux expressions valides sont des ***expressions équivalentes*** *si elles dénotent la même série.*

Remarquons que la définition du terme constant est bien cohérente. En effet, $\mathsf{c}(\mathsf{E})$ et $|\mathsf{E}|$ sont définis par la même induction et on a :

$$\begin{aligned}
\mathsf{c}(|0|) = \mathsf{c}(0_{\mathbb{K}}) &= 0_{\mathbb{K}} = \mathsf{c}(0), \\
\mathsf{c}(|1|) = \mathsf{c}(1_{A^*}) &= 1_{\mathbb{K}} = \mathsf{c}(1), \\
\forall a \in A^* \qquad \mathsf{c}(|a|) = \mathsf{c}(a) &= 0_{\mathbb{K}} = \mathsf{c}(a).
\end{aligned}$$

On obtient immédiatement que $\mathsf{c}(\mathsf{E})$ est égal à $\mathsf{c}(|\mathsf{E}|)$.

L'équation qui permet d'interpréter l'étoile est rendue consistante par la proposition 1.1 : $|(\mathsf{E}^{\star})|$ est défini si et seulement si $\mathsf{c}(\mathsf{E})^{*}$ est défini, donc si et seulement si $(\mathsf{E}^{\star})$ est valide.

Remarque 6.1 Il faut noter que la proposition 1.1 permet de définir l'étoile d'une série avec terme constant dénotée par une expression E. Toutefois, cette proposition ne donne aucune information sur la forme de l'expression dénotant la série propre. La proposition peut donc être appliquée aux séries mais ne devra pas être utilisée pour effectuer des récurrences sur les expressions.

1. Cette définition est distincte de celle de « terme constant d'une série » donnée en 1.28 page 29.

Exemple 25.1 On définit l'expression E_1 de $\mathbb{Q}\mathsf{RExp}\{a,b\}$:

$$\mathsf{E}_1 = (((\tfrac{1}{6}\, a^\star) + (\tfrac{1}{3}\, b^\star))^\star).$$

Soit l'expression $\mathsf{F}_1 = ((\frac{1}{6}\, a^\star) + (\frac{1}{3}\, b^\star))$. On a $\mathsf{c}(\mathsf{F}_1) = \frac{1}{2}$, et donc $\mathsf{c}(\mathsf{F}_1)^* = 2$, et bien que $|\mathsf{F}_1|$ ne soit pas propre, la série dénotée par E_1 est bien définie.

Exemple 26.1 On définit l'expression E_2 de $\mathbb{Z}\mathsf{RExp}\{a,b\}$:

$$\mathsf{E}_2 = (a\, b\, a + (a\,(a - b\, a))).$$

Afin d'alléger l'écriture des expressions, on se permet certaines licences et on note $a\, b$ pour $(a \cdot b)$, $a\, b\, a$ pour $((a \cdot b) \cdot a)$ ou encore $(a - b\, a)$ pour $(a + (-1_{\mathbb{K}}\,(b \cdot a)))$.

Exemple 27.1 On définit l'expression E_3 de $\mathbb{N}\mathsf{RExp}\{a,b\}$:

$$\mathsf{E}_3 = 5\,((2\, a\, b) + ((3\, b) \cdot (4\,(a\, b)^\star)))^\star$$

Cet exemple apparaît dans l'article [11]. Les étoiles s'appliquent dans cette expression à des sous-expressions propres. Avec la topologie usuelle, c'est la seule configuration pour laquelle l'étoile d'une série à coefficients entiers peut être définie.

— ∘ —

Identités triviales. Bien qu'on désire travailler formellement sur les expressions rationnelles, et que, pour cette raison, on refuse de recourir à des axiomes tels que l'associativité ou la commutativité, nous aurons besoin de définir une forme « normale » pour les expressions. C'est pourquoi nous définissons les règles de réécriture suivantes qui peuvent être appliquées localement aux expressions rationnelles.

$$(1\, k) \equiv (k\, 1)\ , \quad (a\, k) = (k\, a), \tag{1}$$

$$(k\, 0) \equiv (0\, k) \equiv 0\ , \quad (0_{\mathbb{K}}\, \mathsf{E}) \equiv (\mathsf{E}\, 0_{\mathbb{K}}) \equiv 0\ , \quad (0 \cdot \mathsf{E}) \equiv (\mathsf{E} \cdot 0) \equiv 0\ , \tag{2}$$

$$0 + \mathsf{E} \equiv \mathsf{E} + 0 \equiv \mathsf{E}\ , \quad (1_{\mathbb{K}}\, \mathsf{E}) \equiv (\mathsf{E}\, 1_{\mathbb{K}}) \equiv \mathsf{E}\ , \tag{3}$$

$$((k\, 1) \cdot \mathsf{E}) \equiv (k\, \mathsf{E})\ , \quad (\mathsf{E} \cdot (k\, 1)) \equiv (\mathsf{E}\, k), \tag{4}$$

$$(k\,(k'\, \mathsf{E})) \equiv ([k \otimes k']\, \mathsf{E})\ , \quad ((\mathsf{E}\, k)\, k') \equiv (\mathsf{E}\,[k \otimes k'])\ , \quad ((k\, \mathsf{E})\, k') \equiv (k\,(\mathsf{E}\, k')). \tag{5}$$

L'identité (1) représente la commutativité des coefficients avec les expressions atomiques, les identités (2) reflètent le caractère absorbant du zéro; les identités (3) et (4) traduisent le fait que $0_{\mathbb{K}}$ et $1_{\mathbb{K}}$ sont des éléments neutres; l'identité (5) indique l'associativité du produit externe.

On vérifie immédiatement que l'interprétation de chaque membre de n'importe laquelle de ces identités est la même.

Noter que la commutativité des variables et des coefficients n'entraîne pas la commutativité des coefficients avec les expressions. Si $(\mathsf{E}\, k)$ est une expression, il est possible qu'un coefficient k' soit niché dans E et ne commute pas avec k. Dans ce cas $(k\, \mathsf{E})$ et $(\mathsf{E}\, k)$ risquent fort de ne pas commuter.

Les identités définies ci-dessus sont des propriétés locales et si chaque identité est considérée comme une règle de réécriture (dont le résultat est le membre droit), chaque expression rationnelle peut se réécrire en une **expression réduite** unique qui peut être calculée en temps proportionnel à la longueur de l'expression. Dans ce qui suit, on suppose qu'on applique systématiquement cette réduction.

REMARQUE 6.2 Les identités triviales ne reflètent en rien l'associativité ou la commutativité de l'addition et de la multiplication des séries. Ainsi, $(a + (b + c))$, $((a + b) + c)$ et $(a + (c + b))$ sont trois expressions réduites différentes qui dénotent la même série.

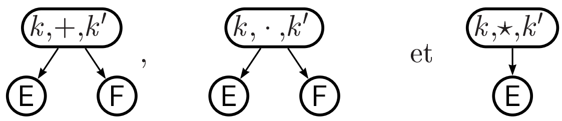
— ∘ —

2 Motivation de la dérivation

L'égalité (5) permet de voir les expressions rationnelles comme des arbres.

i) 0 , $k,1$ et k,a sont des expressions.

ii) Si E et F sont des expressions, alors, pour tous k, k' dans $\mathbb{K}$,



sont des expressions.

On voit que dans cette définition, chaque opérateurs est doté systématiquement de coefficients. L'identité (3) permet, en l'absence de tels coefficients, de supposer qu'il s'agit de $1_{\mathbb{K}}$.

EXEMPLE 24.2 L'arbre correspondant à l'expression E_4 est donné figure 1.

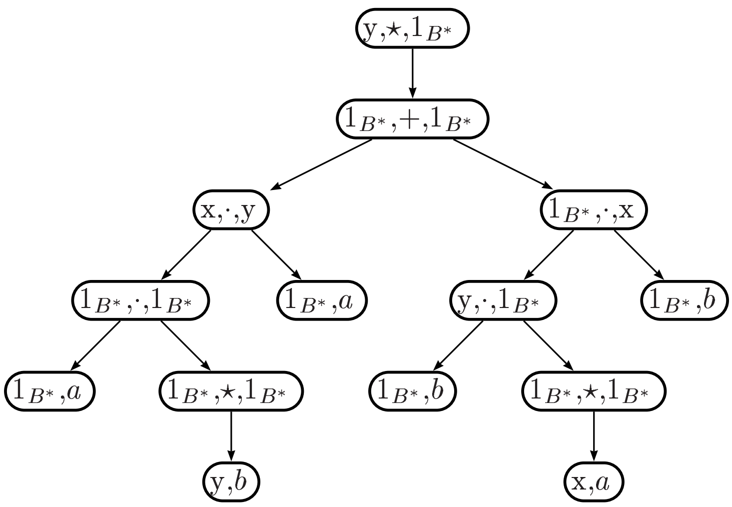


FIG. 1 – *L'arbre de l'expression* E_4.

Chaque nœud peut ensuite être « interprété ». La lecture d'un mot dans l'expression se fait de la façon suivante :

- Si le nœud est étiqueté par $+$, on explore *soit* le fils gauche, *soit* le fils droit, puis on remonte;
- si le nœud est étiqueté par $\cdot$, on explore le fils gauche, puis le fils droit, puis on remonte;
- si le nœud est étiqueté par $\star$, on explore le fils un nombre arbitraire de fois (éventuellement nul), puis on remonte;
- si le nœud est une feuille, on lit la constante qu'il contient puis on remonte;
- lorsqu'on descend, on prend en compte la multiplicité « d'entrée » du nœud.
- lorsqu'on remonte, on prend en compte la multiplicité « de sortie » du nœud.

La lecture se termine lorsqu'on essaie de remonter à partir de la racine.

Le comportement de chaque nœud peut être localement vu comme un automate :

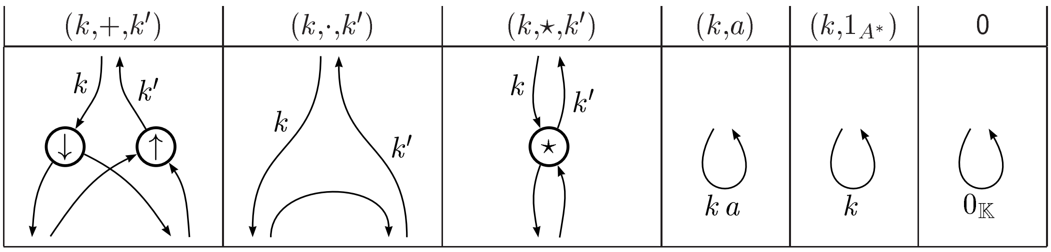


Fig. 2- L'interprétation des différents opérateurs rationnels.

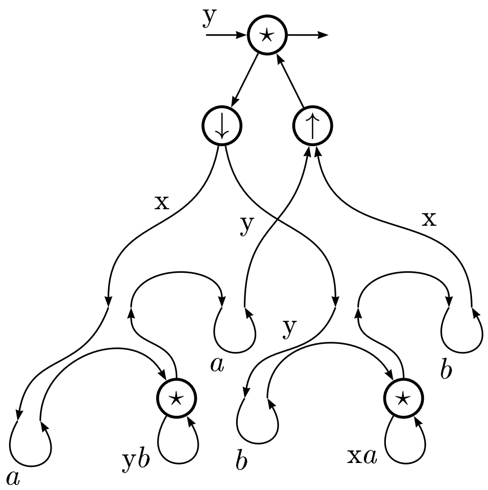
Le remplacement de chaque nœud de l'arbre de l'expression par la configuration indiquée ci-dessus donne un automate qui réalise la série dénotée par l'expression. Cet automate est généralisé; en effet, chaque transition est étiquetée par un mot, éventuellement vide.

La dérivation consiste à parcourir l'automate de l'expression. Le terme constant de l'expression est le poids total des chemins qui permettent de faire une boucle autour de la racine sans lire de lettre. La dérivée par rapport à une lettre est l'expression qui décrit l'endroit où l'on se trouve après avoir lu la lettre et le nombre de façons (comptées dans $\mathbb{K}$) de le faire. L'automate n'étant pas nécessairement déterministe, le résultat de la dérivation peut donner plusieurs expressions.

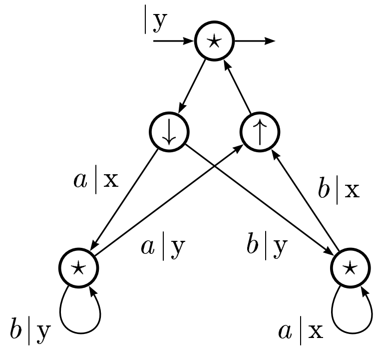
Ce procédé peut être systématisé et les dérivées calculées sur les expressions. C'est le propos du paragraphe suivant.

Exemple 24.3 En appliquant mécaniquement le remplacement des nœuds de l'arbre représentant l'expression E_4, on obtient l'automate présenté figure 3 a) qui se réécrit immédiatement en l'automate de la figure 3 b). Comme il s'agit d'un transducteur, on utilise la notations $a\,|\,\mathrm{x}$, plutôt que $\mathrm{x}\,a$.

— ◦ —



(a) Première version



(b) Seconde version

FIG. 3 – *L'automate issu de l'arbre de* E_4.

3 Dérivation et termes dérivés

On présente tout d'abord les « polynômes » d'expressions rationnelles. L'ensemble $\mathbb{K}\langle\mathbb{K}\mathsf{RExp}A\rangle$ des combinaisons linéaires d'expressions rationnelles, ou **polynômes d'expressions**, est un $\mathbb{K}$-semi-module (à gauche); l'addition y est commutative et la multiplication par un élément de $\mathbb{K}$ distributive :

$$k\,\mathsf{E} \oplus k'\,\mathsf{F} = k'\,\mathsf{F} \oplus k\,\mathsf{E} \qquad\qquad k\,\mathsf{E} \oplus k'\,\mathsf{E} = [k \oplus k']\mathsf{E}$$

Nous définissons de plus une loi de multiplication sur les monômes qui s'étend sur les polynômes par linéarité :

$$[k\,\mathsf{E}][k'\,\mathsf{F}] \equiv k\,(\mathsf{E}\cdot(k'\,\mathsf{F})), \tag{6}$$

$$\big([\mathsf{E}\oplus\mathsf{E}']\cdot\mathsf{F}\big) \equiv (\mathsf{E}\cdot\mathsf{F})\oplus(\mathsf{E}'\cdot\mathsf{F})\,, \quad \big(\mathsf{E}\cdot[\mathsf{F}\oplus\mathsf{F}']\big) \equiv (\mathsf{E}\cdot\mathsf{F}')\oplus(\mathsf{E}\cdot\mathsf{F}'). \tag{7}$$

$$\big([\mathsf{E}\oplus\mathsf{E}']\ k\big) \equiv (\mathsf{E}\,k)\oplus(\mathsf{E}'\cdot k)\,, \quad \big(k\ [\mathsf{E}\oplus\mathsf{E}']\big) \equiv (k\,\mathsf{E}')\oplus(k\,\mathsf{E}'). \tag{8}$$

Afin de différencier le plus possible les expressions des polynômes, nous essaierons de nous abstenir d'utiliser les parenthèses pour ces derniers. Ainsi, dans tout ce qui suit, $[k\,\mathsf{E}]$ ou $k\,\mathsf{E}$ désigne un monôme, alors que $(k\,\mathsf{E})$ représente une expression.

La série dénoté par un polynôme d'expressions rationnelles est obtenue par linéarité à partir de l'interprétation définie sur les expressions rationnelles (qui forment une base de $\mathbb{K}\langle\mathbb{K}\mathsf{RExp}A\rangle$).

REMARQUE 6.3 On pourrait être tenté de définir la multiplication des monômes par $[k\,\mathsf{E}][k'\,\mathsf{F}] \equiv [k\otimes k']\,(\mathsf{E}\cdot\mathsf{F})$; si $\mathbb{K}$ n'est pas commutatif, l'interprétation des membres gauche et droit peut alors être différente. C'est la raison pour laquelle on pose l'identité (6).

L'ensemble des polynômes d'expressions rationnelles n'est pas une semi-algèbre. En effet, la multiplication que nous avons définie n'est pas associative :

$$[[\mathsf{E}]\,[\mathsf{F}]]\,[\mathsf{G}] = ((\mathsf{E}\cdot\mathsf{F})\cdot\mathsf{G}) \neq (\mathsf{E}\cdot(\mathsf{F}\cdot\mathsf{G})) = [\mathsf{E}]\,[[\mathsf{F}]\,[\mathsf{G}]]$$

Ceci ne cause heureusement aucun problème pour traiter les dérivations.

— ◦ —

Derivation

Définition 6.5 *Soit* E *une expression de* $\mathbb{K}\mathsf{RExp}A$ *et* a *une lettre de* A. *La* ***dérivée de l'expression*** E ***par rapport à*** a, *notée* $\frac{\partial}{\partial a}\mathsf{E}$, *est un polynôme d'expressions rationnelles à coefficient dans* $\mathbb{K}$, *défini par récurence selon les formules suivantes :*

$$\frac{\partial}{\partial a}\mathsf{0} = \frac{\partial}{\partial a}\mathsf{1} = \mathsf{0}$$

$$\forall b \in A \qquad \frac{\partial}{\partial a} b = \begin{cases} \mathsf{1} & \textit{si } b = a \\ \mathsf{0} & \textit{sinon} \end{cases} \tag{9}$$

$$\frac{\partial}{\partial a}(k\,\mathsf{E}) = k\ \frac{\partial}{\partial a}\mathsf{E} \tag{10}$$

$$\frac{\partial}{\partial a}(\mathsf{E}\,k) = (\left[\frac{\partial}{\partial a}\mathsf{E}\right]\ k) \tag{11}$$

$$\frac{\partial}{\partial a}(\mathsf{E}{+}\mathsf{F}) = \frac{\partial}{\partial a}\mathsf{E} \oplus \frac{\partial}{\partial a}\mathsf{F} \tag{12}$$

$$\frac{\partial}{\partial a}(\mathsf{E}\cdot\mathsf{F}) = \left(\left[\frac{\partial}{\partial a}\mathsf{E}\right]\cdot\mathsf{F}\right) \oplus \mathsf{c}(\mathsf{E})\ \frac{\partial}{\partial a}\mathsf{F} \tag{13}$$

$$\frac{\partial}{\partial a}(\mathsf{E}^{\star}) = \mathsf{c}(\mathsf{E})^{*}\ \left(\left[\frac{\partial}{\partial a}\mathsf{E}\right]\cdot(\mathsf{E}^{\star})\right) \tag{14}$$

La dérivation d'un polynôme d'expressions est évidemment définie en étendant linéairement la dérivation :

$$\frac{\partial}{\partial a}\left[\bigoplus_{i\in I} k_i\,\mathsf{E}_i\right] = \bigoplus_{i\in I} k_i\ \frac{\partial}{\partial a}\mathsf{E}_i \tag{15}$$

Les équations (10) à (14) sont sensiblement différentes de celles posées par Brzozowski [9]. En effet, l'opérateur (symbolique) « + » est remplacé, dans les équations (12) et (13), par l'opération « $\oplus$ ». Ceci est la généralisation de l'idée d'Antimirov; les langages rationnels sont en effet des séries rationnelles à multiplicité dans le semi-anneau de Boole et l'addition est alors l'union.

Les équations (10), (14) et (15) introduisent la prise en compte des multiplicités.

On peut relever que l'équation (14) n'est définie que si l'expression E est valide.

Le résultat de la dérivation d'une expression est donc un polynôme d'expressions à coefficients dans $\mathbb{K}$.

Contrairement au cas booléen, les polynômes obtenus en itérant la dérivation peuvent ne pas être en nombre fini. Le théorème 6.1 établira cependant que tous ces polynômes sont engendrés par un nombre fini d'expressions.

DÉFINITION 6.6 *La **dérivée d'une expression par rapport à un mot** u est définie par récurrence sur la longueur de u. Par convention, la dérivation par rapport au mot vide est l'identité.*

$$\forall u \in A^* \,, \quad \forall a \in A \qquad \frac{\partial}{\partial ua}\mathsf{E} = \frac{\partial}{\partial a}\left(\frac{\partial}{\partial u}\mathsf{E}\right) \tag{16}$$

— ∘ —

A présent que la dérivation est définie, il faut établir le sens qu'a cette opération syntaxique. Le lien entre les dérivées d'une expression et les quotients d'une série est exposé dans la proposition suivante :

PROPOSITION 6.1 *L'interprétation de la dérivée d'une expression rationnelle par rapport à un mot est le quotient par rapport à ce mot de l'interprétation de l'expression :*

$$\forall u \in A^* \qquad |\frac{\partial}{\partial u}(\mathsf{E})| = u^{-1}|\mathsf{E}|$$

Ce résultat peut être montré directement, mais il apparaîtra comme un corollaire naturel de propriétés plus fines établies sur les dérivées.

REMARQUE 6.4 Il faut souligner dès à présent la différence de statut entre dérivée et quotient. Le quotient d'une série par un mot est un objet mathématique, la dérivée d'une expression est une représentation de cet objet. Ainsi, on peut obtenir plusieurs dérivées qui représentent le même quotient. Le fait que les quotients d'une série rationnelle appartiennent à un $\mathbb{K}$-semi-module finiment engendré n'implique pas qu'il en est de même pour les dérivées d'une expression qui la représente. Nous verrons d'ailleurs qu'il est possible qu'il existe des systèmes générateurs d'un $\mathbb{K}$-semi-module de quotients d'une série de cardinal strictement inférieur aux plus petits systèmes générateurs du $\mathbb{K}$-semi-module des dérivées d'une expression correspondant à la série.

La dérivée d'une expression par rapport à un mot est donnée explicitement par les formules suivantes :

PROPOSITION 6.2 *Pour tout u dans A^+, pour tout couple* (E,F) *d'expressions,*

i)

$$\frac{\partial}{\partial u}(k\,\mathsf{E}) = k\,\frac{\partial}{\partial u}\mathsf{E}\,, \qquad \frac{\partial}{\partial u}(\mathsf{E}\,k) = (\left[\frac{\partial}{\partial u}\mathsf{E}\right]\,k),$$

ii)

$$\frac{\partial}{\partial u}(\mathsf{E}+\mathsf{F}) = \frac{\partial}{\partial u}\,\mathsf{E} \oplus \frac{\partial}{\partial u}\,\mathsf{F},$$

iii)

$$\frac{\partial}{\partial u}(\mathsf{E}\cdot\mathsf{F}) = \left[\frac{\partial}{\partial u}\,\mathsf{E}\right]\cdot\mathsf{F} \oplus \left[\bigoplus_{\substack{u=v\,w\\ v\in A^*,\, w\in A^+}} \mathsf{c}(\frac{\partial}{\partial v}\,\mathsf{E})\ \frac{\partial}{\partial w}\,\mathsf{F}\right],$$

iv)

$$\frac{\partial}{\partial u}(\mathsf{E}^\star) = \bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\dots,v_n\in A^+}} \mathsf{c}(\mathsf{E})^*\,\mathsf{c}(\frac{\partial}{\partial v_1}\,\mathsf{E})\ \mathsf{c}(\mathsf{E})^*\cdots\mathsf{c}(\mathsf{E})^*\,\mathsf{c}(\frac{\partial}{\partial v_{n\text{-}1}}\,\mathsf{E})\ \mathsf{c}(\mathsf{E})^*\ \left([\frac{\partial}{\partial v_n}\,\mathsf{E}]\cdot(\mathsf{E}^\star)\right).$$

Démonstration. La preuve est par récurrence sur la longueur de u. Si u est de longueur 1, on vérifie que les formules données rejoignent celles de la définition 6.5. On suppose que u vérifie la proposition. Soit E et F deux expressions rationnelles et a une lettre de A. Posons

$$\frac{\partial}{\partial a}\,\mathsf{E} = \bigoplus k_i\,\mathsf{E}_i \text{ et } \frac{\partial}{\partial a}\,\mathsf{F} = \bigoplus k_j\,\mathsf{F}_j.$$

On obtient les égalités suivantes :

i)

$$\frac{\partial}{\partial au}(k\,\mathsf{E}) = \frac{\partial}{\partial u}\,\frac{\partial}{\partial a}(k\,\mathsf{E}) = \frac{\partial}{\partial u}\left[k\ \frac{\partial}{\partial a}\,\mathsf{E}\right] = k\ \frac{\partial}{\partial u}\left[\frac{\partial}{\partial a}\,\mathsf{E}\right] = k\ \frac{\partial}{\partial au}\,\mathsf{E}.$$

$$\begin{aligned}\frac{\partial}{\partial au}(\mathsf{E}\,k) = \frac{\partial}{\partial u}\,\frac{\partial}{\partial a}(\mathsf{E}\,k) = \frac{\partial}{\partial u}[\bigoplus k_i\,(\mathsf{E}_i\,k)] = \bigoplus k_i\ \frac{\partial}{\partial u}(\mathsf{E}_i\,k) &= \bigoplus k_i([\frac{\partial}{\partial u}\,\mathsf{E}_i]\,k)\\ &= \left([\bigoplus k_i\,\frac{\partial}{\partial u}\,\mathsf{E}_i]\,k\right) = \left(\frac{\partial}{\partial u}[\bigoplus k_i\mathsf{E}_i]\,k\right)\\ &= \left(\left[\frac{\partial}{\partial u}[\frac{\partial}{\partial a}\,\mathsf{E}]\right]\ k\right) = \left([\frac{\partial}{\partial au}\,\mathsf{E}]\,k\right).\end{aligned}$$

ii)

$$\frac{\partial}{\partial au}(\mathsf{E}{+}\mathsf{F}) = \frac{\partial}{\partial u}[\frac{\partial}{\partial a}\,\mathsf{E} \oplus \frac{\partial}{\partial a}\,\mathsf{F}] = \frac{\partial}{\partial u}[\frac{\partial}{\partial a}\,\mathsf{E}] \oplus \frac{\partial}{\partial u}[\frac{\partial}{\partial a}\,\mathsf{F}] = \frac{\partial}{\partial au}\,\mathsf{E} \oplus \frac{\partial}{\partial au}\,\mathsf{F}.$$

iii)

$$
\begin{aligned}
\frac{\partial}{\partial au}(\mathsf{E}{\cdot}\mathsf{F}) &= \frac{\partial}{\partial u}\left[\frac{\partial}{\partial a}(\mathsf{E}{\cdot}\mathsf{F})\right] = \frac{\partial}{\partial u}\left[\left([\frac{\partial}{\partial a}\,\mathsf{E}]\cdot\mathsf{F}\right)\oplus \mathsf{c}(\mathsf{E})\ \frac{\partial}{\partial a}\,\mathsf{F}\right] \\
&= \left(\frac{\partial}{\partial u}[\frac{\partial}{\partial a}\,\mathsf{E}]{\cdot}\mathsf{F}\right)\oplus\left[\bigoplus_{\substack{u=v\,w\\ v\in A^*,\,w\in A^+}} \mathsf{c}(\frac{\partial}{\partial v}[\frac{\partial}{\partial a}\,\mathsf{E}])\ \frac{\partial}{\partial w}\,\mathsf{F}\right]\oplus \mathsf{c}(\mathsf{E})\ \frac{\partial}{\partial au}\,\mathsf{F} \\
&= \left([\frac{\partial}{\partial au}\,\mathsf{E}]{\cdot}\mathsf{F}\right)\oplus\left[\bigoplus_{\substack{au=v\,w\\ v\in A^*,\,w\in A^+}} \mathsf{c}(\frac{\partial}{\partial v}\,\mathsf{E})\ \frac{\partial}{\partial w}\,\mathsf{F}\right]
\end{aligned}
$$

iv)

$$
\begin{aligned}
\frac{\partial}{\partial au}(\mathsf{E}^\star) &= \mathsf{c}(\mathsf{E})^*\ \frac{\partial}{\partial u}\left([\frac{\partial}{\partial a}\,\mathsf{E}]{\cdot}(\mathsf{E}^\star)\right) \\
&= \mathsf{c}(\mathsf{E})^*\left[\left([\frac{\partial}{\partial au}\,\mathsf{E}]{\cdot}(\mathsf{E}^\star)\right)\oplus \bigoplus_{\substack{au=v\,w\\ v\in A^*,\,w\in A^+}} [\mathsf{c}(\frac{\partial}{\partial av}\,\mathsf{E})\ \frac{\partial}{\partial w}(\mathsf{E}^\star)]\right] \\
&= \mathsf{c}(\mathsf{E})^*\ \left([\frac{\partial}{\partial au}\,\mathsf{E}]{\cdot}(\mathsf{E}^\star)\right)\oplus \\
&\quad \bigoplus_{\substack{au=v_0\,w\\ v_0\in A^*,\,w\in A^+}}\ \bigoplus_{\substack{w=v_1\ldots v_n\\ v_n\in A^+}} \mathsf{c}(\mathsf{E})^*\,\mathsf{c}(\frac{\partial}{\partial av_0}\,\mathsf{E})\ \mathsf{c}(\mathsf{E})^*\,\mathsf{c}(\frac{\partial}{\partial v_1}\,\mathsf{E})\ldots \mathsf{c}(\mathsf{E})^*\left([\frac{\partial}{\partial v_n}\,\mathsf{E}]{\cdot}(\mathsf{E}^\star)\right) \\
&= \bigoplus_{\substack{au=v_1\ldots v_n\\ v_n\in A^+}} \mathsf{c}(\mathsf{E})^*\ \mathsf{c}(\frac{\partial}{\partial v_1}\,\mathsf{E})\ldots \mathsf{c}(\frac{\partial}{\partial v_{n\text{-}1}}\,\mathsf{E})\ \mathsf{c}(\mathsf{E})^*\ \left([\frac{\partial}{\partial v_n}\,\mathsf{E}]{\cdot}(\mathsf{E}^\star)\right)
\end{aligned}
$$

□

Exemple 25.2

$$\frac{\partial}{\partial a}\,\mathsf{E}_1 = \frac{\partial}{\partial a}\,(\mathsf{F}_1{}^\star) = 2\ \frac{\partial}{\partial a}\left(\tfrac{1}{6}\,a^\star\right)\cdot\mathsf{F}_1{}^\star \oplus 2\ \frac{\partial}{\partial a}\left(\tfrac{1}{3}\,b^\star\right)\cdot\mathsf{F}_1{}^\star = \tfrac{1}{3}\ (a^\star\cdot\mathsf{F}_1{}^\star)$$

$$\frac{\partial}{\partial b}\,\mathsf{E}_1 = 2\ \frac{\partial}{\partial b}\left(\tfrac{1}{3}\,b^\star\right)\cdot\mathsf{F}_1{}^\star = \tfrac{2}{3}\ (b^\star\cdot\mathsf{F}_1{}^\star)$$

$$\frac{\partial}{\partial aa}\,\mathsf{E}_1 = \tfrac{1}{3}\ \frac{\partial}{\partial a}\,(a^\star\cdot\mathsf{F}_1{}^\star) = \tfrac{1}{3}\ \left(\frac{\partial}{\partial a}\,a^\star\right)\cdot\mathsf{F}_1{}^\star \oplus \tfrac{1}{3}\ \mathsf{c}(a)^*\ \frac{\partial}{\partial a}\,(\mathsf{F}_1{}^\star)$$

$$= \tfrac{1}{3}\ (a^\star\cdot\mathsf{F}_1{}^\star) \oplus \tfrac{1}{9}\ (a^\star\cdot\mathsf{F}_1{}^\star) = \tfrac{4}{9}\ (a^\star\cdot\mathsf{F}_1{}^\star)$$

$$\frac{\partial}{\partial ab}\,\mathsf{E}_1 = \tfrac{1}{3}\ \frac{\partial}{\partial b}\,(a^\star\cdot\mathsf{F}_1{}^\star) = \tfrac{1}{3}\ \left(\frac{\partial}{\partial b}\,a^\star\right)\cdot\mathsf{F}_1{}^\star \oplus \tfrac{1}{3}\ \mathsf{c}(a)^*\ \frac{\partial}{\partial b}\,(\mathsf{F}_1{}^\star) = \tfrac{2}{9}\ (b^\star\cdot\mathsf{F}_1{}^\star)$$

$$\frac{\partial}{\partial ba}\,\mathsf{E}_1 = \tfrac{2}{3}\ \frac{\partial}{\partial a}\,(b^\star\cdot\mathsf{F}_1{}^\star) = \tfrac{2}{3}\ \left(\frac{\partial}{\partial a}\,b^\star\right)\cdot\mathsf{F}_1{}^\star \oplus \tfrac{2}{3}\ \mathsf{c}(b)^*\ \frac{\partial}{\partial a}\,(\mathsf{F}_1{}^\star) = \tfrac{2}{9}\ (a^\star\cdot\mathsf{F}_1{}^\star)$$

$$\frac{\partial}{\partial bb}\,\mathsf{E}_1 = \tfrac{2}{3}\ \frac{\partial}{\partial b}\,(b^\star\cdot\mathsf{F}_1{}^\star) = \tfrac{2}{3}\ \left(\frac{\partial}{\partial b}\,b^\star\right)\cdot\mathsf{F}_1{}^\star \oplus \tfrac{2}{3}\ \mathsf{c}(b)^*\ \frac{\partial}{\partial b}\,(\mathsf{F}_1{}^\star)$$

$$= \tfrac{2}{3}\ (b^\star\cdot\mathsf{F}_1{}^\star) \oplus \tfrac{4}{9}\ (b^\star\cdot\mathsf{F}_1{}^\star) = \tfrac{10}{9}\ (b^\star\cdot\mathsf{F}_1{}^\star)$$

Exemple 26.2

$$\frac{\partial}{\partial a}\,\mathsf{E}_2 = b\,a \oplus (a - b\,a) \qquad\qquad \frac{\partial}{\partial b}\,\mathsf{E}_2 = 0$$

$$\frac{\partial}{\partial aa}\,\mathsf{E}_2 = \frac{\partial}{\partial a}\,b\,a \oplus \frac{\partial}{\partial a}(a - b\,a) = 1$$

$$\frac{\partial}{\partial ab}\,\mathsf{E}_2 = \frac{\partial}{\partial b}\,b\,a \oplus \frac{\partial}{\partial b}(a - b\,a) = a \oplus (-1_{\mathbb{K}})\,a = 0$$

— ∘ —

Termes dérivés. Nous énonçons maintenant le théorème principal qui est une généralisation du résultat d'Antimirov.

Théorème 6.1 *Soit* E *une expression de* $\mathbb{K}\mathsf{RExp}A$. *Il existe un entier* m *inférieur à* $\ell(\mathsf{E})$, *et* m *expressions rationnelles* K_1, K_2, … , K_m *telles que, pour tout mot* u *dans* A^+, *il existe* m *coefficients* $k_1^{(u)}$, $k_2^{(u)}$, … , $k_m^{(u)}$ *de* $\mathbb{K}$ *tels que*

$$\frac{\partial}{\partial u}\,\mathsf{E} = \bigoplus_{i=1}^{i=m} k_i^{(u)}\,\mathsf{K}_i.$$

En fait, le théorème 6.1 est un corollaire de la proposition suivante. Celle-ci permet, en outre, de décrire un procédé plus proche de l'algorithme effectif présenté par la suite.

Proposition 6.3 *Soit* E *une expression de* $\mathbb{K}\mathsf{RExp}A$. *Il existe un entier* n *inférieur à* $\ell(\mathsf{E})$, *et* n *expressions rationnelles* K_1, K_2, … , K_n *telles que, pour toute lettre* a *de* A, *il existe* n *coefficients* $k_1^{(a)}$, $k_2^{(a)}$, … , $k_n^{(a)}$, *et* n^2 *coefficients* $\{z_{i,j}^{(a)}\}_{i,j\in[n]}$ *in* $\mathbb{K}$ *tels que*

i) $\dfrac{\partial}{\partial a}\mathsf{E} = \bigoplus_{i\in[n]} k_i^{(a)}\,\mathsf{K}_i\,;$

ii) $\forall i \in [n] \qquad \dfrac{\partial}{\partial a}\mathsf{K}_i = \bigoplus_{j\in[n]} z_{i,j}^{(a)}\,\mathsf{K}_j$

Les expressions K_i, dont l'existence est assurée par la proposition, sont appelés les **termes dérivés de** E.

Si $\mathbb{K}$ est le semi-anneau de Boole, ce sont exactement ce qu'Antimirov [3] appelle les « dérivées partielles » de E, avec la justification que ce sont des « parts » des dérivées de E. Comme le terme *dérivée partielle* évoque en mathématiques le fait de dériver selon seulement un paramètre (ici, une lettre), et que les dérivées telles que les définit Brzozowski sont, à cet égard, déjà partielles, nous préférons nous abstenir d'employer ce terme.

Exemple 25.3 Les termes dérivés de E_1 sont $(a^* \cdot \mathsf{F_1}^*)$ et $(b^* \cdot \mathsf{F_1}^*)$.

Il nous faut maintenant prouver la proposition.

Démonstration. La preuve est par récurrence sur la profondeur de l'expression E. L'énoncé est trivialement vrai pour les expressions atomiques $\mathsf{0}$, $\mathsf{1}$ et a (pour a dans A).

On prouve successivement que si le résultat est vrai pour deux expressions rationnelles E et F, il l'est pour :

a) $(k\,\mathsf{E})$ (pour k dans $\mathbb{K}$). En effet, d'après l'équation (10), les termes dérivés de $(k\,\mathsf{E})$ sont les mêmes que ceux de E.

b) $(\mathsf{E}\,k)$. Le nombre de termes dérivés est le même que celui de E, puisque d'après l'équation (11), il s'agit des $(\mathsf{K}_i\,k)$, où les K_i sont les termes dérivés de E.

c) $(\mathsf{E}+\mathsf{F})$. En effet, avec des notations évidentes,

$$\frac{\partial}{\partial a}(\mathsf{E}+\mathsf{F}) = \frac{\partial}{\partial a}\mathsf{E} \oplus \frac{\partial}{\partial a}\mathsf{F} = \bigoplus_{i\in[1;n]} k_i^{(a)}\,\mathsf{K}_i \oplus \bigoplus_{j\in[1;m]} l_j^{(a)}\,\mathsf{L}_j\,.$$

L'ensemble des termes dérivés de $(\mathsf{E}+\mathsf{F})$ est donc l'union de ceux de E et de ceux de F, ce qui répond à la proposition.

d) $(\mathsf{E}\cdot\mathsf{F})$. De la même façon,

$$\frac{\partial}{\partial a}(\mathsf{E}\cdot\mathsf{F}) = \left(\left[\frac{\partial}{\partial a}\mathsf{E}\right]\cdot\mathsf{F}\right) \oplus \mathsf{c}(\mathsf{E})\ \frac{\partial}{\partial a}\mathsf{F} = \bigoplus_{i\in[1;n]} k_i^{(a)}\,(\mathsf{K}_i\cdot\mathsf{F}) \oplus \bigoplus_{j\in[1;m]} \left(\mathsf{c}(\mathsf{E})\,l_j^{(a)}\right)\,\mathsf{L}_j$$

L'ensemble des termes dérivés de $(\mathsf{E}\cdot\mathsf{F})$ est donc l'union des $(\mathsf{K}_i\cdot\mathsf{F})_{i\in[1;n]}$ et des termes dérivés de F; on vérifie que cet ensemble est bien clos :

$$\frac{\partial}{\partial a}(\mathsf{K}_i\cdot\mathsf{F}) = \bigoplus_{p\in[1;n]} z_{i,p}^{(a)}\,(\mathsf{K}_p\cdot\mathsf{F}) \oplus \bigoplus_{j\in[1;m]} \left[\mathsf{c}(\mathsf{K}_i)\otimes l_j^{(a)}\right]\,\mathsf{L}_j$$

e) $(\mathsf{E}^\star)$. On calcule les termes dérivés :

$$\frac{\partial}{\partial a}(\mathsf{E}^\star) = \mathsf{c}(\mathsf{E})^*\ \left(\left[\frac{\partial}{\partial a}\mathsf{E}\right]\cdot(\mathsf{E}^\star)\right) = \bigoplus_{i\in[1;n]} \left[\mathsf{c}(\mathsf{E})^\star\otimes k_i^{(a)}\right]\,(\mathsf{K}_i\cdot\mathsf{E}^\star)$$

Le nombre des termes dérivés ne change pas, puisqu'il s'agit des $(\mathsf{K}_i{\cdot}\mathsf{E}^\star)_{i\in[1;n]}$. Cet ensemble est bien clos pour la dérivation :

$$\frac{\partial}{\partial a}(\mathsf{K}_i \cdot \mathsf{E}^\star) = \bigoplus_{j\in[n]} z_{i,j}^{(a)}\,(\mathsf{K}_j \cdot \mathsf{E}^\star) \oplus \bigoplus_{j\in[n]} \left[\mathsf{c}(\mathsf{K}_i) \otimes \mathsf{c}(\mathsf{E})^* \otimes k_j^{(a)}\right]\,(\mathsf{K}_j \cdot \mathsf{E}^\star)$$

□

La preuve du théorème 6.1 découle naturellement de la proposition et évite l'utilisation des formules lourdes introduites dans la propositions 6.2.

Démonstration du théorème 6.1. La preuve est par récurrence sur la longueur de u. Si u est réduit à une lettre, le théorème est équivalent à l'égalité i) de la proposition 6.3. Pour tout u dans A^+, et tout a dans A,

$$\frac{\partial}{\partial ua}\,\mathsf{E} = \frac{\partial}{\partial a}\left(\frac{\partial}{\partial u}\,\mathsf{E}\right) = \bigoplus_{i\in[1;n]} k_i^{(u)}\,\frac{\partial}{\partial a}\,\mathsf{K}_i$$

$$= \bigoplus_{i\in[1;n]} k_i^{(u)}\left[\bigoplus_{j\in[1;n]} z_{i,j}^{(a)}\,\mathsf{K}_j\right] = \bigoplus_{j\in[1;n]}\left[\bigoplus_{i\in[1;n]}\left[k_i^{(u)} \otimes z_{i,j}^{(a)}\right]\right]\mathsf{K}_j \qquad (17)$$

□

Comme nous l'indique la preuve du théorème 6.1, les expressions qui apparaissent dans la dérivée d'une expression E par rapport à un mot sont toutes des termes dérivés de l'expression. Cependant, il se peut que tous n'apparaissent pas; les termes dérivés qui n'apparaissent dans aucun polynôme de l'ensemble $\{\frac{\partial}{\partial u}\,\mathsf{E} \mid u \in A^*\}$ sont appelés **termes dérivés fantômes**. Ils seront étudiés plus avant dans le paragraphe suivant.

Exemple 26.3 Les termes dérivés de E_2 sont $b\,a$, $(a - b\,a)$, a et 1, parmi lesquels $a = \frac{\partial}{\partial b}(a - b\,a)$ est un terme dérivé fantôme de E_2 (il n'apparaît pas dans l'exemple 26.2).

Remarque 6.5 La proposition 6.3 n'est pas seulement un lemme permettant de prouver le théorème 6.1; c'est aussi la description de l'algorithme le plus naturel pour calculer les termes dérivés de E. Ceux-ci sont calculés par dérivations successives jusqu'à ce que l'ensemble obtenu soit clos.

Le théorème suivant permet d'établir le lien entre le coefficient du mot u dans la série dénotée par une expression E et la dérivée de E par u.

Théorème 6.2 *Soit E une expression de $\mathbb{K}\mathsf{RExp}A$, et $\mathsf{K}_1, \mathsf{K}_2, \ldots, \mathsf{K}_m$ ses termes dérivés. Soit, pour tout mot u de A^+, $k_1^{(u)}, k_2^{(u)}, \ldots, k_m^{(u)}$ les coefficients définis dans le théorème 6.1. On a alors les égalités suivantes :*

$$\langle|\mathsf{E}|,u\rangle = \mathsf{c}(\frac{\partial}{\partial u}\,\mathsf{E}) = \bigoplus_{i=1}^{i=m} k_i^{(u)} \otimes \mathsf{c}(\mathsf{K}_i) \qquad (18)$$

Démonstration. La preuve est par récurrence sur la profondeur de l'expression et utilise la proposition 6.2.

L'égalité est vraie pour les expressions $\mathsf{0}$ et $\mathsf{1}$; la dérivée par rapport à n'importe quel mot u de A^+ est nulle et le coefficient de u dans ces séries est effectivement nul.

De même, si l'expression est une lettre a de A, sauf si $u = a$, auquel cas la dérivée est égale à $\mathsf{1}$ et l'égalité est vraie.

Pour toute paire d'expressions (E,F) pour lesquelles l'égalité est vraie, pour tout k dans $\mathbb{K}$ et tout u dans A^+, on a les égalités suivantes :

$$\begin{aligned}
\langle|(k\,\mathsf{E})|,u\rangle &= k \otimes \langle|\mathsf{E}|,u\rangle = k \otimes \mathsf{c}(\frac{\partial}{\partial u}\,\mathsf{E}) = \mathsf{c}(\frac{\partial}{\partial u}(k\,\mathsf{E}))\\
\langle|(\mathsf{E}\,k)|,u\rangle &= \langle|\mathsf{E}|,u\rangle \otimes k = \mathsf{c}(\frac{\partial}{\partial u}\,\mathsf{E}) \otimes k = \mathsf{c}([\frac{\partial}{\partial u}\,\mathsf{E}]\,k) = \mathsf{c}(\frac{\partial}{\partial u}(\mathsf{E}\,k))\\
\langle|(\mathsf{E}+\mathsf{F})|,u\rangle &= \langle|\mathsf{E}|,u\rangle \oplus \langle|\mathsf{F}|,u\rangle\\
&= \mathsf{c}(\frac{\partial}{\partial u}\,\mathsf{E}) \oplus \mathsf{c}(\frac{\partial}{\partial u}\,\mathsf{F})\\
&= \mathsf{c}(\frac{\partial}{\partial u}(\mathsf{E}+\mathsf{F}))\\
\langle|(\mathsf{E}\cdot\mathsf{F})|,u\rangle &= \langle|(\mathsf{E})|,u\rangle \otimes \langle|(\mathsf{F})|,1_{A^*}\rangle\\
&\qquad \oplus \bigoplus_{\substack{u=v\,w\\ v\in A^*,\, w\in A^+}} \langle|(\mathsf{E})|,v\rangle \otimes \langle|(\mathsf{F})|,w\rangle\\
&= \mathsf{c}(\frac{\partial}{\partial u}\,E) \otimes \mathsf{c}(\mathsf{F}) \oplus \bigoplus_{\substack{u=v\,w\\ v\in A^*,\, w\in A^+}} \mathsf{c}(\frac{\partial}{\partial v}\,\mathsf{E}) \otimes \mathsf{c}(\frac{\partial}{\partial w}\,\mathsf{F})\\
&= \mathsf{c}\left(\left[\frac{\partial}{\partial u}\,E\right]\cdot\mathsf{F} \oplus \bigoplus_{\substack{u=v\,w\\ v\in A^*,\, w\in A^+}} \mathsf{c}(\frac{\partial}{\partial v}\,\mathsf{E})\,\frac{\partial}{\partial w}\,\mathsf{F}\right)\\
&= \mathsf{c}(\frac{\partial}{\partial u}(\mathsf{E}\cdot\mathsf{F}))
\end{aligned}$$

Toutes ces équations, ainsi que la seconde égalité du théorème, proviennent directement de la linéarité de $\mathsf{c}(\mathsf{E})$. Avant d'utiliser les mêmes arguments dans le but de prouver le résultat pour $\mathsf{E}^\star$, on utilise la proposition 1.1. Ceci permet d'éviter d'avoir à traiter une somme infinie; toutefois, on prend soin de revenir ensuite à la série non propre, puisque c'est la seule dont on connaît l'expression.

$$\begin{aligned}
\langle|(\mathsf{E}^\star)|,u\rangle &= \langle(\mathsf{c}(\mathsf{E})^* \otimes |\mathsf{E}|_p)^* \otimes \mathsf{c}(\mathsf{E})^*,u\rangle = \langle(\mathsf{c}(\mathsf{E})^* \otimes |\mathsf{E}|_p)^*,u\rangle \otimes \mathsf{c}(\mathsf{E})^*\\
&= \bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\ldots,v_n\in A^+}} \langle\mathsf{c}(\mathsf{E})^* \otimes |\mathsf{E}|_p,v_1\rangle \otimes \ldots \otimes \langle\mathsf{c}(\mathsf{E})^* \otimes |\mathsf{E}|_p,v_n\rangle \otimes \mathsf{c}(\mathsf{E})^*\\
&= \bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\ldots,v_n\in A^+}} \mathsf{c}(\mathsf{E})^* \otimes \langle|\mathsf{E}|_p,v_1\rangle \otimes \ldots \otimes \mathsf{c}(\mathsf{E})^* \otimes \langle|\mathsf{E}|_p,v_n\rangle \otimes \mathsf{c}(\mathsf{E})^*
\end{aligned}$$

Comme les v_i sont *tous différents du mot vide*, $\langle |\mathsf{E}|_p, v_i\rangle = \langle |\mathsf{E}|, v_i\rangle$.

$$\begin{aligned}
\langle |(\mathsf{E}^\star)|, u\rangle &= \bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\ldots,v_n\in A^+}} \mathsf{c}(\mathsf{E})^* \otimes \mathsf{c}(\frac{\partial}{\partial v_1}\mathsf{E}) \otimes \ldots \otimes \mathsf{c}(\mathsf{E})^* \otimes \mathsf{c}(\frac{\partial}{\partial v_n}\mathsf{E}) \otimes \mathsf{c}(\mathsf{E})^* \\
&= \mathsf{c}\left(\bigoplus_{\substack{u=v_1v_2\cdots v_n\\ v_1,v_2,\ldots,v_n\in A^+}} \mathsf{c}(\mathsf{E})^* \otimes \mathsf{c}(\frac{\partial}{\partial v_1}\mathsf{E}) \otimes \ldots \otimes \mathsf{c}(\mathsf{E})^* \left(\left[\frac{\partial}{\partial v_n}\mathsf{E}\right] \cdot (\mathsf{E}^\star)\right)\right) \\
&= \mathsf{c}(\frac{\partial}{\partial u}\mathsf{E}^\star)
\end{aligned}$$

□

La proposition 6.1 apparaît alors comme un corollaire :

Démonstration de la proposition 6.1. Une preuve par récurrence immédiate montre que, pour tout couple de mots u et v de A^* et pour toute expression E,

$$\frac{\partial}{\partial uv}\mathsf{E} = \frac{\partial}{\partial v}\left[\frac{\partial}{\partial u}\mathsf{E}\right],$$

D'où, pour toute paire de mots u et v de A^*,

$$\langle u^{-1}|\mathsf{E}|, v\rangle = \langle |\mathsf{E}|, uv\rangle = \mathsf{c}(\frac{\partial}{\partial uv}\mathsf{E}) = \mathsf{c}\left(\frac{\partial}{\partial v}\left[\frac{\partial}{\partial u}\mathsf{E}\right]\right) = \langle |\frac{\partial}{\partial u}\mathsf{E}|, v\rangle$$

□

Remarque 6.6 La dérivation et le quotient sont deux actions à droite de A^* sur l'ensemble des polynômes d'expressions rationnelles et l'ensemble des séries respectivement. Le théorème 6.1 nous dit que l'orbite d'une expression rationnelle sous l'action de A^* appartient à un $\mathbb{K}$-semi-module finiment engendré. La fonction qui associe à chaque polynôme d'expressions P, la série rationnelle $|P|$ est un morphisme d'actions. Le théorème 6.1 implique donc que l'orbite d'une série rationnelle sous l'action de A^* appartient elle aussi à un $\mathbb{K}$-semi-module finiment engendré, ce qui donne une nouvelle preuve d'un résultat classique [8]. Soulignons que ce résultat n'est pas le but que nous poursuivons. Il s'agit avant tout de trouver un moyen effectif, en manipulant des expressions, de construire un automate.

— ◦ —

4 L'automate des termes dérivés

A toute expression avec multiplicité E de $\mathbb{K}\mathsf{RExp}A$, on associe un $\mathbb{K}$-automate de la façon suivante :

Définition 6.7 *Soit* $P = \{\mathsf{K}_1, \mathsf{K}_2, \ldots, \mathsf{K}_n\}$ *l'ensemble des termes dérivés de* E. *Soit* $\mathsf{K}_0 =$

*E et P_{E} l'union de P et de $\{\mathsf{K}_0\}$. L'**automate des termes dérivés** de E est le $\mathbb{K}$-automate $\mathcal{A}_{\mathsf{E}} = \langle P_{\mathsf{E}},A,Z,I,T \rangle$ défini par :*

$$I_{\mathsf{K}_i} = \begin{cases} 1_{\mathbb{K}} & si \quad \mathsf{K}_i = \mathsf{K}_0 \\ 0_{\mathbb{K}} & sinon \end{cases}, \qquad Z_{\mathsf{K}_i,\mathsf{K}_j} = \bigoplus_{a\in A} z_{i,j}^{(a)}\, a\,, \qquad T_{\mathsf{K}_j} = \mathsf{c}(\mathsf{K}_j),$$

où les $z_{i,j}^{(a)}$ sont définis dans l'énoncé de la proposition 6.3.

Théorème 6.3 *Soit E une expression de $\mathbb{K}\mathsf{RExp}A$. La série réalisée par l'automate des termes dérivés de E est égale à la série dénotée par E.*

$$|\mathcal{A}_{\mathsf{E}}| = |\mathsf{E}|$$

Démonstration. La définition du $\mathbb{K}$-automate $\mathcal{A}_{\mathsf{E}}$ est en effet équivalente à celle d'une d'une $\mathbb{K}$-représentation linéaire (I,ζ,T). I et T sont vus respectivement comme des vecteurs ligne et colonne indicés par P_{E}, alors que ζ est un morphisme de A^* dans la semi-algèbre des matrices indicées par P_{E} défini par $(a)\zeta_{\mathbb{K}_i,\mathbb{K}_j} = (z_{i,j}^{(a)})$, pour tout a dans A.
La série réalisée par la représentation et par l'automate est :

$$|\mathcal{A}_{\mathsf{E}}| = \bigoplus_{f\in A^*} (I\cdot (f)\zeta \cdot T)\, f.$$

D'après l'équation 17, on obtient par récurence sur la longueur de f que

$$\forall f \in A^+\,, \quad \forall i \in [n] \qquad (I\cdot (f)\zeta)_i = k_i^{(f)},$$

d'où, pour tout mot f de A^+, d'après le théorème 6.3,

$$\langle |\mathcal{A}_{\mathsf{E}}|,f\rangle = \bigoplus_{i[n]} k_i^{(f)}\ \mathsf{c}(\mathsf{K}_i) = \langle |\mathsf{E}|,f\rangle.$$

Il suffit enfin de vérifier qu'on a une égalité similaire pour le mot vide :

$$\langle |\mathcal{A}_{\mathsf{E}}|,1_{A^*}\rangle = \mathsf{c}(\mathsf{K}_0) = \mathsf{c}(\mathsf{E}).$$

□

Le passé et le futur d'un état de l'automate des termes dérivés dépendent directement du terme dérivé auquel cet état correspond.

Proposition 6.4 *Soit E une expression de $\mathbb{K}\mathsf{RExp}A$ et K un terme dérivé de E. Le passé et le futur de l'état K de l'automate $\mathcal{A}_{\mathsf{E}}$ des termes dérivés de E vérifient les égalités suivantes :*

$$\forall u \in A^* \qquad \langle \mathsf{Past}_{\mathcal{A}_{\mathsf{E}}}(\mathsf{K}),u\rangle = \langle \frac{\partial}{\partial u}\,\mathsf{E},\mathsf{K}\rangle$$
$$\mathsf{Fut}_{\mathcal{A}_{\mathsf{E}}}(\mathsf{K}) = |\mathsf{K}|.$$

Démonstration. La preuve de la première égalité est par récurrence sur la longueur de u. Le résultat est vrai pour le mot vide, puisque chacun de ses membres est nul, sauf si $\mathsf{K} = \mathsf{E}$, auquel cas, chacun des membres vaut $1_{\mathbb{K}}$. On suppose le résultat vrai pour u. Alors, pour tout a dans A, pour tout terme dérivé K_j, d'après l'égalité (17),

$$\begin{aligned}
\langle \tfrac{\partial}{\partial ua}\,\mathsf{E},\mathsf{K}_j\rangle &= \bigoplus_{i\in[1;n]} \left[k_i^{(u)} \otimes z_{i,j}^{(a)}\right] \\
&= \bigoplus_{i\in[1;n]} \langle \frac{\partial}{\partial u}\,\mathsf{E},\mathsf{K}_j\rangle \otimes z_{i,j}^{(a)} \\
&= \bigoplus_{i\in[1;n]} \langle \mathsf{Past}_{\mathcal{A}_E d}(\mathsf{K}_i),u\rangle \otimes z_{i,j}^{(a)} \\
&= \bigoplus_{i\in[1;n]} \langle \mathsf{Past}_{\mathcal{A}_E d}(\mathsf{K}_i) Z_{\mathsf{K}_i,\mathsf{K}_j},ua\rangle \\
&\langle \mathsf{Past}_{\mathcal{A}_\mathsf{E}}(\mathsf{K}_j),ua\rangle
\end{aligned}$$

L'autre égalité se démontre de la même façon que le théorème 6.3. Il suffit de prendre comme expression de départ non plus E mais K. On obtient un automate $\mathcal{A}_\mathsf{K}$ qui réalise la série $|\mathsf{K}|$ et qui est identique à l'automate $\mathcal{A}_\mathsf{E}$ en aval de l'état K. Le comportement de $\mathcal{A}_\mathsf{K}$ est donc exactement le futur de K dans $\mathcal{A}_\mathsf{E}$. □

— ∘ —

5 Les termes dérivés fantômes

La proposition 6.4 nous indique que le passé d'un état de l'automate des termes dérivés est lié à l'apparition de ce terme dans les dérivées de l'expression de départ par rapport aux mots de A^*. Comme nous l'avons déjà dit, il est possible que certains termes dérivés n'apparaissent dans aucune dérivée de E. Ce sont les termes dérivés fantômes.

D'après la proposition 6.4, le passé d'un état de l'automate des termes dérivés qui correspond à un terme fantôme est nul. On peut donc supprimer ces états sans modifier le comportement de l'automate. Encore faut-il les identifier !

Semi-anneaux positifs

Définition 6.8 *Un semi-anneau est* ***positif*** *si aucun n'élément n'a d'opposé (pour l'addition) et s'il est intègre, c'est-à-dire que le produit de deux éléments non nuls est non nul.*

De nombreux semi-anneaux courants sont positifs. Citons, bien sûr, le semi-anneau des entiers naturels, mais aussi tous les semi-anneaux $(\mathsf{max},+)$ ou $(\mathsf{min},+)$ construits à partir de sous-ensembles de $\mathbb{R}$. Le semi-anneau $\mathcal{P}(A^*)$ formé à partir du monoïde libre est aussi un semi-anneau positif.

PROPOSITION 6.5 *Aucun terme dérivé d'une série formelle s sur A^*, à coefficient dans un semi-anneau positif, n'est un terme dérivé fantôme.*

Démonstration. A partir de l'ágalité (17) page 163, on montre immédiatement par récurrence sur la longueur des mots, que, si u est un mot de A^+ longueur m, alors, pour tout i dans $[1;n]$, on a :

$$k_i^{(u)} = \bigoplus_{i_1,\dots,i_{m-1}\in[1;n]} k_{i_1}^{(u_1)} \otimes z_{i_1,i_2}^{(u_2)} \otimes z_{i_2,i_3}^{(u_3)} \otimes \dots \otimes z_{i_{m-1},i}^{(u_m)}. \tag{19}$$

Si un terme dérivé K existe, c'est qu'il existe une suite $\mathsf{K}_{i_0} = \mathsf{E}, \mathsf{K}_{i_1}, \dots, \mathsf{K}_{i_m} = \mathsf{K}$ de termes dérivés et un mot $u_1u_2\dots u_m$ tels que, pour tout j de $[1;m]$, $\langle \frac{\partial}{\partial u_j} \mathsf{K}_{i_{j-1}}, \mathsf{K}_{i_j} \rangle = z_{i_{j-1},i_j}^{(u_j)} \neq 0_{\mathbb{K}}$. Il y a donc, dans l'égalité (19), un produit de facteurs non nuls; comme le semi-anneau est positif, ce produit n'est pas nul et la somme dont il fait partie non plus, donc le terme dérivé K apparaît dans la dérivée de E par rapport à u. □

Comme le semi-anneau de Boole est un semi-anneau positif, il n'y a pas de termes dérivés fantômes dans le cadre des dérivations d'expressions sans multiplicité.

Semi-anneaux plongeables dans un corps. Dans ce cas, on peut voir apparaître des termes dérivés fantômes. C'est d'ailleurs ce qu'illustre l'exemple 26.3. On peut toutefois simplifier l'automate si des termes dérivés fantômes apparaissent. On utilise pour cela le « théorème d'égalité » (*cf.* [23] page 143) pour tester si le passé d'un état est nul.

Autres semi-anneaux. Dans le cas général, la décision dépend non seulement de la taille et de la structure de l'automate, mais aussi de la struture du semi-anneau.

EXEMPLE 28 Considérons l'expression rationnelle $\mathsf{E}_5 = (ya)\cdot(xb)^\star\cdot(ya)$ sur l'alphabet $\{a,b\}$ à coefficients dans un semi-anneau contenant deux éléments x et y. La dérivation de cette expression donne les termes dérivés $(xb)^\star\cdot(ya)$ et 1. L'automate qui en résulte est dessiné figure 4. On peut se demander si 1 est un terme dérivé fantôme. Sans autre rensei-

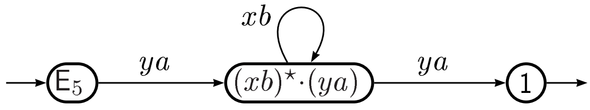


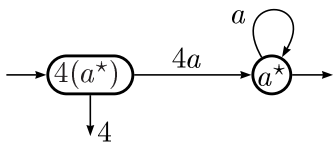
FIG. 4 – *L'automate des termes dérivés de* E_5.

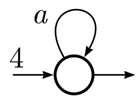
gnement sur le semi-anneau, cette question est indécidable. En effet, pour tout k dans $\mathbb{N}$, la dérivée de E_5 par rapport à ab^ka est égale à $y \otimes x^k \otimes y\,1$. Et pour tout N, il existe un semi-anneau et des éléments x et y tels que le plus petit k pour lequel cette quantité n'est pas nulle est N. (On peut par exemple prendre le semi-anneau des relations sur $[1; N+1]$, x égal à la rotation $(1,2,\dots,N+1)$ et y à la fonction partielle qui envoie 1 sur 2.)

— ∘ —

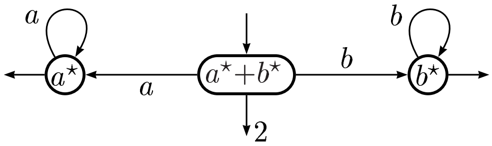
6 Variations

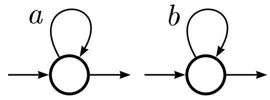
On peut imaginer un certain nombres de variations dans la façon dont est définie la dérivation. On peut par exemple regretter que l'expression $4(a^\star)$ soit représentée par le premier de ces deux automates et non par le second :





De même pour l'expression $a^\star + b^\star$:





Nous allons voir que le comportement de l'algorithme, c'est-à-dire la portée de la dérivation, est très fortement conditionné par la dérivation par rapport au mot vide. Nous avons dit dans la définition 6.6 que la dérivation d'une expression par rapport au mot vide était l'identité. C'est cette convention que nous allons amender dans les paragraphes suivants.

Termes dérivés unitaires. Dans un premier temps, il apparaît souhaitable d'extraire le plus rapidement possible les coefficients de l'expression, par exemple lorsque l'expression a un coefficient principal gauche non trivial (c'est-à-dire lorsqu'il y a un coefficient gauche dans la racine de l'arbre de l'expression qui est différent de $1_{\mathbb{K}}$).

DÉFINITION 6.9 *Une expression est **unitaire** (gauche) si son coefficient principal à gauche est égal à* $1_{\mathbb{K}}$. *Toute expression* E *de* $\mathbb{K}\mathsf{RExp}A$ *peut s'écrire* $(k\,\mathsf{F})$, *où* k *est un élément de* $\mathbb{K}$ *et* F *est une expression unitaire gauche. Pour toute expression unitaire* F, *tout élément* k *de* $\mathbb{K}$, *on définit la **dérivée unitaire** de* $(k\,\mathsf{F})$ *par rapport au mot vide par :*

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} 1_{A^*}}(k\,\mathsf{F}) = k\,\mathsf{F}.$$

Par suite, la dérivée unitaire d'une expression E *par rapport à une lettre* a *est définie par :*

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\,\mathsf{E} = \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} 1_{A^*}}\left(\frac{\partial}{\partial a}\,\mathsf{E}\right),$$

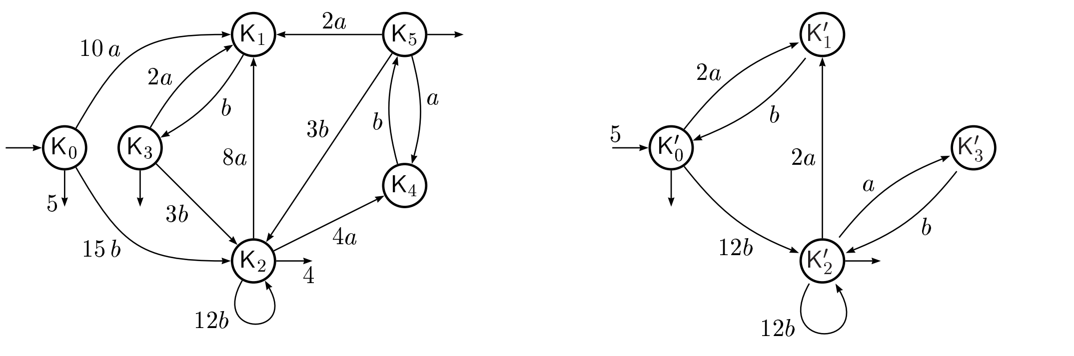
et en remplaçant chaque dérivation par une dérivation unitaire dans les formules de la définition 6.5.

Les termes dérivés obtenus par une telle dérivation sont unitaires gauches.

On peut montrer que, de même que pour la dérivation « simple », le nombre de termes dérivés unitaires est inférieur à la longueur de l'expression. La borne sur la taille de l'automate obtenu est donc respectée. Mieux, il existe une fonction surjective des termes dérivés sur les termes dérivés unitaires, l'automate des termes dérivés unitaires est donc plus petit que celui des termes dérivés.

Exemple 27.2 Calculons les termes dérivés (colonne de gauche) et les termes dérivés unitaires de $\mathsf{E}_3 = (5((2ab) + ((3b) \cdot (4(ab)^*)))^*)$:

$$
\begin{aligned}
\mathsf{K}_0 &= \mathsf{E}_3 & \mathsf{K}'_0 &= \mathsf{F}_3 \qquad k_0 = 5\\
\frac{\partial}{\partial a}\mathsf{K}_0 &= 10\,(b \cdot \mathsf{F}_3) = 10\,\mathsf{K}_1 & \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\mathsf{K}'_0 &= 2\,(b \cdot \mathsf{F}_3) = 2\,\mathsf{K}'_1\\
\frac{\partial}{\partial b}\mathsf{K}_0 &= 15\,((4\,(ab)^*) \cdot \mathsf{F}_3) = 15\,\mathsf{K}_2 & \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\mathsf{K}'_0 &= 12\,((ab)^* \cdot \mathsf{F}_3) = 12\,\mathsf{K}'_2\\
\frac{\partial}{\partial b}\mathsf{K}_1 &= \mathsf{F}_3 = \mathsf{K}_3 & \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\mathsf{K}'_1 &= \mathsf{F}_3 = \mathsf{K}'_0\\
\frac{\partial}{\partial a}\mathsf{K}_2 &= 4\,((b \cdot (ab)^*) \cdot \mathsf{F}_3) \oplus 8\,(b \cdot \mathsf{F}_3) & \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\mathsf{K}'_2 &= ((b \cdot (ab)^*) \cdot \mathsf{F}_3) \oplus 2\,(b \cdot \mathsf{F}_3)\\
&= 4\,\mathsf{K}_4 \oplus 8\,\mathsf{K}_1 & &= \mathsf{K}'_3 \oplus 2\,\mathsf{K}'_1\\
\frac{\partial}{\partial b}\mathsf{K}_2 &= 12\,((4\,(ab)^*) \cdot \mathsf{F}_3) = 12\,\mathsf{K}_2 & \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\mathsf{K}'_2 &= 12\,((ab)^* \cdot \mathsf{F}_3) = 12\,\mathsf{K}'_2\\
\frac{\partial}{\partial a}\mathsf{K}_3 &= 2\,(b \cdot \mathsf{F}_3) = 2\,\mathsf{K}_1 & &\\
\frac{\partial}{\partial b}\mathsf{K}_3 &= 3\,\mathsf{K}_2 & &\\
\frac{\partial}{\partial b}\mathsf{K}_4 &= ((ab)^* \cdot \mathsf{F}_3) = \mathsf{K}_5 & \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\mathsf{K}'_3 &= ((ab)^* \cdot \mathsf{F}_3) = \mathsf{K}'_2\\
\frac{\partial}{\partial a}\mathsf{K}_5 &= \mathsf{K}_4 \oplus 2\,\mathsf{K}_1, \quad \frac{\partial}{\partial b}\mathsf{K}_5 = 3\,\mathsf{K}_2 & &\\
\mathsf{c}(\mathsf{K}_0) &= 5, \mathsf{c}(\mathsf{K}_1) = \mathsf{c}(\mathsf{K}_4) = 0 & \mathsf{c}(\mathsf{K}'_0) &= \mathsf{c}(\mathsf{K}'_2) = 1\\
\mathsf{c}(\mathsf{K}_2) &= 4, \mathsf{c}(\mathsf{K}_3) = \mathsf{c}(\mathsf{K}_5) = 1 & \mathsf{c}(\mathsf{K}'_1) &= \mathsf{c}(\mathsf{K}'_3) = 0
\end{aligned}
$$



(a) L'automate des termes dérivés de E_3 (b) L'automate des termes dérivés unitaires

Fig. 5 – *Deux* $\mathbb{K}$*-automates pour* E_3

La figure 5 a) représente l'automate des termes derivés de E_3 (qui est isomorphe à l'automate de Glushkov calculé dans [11]); b) représente l'automate des termes dérivés unitaires de E_3.

Exemple 24.4 On calcule l'automate des termes unitaires de E_4. Comme E_4, n'est pas une expression unitaire, ce choix semble s'imposer. On pose $\mathsf{E}_4 = \mathrm{y}\,\mathsf{F}_4$.

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} 1_{A^*}}\,\mathsf{E}_4 = \mathrm{y}\,\mathsf{F}_4 = \mathrm{y}\,\mathsf{K}_0$$

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\,\mathsf{K}_0 = \mathrm{x}\,((((\mathrm{y}\,b)^\star\cdot a)\,\mathrm{y})\cdot\mathsf{F}_4) = \mathrm{x}\,\mathsf{K}_1 \qquad \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\,\mathsf{K}_0 = \mathrm{y}\,((((\mathrm{x}\,a)^\star\cdot b)\,\mathrm{x})\cdot\mathsf{F}_4) = \mathrm{y}\,\mathsf{K}_2$$

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\,\mathsf{K}_1 = \mathrm{y}\,\mathsf{F}_4 = \mathrm{y}\,\mathsf{K}_0 \qquad \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\,\mathsf{K}_2 = \mathrm{x}\,((((\mathrm{x}\,a)^\star\cdot b)\,\mathrm{x})\cdot\mathsf{F}_4) = \mathrm{x}\,\mathsf{K}_2$$

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\,\mathsf{K}_1 = \mathrm{y}\,((((\mathrm{y}\,b)^\star\cdot a)\,\mathrm{y})\cdot\mathsf{F}_4) = \mathrm{x}\,\mathsf{K}_1 \qquad \frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} b}\,\mathsf{K}_2 = \mathrm{x}\,\mathsf{F}_4 = \mathrm{x}\,\mathsf{K}_0$$

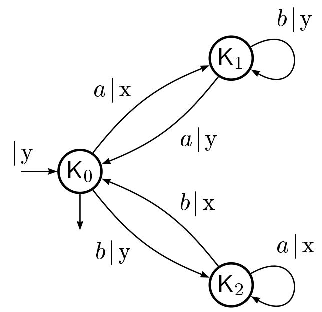


Fig. 6 – *Automate des termes unitaires de* E_4

— ∘ —

Dérivation cassante. Par ailleurs, on peut considérer que le symbole + d'une expression rationnelle est interprété directement lorsqu'il apparaît en dehors d'une expression « étoilée » ou d'un produit.

Définition 6.10 *Pour toutes expressions* E *et* F *de* $\mathbb{K}\mathsf{RExp}A$, *on définit la* ***dérivée cassante*** *de* $(\mathsf{E}+\mathsf{F})$ *par rapport au mot vide par :*

$$\frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} 1_{A^*}}(\mathsf{E}+\mathsf{F}) = \mathsf{E}\oplus\mathsf{F}.$$

Par suite, la dérivée cassante d'une expression E *par rapport à une lettre* a *est définie par :*

$$\frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} a}\,\mathsf{E} = \frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} 1_{A^*}}\left(\frac{\partial}{\partial a}\,\mathsf{E}\right),$$

et en remplaçant chaque dérivation par une dérivation unitaire dans les formules de la définition 6.5. De plus, on peut poser :

$$\frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} a}(\mathsf{E}\cdot\mathsf{F}) = \left(\frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} a}\,\mathsf{E}\cdot\frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} 1_{A^*}}\,\mathsf{F}\right)\oplus\mathsf{c}(\mathsf{E})\,\frac{\partial_{\mathrm{c}}}{\partial_{\mathrm{c}} a}\,\mathsf{F}.$$

Cette nouvelle façon de faire permet d'obtenir sur l'exemple $a^\star+b^\star$ déjà cité un résultat plus proche de l'intuition. Cependant, cette méthode ne permet pas de garantir que le nombre d'états de l'automate obtenu sera linéaire en la longueur de l'expression.

— ∘ —

7 Le cas commutatif

L'hypothèse de commutativité du semi-anneau $\mathbb{K}$ simplifie quelque peu les choses. En effet, dans ce cas, pour toute expression rationnelle E et tout élément k de $\mathbb{K}$, les expressions $(k\,\mathsf{E})$ et $(\mathsf{E}\,k)$ sont équivalentes. On peut donc supposer que la multiplication d'une expression par un coefficient s'effectue toujours à gauche. La définition des expressions est alors la suivante :

i) $\mathsf{0}$, $\mathsf{1}$, et a, pour tout a appartenant à A, sont des expressions rationnelles (atomiques).

ii) Si E est une expression rationnelle, et k un élément de $\mathbb{K}$, alors $(k\,\mathsf{E})$ est une expression rationnelle.

iii) Si E et F sont des expressions rationnelles, alors $(\mathsf{E}+\mathsf{F})$, $(\mathsf{E}\cdot\mathsf{F})$ et $(\mathsf{E}^\star)$ aussi.

Les identités triviales sur les expressions sont un tant soit peu plus simples :

$$\begin{gathered}(k\,\mathsf{0}) \equiv \mathsf{0}\,, \quad (0_{\mathbb{K}}\,\mathsf{E}) \equiv \mathsf{0}\,, \quad (\mathsf{0}\cdot\mathsf{E}) \equiv (\mathsf{E}\cdot\mathsf{0}) \equiv \mathsf{0}\,,\\ \mathsf{0}+\mathsf{E} \equiv \mathsf{E}+\mathsf{0} \equiv \mathsf{E}\,, \quad (1_{\mathbb{K}}\,\mathsf{E}) \equiv \mathsf{E}\,,\\ ((k\,\mathsf{1})\cdot\mathsf{E}) \equiv (\mathsf{E}\cdot(k\,\mathsf{1})) \equiv (k\,\mathsf{E})\,,\\ (k\,(k'\,\mathsf{E})) \equiv ([k\otimes k']\,\mathsf{E}).\end{gathered}$$

La représentation en arbre d'une expression est elle aussi simplifiée, puisqu'à chaque nœud n'est affecté qu'un coefficient. Les résultats, quant à eux, restent inchangés; la dérivation est définie de la même manière, si ce n'est qu'on ignore l'axiome $\frac{\partial}{\partial a}(\mathsf{E}\,k) = ([\frac{\partial}{\partial a}\,\mathsf{E}]\,k)$ qui n'a plus lieu d'être.

On peut définir la multiplication des monômes d'expressions de manière légèrement différente :

$$[k\,\mathsf{E}]\cdot[k'\,\mathsf{F}] = [k\otimes k']\,(\mathsf{E}\cdot\mathsf{F}).$$

Ceci permet, dans le cas de la dérivation unitaire, de définir la dérivée du produit :

$$\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}(\mathsf{E}\cdot\mathsf{F}) = \left(\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\,\mathsf{E}\cdot\,\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} 1_{A^*}}\,\mathsf{F}\right) \oplus \mathsf{c}(\mathsf{E})\;\frac{\partial_{\mathrm{u}}}{\partial_{\mathrm{u}} a}\,\mathsf{F}.$$

Cet aménagement n'aurait pas de sens dans le cas non commutatif car le coefficient qui peut « sortir » du second facteur serait immédiatement réenchâssé à cause de la multiplication des monômes.

— ∘ —

Annexe A

Le problème de Dedekind, posé en 1897, est généralement exprimé en termes de fonctions booléennes monotones, c'est-dire de fonctions réalisées uniquement avec les opérateurs logiques Et et Ou.

Le nombre de Dedekind $M(n)$ est le nombre de fonctions booléennes monotones à n variables distinctes.

On ne connaît pas de formule close pour exprimer $M(n)$. Les valeurs connues à ce jours [61] sont :

n	$M(n)$
0	2
1	3
2	6
3	20
4	168
5	7581
6	7828354
7	2414682040998
8	56130437228687557907788

Une fonction booléenne monotone est réalisée par une formule qu'on peut mettre sous forme disjonctive. Les constantes (Vrai et Faux) étant absorbantes ou élément neutre pour les opérateurs Et et Ou, les fonctions monotones qu'on peut réaliser par des formules non vides sont les fonctions monotones non constantes.

Chaque clause de la formule en forme normale disjonctive est formée d'un ensemble de littéraux; si les ensembles de littéraux de deux clauses sont inclus l'un dans l'autre, la clause ayant le plus grand de ces ensembles est inutile (si la plus petite est vraie, la formule aussi, si elle est fausse, la plus grande l'est aussi). Chaque formule est donc caractérisée par une anti-chaîne de sous-ensembles de l'ensemble des littéraux. On montre que deux anti-chaînes différentes correspondent à des fontions distinctes. Le nombre d'anti-chaînes (non vides) d'un ensemble à n éléments est donc $M(n) - 2$.

Annexe B

Nous présentons ici l'algorithme d'Hashiguchi tel qu'on peut le trouver dans [33]. Afin de simplifier les choses, certaines parties de l'algorithme, notamment la décision de la hauteur d'étoile relative nulle sont présentées comme des boîtes noires.

Tout d'abord, une petite définition :

DÉFINITION 6.11 *Soit $\mathcal{L}$ un langage rationnel sur A^* et $\mathcal{C}$ une famille (finie) de langages rationnels sur A^*. L'ensemble $\mathsf{RExp}\mathcal{C}$ des expressions rationnelles sur $\mathcal{C}$ est défini de façon similaire à $\mathsf{RExp}A$. En revanche, l'interprétation d'une expression de $\mathsf{RExp}\mathcal{C}$ est induite par l'interprétation d'un élément $\mathcal{L}_i$ de $\mathcal{C}$ qui est le langage $\mathcal{L}_i$ sur A^*. La* ***hauteur d'étoile relative*** *de $\mathcal{L}$ par rapport à $\mathcal{C}$, noté $\mathsf{h}_\mathcal{C}(\mathcal{L})$, est la hauteur d'étoile minimale des expressions de $\mathsf{RExp}\mathcal{C}$ qui représentent $\mathcal{L}$. Si $\mathcal{L}$ n'appartient pas à $\mathrm{Rat}\,\mathcal{C}^*$, alors $\mathsf{h}_\mathcal{C}(\mathcal{L}) = \infty$.*

Par exemple, la hauteur d'étoile relative de $\mathcal{L}$ par rapport à $\{\mathcal{L}\}$ est nulle.

Présentons tout d'abord l'algorithme pour le calcul de la hauteur d'étoile relative :

Soit $\mathcal{A}$ un automate minimal reconnaissant un langage $\mathcal{L}$ et Q l'ensemble des états de $\mathcal{L}$. Soit M le monoïde syntaxique de $\mathcal{L}$. Soit $\mathcal{C} = \{\mathcal{L}_i \mid i \in I\}$ une famille finie de langages. Chaque langage $\mathcal{L}_i$ est reconnu par un automate $\mathcal{B}_i$ dont les états forment un ensemble Q_i.

Si i est l'état initial de $\mathcal{A}$, on définit $\mathcal{A}_\mathcal{C}$ l'automate déterministe sur l'alphabet $\mathcal{C}$ qui représente l'orbite de $\{i\}$ sous l'action des éléments de $\mathcal{C}$. (Les éléments de $\mathcal{C}$ agissent canoniquement sur les sous-ensembles d'états de $\mathcal{A}$.)

On définit une cascade de constantes :

$$\begin{aligned}
g_1 =& (4^{\mathsf{Card}(M)} + 1)(\mathsf{enl}(\mathcal{A}_\mathcal{C}) + 1) \\
g_2 =& \left(\frac{g_1}{(\mathsf{enl}(\mathcal{A}_\mathcal{C}) + 1)}\right)\left(\sum_{i \in I} \mathsf{Card}(Q_i)\right) \\
g_3 =& 16^{g_2^2} \\
g_4 =& (g_3(\mathsf{Card}(Q) + 1)^{\mathsf{Card}(Q)}\,\mathsf{Card}(M))^2 \\
g_5 =& (g_4 + \mathsf{Card}(Q) + 2)^{4\,\mathsf{Card}(Q)}
\end{aligned}$$

L'algorithme pour le calcul de la hauteur d'étoile relative est le suivant :

1. Vérifier si $\mathsf{h}_\mathcal{C}(\mathcal{L}) = \infty$.
2. Vérifier si $\mathsf{h}_\mathcal{C}(\mathcal{L}) = 0$.

3. Calculer g_5 et former

$$\mathcal{C}' = \mathcal{C} \cup \bigcup \{|(W_1 + W_2 \ldots + W_n)^\star| \mid n \geqslant 1\},$$

où chaque W_i est un mot non vide de longueur inférieure à g_5 sur l'alphabet $\mathcal{C}$. Alors $\mathsf{h}_{\mathcal{C}}(\mathcal{L}) = \mathsf{h}_{\mathcal{C}'}(\mathcal{L}) + 1$.

En fait, pour le calcul de la hauteur d'étoile (absolue), on verra que le point 1 est inutile. L'algorithme de calcul de la hauteur d'étoile s'appuie sur le précédent. Il est initié par :

1. Si $\mathcal{L}$ est un langage fini, $\mathsf{h}(\mathcal{L}) = 0$.
2. Soit $g = 16n(n+2)(\mathsf{enl}(\mathcal{A})n(n+2)+1)$, avec $n = \mathsf{Card}(M)$. On pose

$$\mathcal{C} = \{\{a\} \mid a \in A\} \cup \bigcup \{|(w_1 + w_2 \ldots + w_n)^\star| \mid n \geqslant 1, w_i \in A^{\leqslant g}\}.$$

Calculer $\mathsf{h}(\mathcal{L}) = \mathsf{h}_{\mathcal{C}}(\mathcal{L}) + 1$.

Pour illustrer la complexité de cet algorithme, considérons le langage $\mathcal{L}_{r_1}$. Ce langage est un langage réversible. Nous avons calculé sa hauteur d'étoile (égale à 2) dans l'exemple 12 page 119. Le monoïde syntaxique de ce langage est aussi le monoïde de transition de l'automate présenté page 71, il compte trente-et-un éléments (toutes les fonctions injectives d'un ensemble de trois éléments dans lui-même, sauf les trois transpositions); l'enlacement de son automate minimal est égal à 2. On obtient les valeurs suivantes :

$$g = 33505296, \qquad \mathsf{Card}(A^{\leqslant g}) > 2^{33505296}, \qquad \mathsf{Card}(\mathcal{C}) > 2^{2^{33505296}}$$

On peut maintenant appliquer l'algorithme de calcul de la hauteur d'étoile relative. L'automate $\mathcal{A}_{\mathcal{C}}$ a au moins un enlacement égal à 2, puisque $\mathcal{A}$ en est un sous-automate.

$$g_2 > 2^{2^{33505296}}, \qquad g_5 > g_4 > g_3 = 16^{g_2^2} > 2^{2^{2^{33505296}}}$$

On obtient donc :

$$\mathsf{Card}(\mathcal{C}^{\leqslant g_5}) > \left(2^{2^{33505296}}\right)^{2^{2^{2^{33505296}}}}, \qquad \mathsf{Card}(\mathcal{C}') > 2^{\left(2^{2^{33505296}}\right)^{2^{2^{2^{33505296}}}}} > 2^{2^{2^{2^{2^{2^{2^{2^{2}}}}}}}}$$

Remarquons que dans la manipulation de ce type de nombres, le seul ordre de grandeur qui reste est la hauteur de la tour d'exponentielle et que les valeurs des nombres dans cette tour ont une signification qui décroît exponentiellement à mesure qu'on s'éloigne du haut. La dernière inégalité est par exemple une minoration brutale, (on minore 33505296 par 65536 et $\left(2^{2^{33505296}}\right)$ par 2), mais si on remplace le 2 du haut par 2,3, elle devient fausse. En revanche, on peut remplacer le 2 du bas par un milliard sans la contredire.

Pour donner un ordre de grandeur, le nombre estimé de particules de l'univers (10^{80}) ou le nombre de cycles qu'aurait effectué un processeur de fréquence 10 GHz depuis la naissance supposée de l'univers (5 $\overline{\mathrm{M}}$A) est très largement majoré par une tour de hauteur 5 (2^{65536}). Ces grandeurs sont négligeables devant une tour de hauteur 9, le rapport entre les deux est quasiment lui-même une tour de hauteur 9.

A ce stade de l'algorithme, on obtient alors (par une opération de complexité non négligeable) que $\mathsf{h}_{\mathcal{C}'}(\mathcal{L}) = 0$. La taille des ensembles manipulés est assez éloquente.

Table des exemples

Index

Bibliographie

[1] A. Ambainis et R. Freivalds, 1-way quantum finite automata: strengths, weaknesses and generalizations. In *39th Ann. Symposium on FOCS* (1998), 332–342.

[2] D. Angluin, Inference of reversible languages. *J. of the ACM* **29** (1982), 741–765.

[3] V. Antimirov, Partial derivatives of regular expressions and finite automaton constructions. *Theoret. Comput. Sci.* **155** (1996), 291–319.

[4] A. Arnold, A. Dicky, et M. Nivat, A note about minimal non-deterministic automata. *Bull ot the EATCS* **47** (1992), 166–169.

[5] M.-P. Béal, O. Carton, C. Prieur, et J. Sakarovitch, Squaring transducers. *Proc. of Latin 2000, Lect. Notes in Comp. Sci.* **1776** (2000), 397–406.

[6] G. Berry et R. Sethi, From regular expressions to deterministic automata, *Theoret. Comput. Sci.* **48** (1986), 117–126.

[7] J. Berstel et J.-E. Pin, Local languages and the Berry-Sethi algorithm, *Theoret. Comput. Sci.* **155** (1996), 439–446.

[8] J. Berstel et Ch. Reutenauer, *Les séries rationnelles et leurs langages.* Masson, 1984. Traduction: *Rational Series and their Languages.* Springer, 1986.

[9] J. A. Brzozowski, Derivatives of regular expressions. *J. Assoc. Comput. Mach.* **11** (1964), 481–494.

[10] A. Buchsbaum, R. Giancarlo, et J. Westbrook, On the Determinization of Weighted Finite Automata. *Proc. of ICALP'98, Lect. Notes in Comp. Sci.* **1443** (1998), 482–493.

[11] P. Caron et M. Flouret, Glushkov construction for multiplicities. *Pre-Proceedings of CIAA'00*, M. Daley, M. Eramian and S. Yu, eds, Univ. of Western Ontario, (2000), 52–61.

[12] O. Carton, Factorisations et matrice des facteurs. Manuscrit (1994).

[13] J.-M. Champarnaud et D. Ziadi, New finite automaton constructions based on canonical derivatives. *Pre-Proceedings of CIAA'00*, M. Daley, M. Eramian and S. Yu, eds, Univ. of Western Ontario, (2000), 36–43.

[14] Ch. Choffrut, Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoret. Comput. Sci.* **5** (1977), 325–337.

[15] M. Chrobak, Finite Automata and Unary Languages. *Theoret. Comput. Sci.* **47** (1986), 149–158.

[16] G. Cohen, P. Moller, J.-P. Quadrat, et M. Viot, Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proc.: Special issue on D.E.S.* **77.1** (1989).

[17] R. Cohen, Star height of certain families of regular events. *J. Computer System Sci.* **4** (1970), 281–297.

[18] R. Cohen et J. A. Brzozowski, General properties of star height of regular events. *J. Computer System Sci.* **4** (1970), 260–280.

[19] J. H. Conway, *Regular algebra and finite machines.* Chapman and Hall, 1971.

[20] B. Courcelle, D. Niwinski, A. Podelski, A geometrical view of the dererminization ond minimization of finite-state automata. *Math. Systems Theory* **24** (1991), 117–146.

[21] F. Dejean et M.-P. Schützenberger, On a question of Eggan. *Inform. and Control* **9** (1966), 23–25.

[22] L. C. Eggan, Transition graphs and the star-height of regular events. *Michigan Mathematical J.* **10** (1963), 385–397.

[23] S. Eilenberg, *Automata, Languages and Machines, volume A.* Academic Press, 1974.

[24] S. Gaubert, Rational Series over Dioids and Discrete Event Systems. *Proc. of the 11th Conf. on Anal. and Opt. of Systems, Lect. Notes in Contr. and Inf. Sci.* **199** (1994).

[25] S. Gaubert, On the Burnside problem for Semigroups of Matrices in the $(\max,+)$ Algebra. *Semigroup Forum* **52** (1996), 271–292.

[26] S. Gaubert, Methods and applications of $(\max,+)$ linear algebra. *rapport de recherche INRIA* **3088** (1997).

[27] V. Glushkov, The abstract theory of automata. *Russian Mathematical Surveys 16* (1961), 1–53.

[28] Ch. Hagenah et A. Muscholl, Computing ε-Free NFA from regular expressions in $O(n\log^2(n))$ time. *R.A.I.R.O. Inf. Théorique* **34**, (2000), 257–277.

[29] T.E. Hall, Biprefix codes, inverse semigroups and syntactic monoids of injective automata. *Theoret. Comput. Sci.* **32** (1984), 201–213.

[30] T. Harju et J. Karhumäki, The equivalence problem of multitape finite automata. *Theoret. Comput. Sci.* **78** (1991), 347–355.

[31] K. Hashiguchi et N. Honda, The star height of reset-free events and strictly locally testable events. *Inform. and Control* **40** (1979), 267–284.

[32] K. Hashiguchi, Limitedness theorem on finite automata with distance functions. *J. of Comput. Syst. Sci.* **24** (1982), 233–244.

[33] K. Hashiguchi, Algorithms for determining relative star height and star height. *Inform. and Computation* **78** (1988), 124–169.

[34] K. Hashiguchi, Improved limitedness theorem on finite automata with distance functions. *Theoret. Comput. Sci.* **72** (1990), 27–38.

[35] P.-C. Héam, A lower bound for reversible automata. *Theoret. Informatics Appl.* **34** (2000), 331–341.

[36] P.-C. Héam, Contribution à l'algorithmique des automates : complexité et aspects topologiques. *Thèse de doctorat*, Université Paris 7, 2001.

[37] J. Hromkovič, S. Seibert, et T. Wilke, Translating regular expressions into small ε-free nondeterministic finite automata. *Proc. of STACS'97, Lect. Notes in Comp. Sci.* **1200** (1997), 55–66.

[38] D. Krob, Differentiation of K-rational expressions. *Int. J. of Algebra and Computation 2* (1992), 57–87.

[39] W. Kuich et A. Salomaa, *Semirings, Automata, Languages.* Springer, 1986.

[40] G. Lallement, *Semigroups and combinatorial applications.* Wiley, 1979.

[41] H. Leung, Limitedness theorem on finite automata with distance functions : an algebraic proof *Theoret. Comput. Sci.* **81** (1991), 137–145.

[42] S. Lombardy et J. Sakarovitch, On the star height of rational languages, a new presentation for two old results *Proc. of 3rd ICLWC, Kyoto* (2000) (M. Ito, ed.), World Scientific, *à paraître.*

[43] S. Lombardy et J. Sakarovitch, Star height of reversible languages and universal automata, *accepté à Latin'02.*

[44] S. Lombardy, Sequentialization and unambiguity of (max, +) rational series over one letter. *Pre-Proc. of Workshop on Max-plus algebra, Prague* (2001) (S. Gaubert, ed.).

[45] O. Matz et A. Potthoff, Computing small nondeterministic finite automata. *proc. of TACAS'95, BRICS Notes Series* (1995), 74–88.

[46] R. McNaughton et H. Yamada, Regular expressions and state graphs for automata. *IRE Trans. on electronic computers* **9** (1960), 39–47.

[47] R. McNaughton, The loop complexity of pure-group events. *Inform. and Control* **11** (1967), 167–176.

[48] Moebius, *Sur l'étoile.* Gentiane, 1983, Rééd. Casterman, 1990.

[49] M. Mohri, Finite-State Transducers in Language and Speech Processing. *Computat. Ling.* **23.2** (1997), 269–311.

[50] C. Nicaud, Étude du comportement en moyenne des automates finis et des langages rationnels. *Thèse de doctorat*, Université Paris 7, 2000.

[51] J.-E. Pin, On reversible automata. In *Proc. 1st LATIN Conf., (I. Simon, Ed.), Lecture Notes in Comput. Sci.* **583** (1992), 401–416.

[52] J.-E. Pin, Variétés de langages formels. Masson, 1984. Traduction: *Varieties of formal languages* North Oxford Acad. Pub., 1986.

[53] G. Raney, Sequential functions. *J. Assoc. Comput. Mach.* **5** (1958) 177–180.

[54] J. Sakarovitch, A construction on automata that has remained hidden. *Theoret. Comput. Sci.* **204** (1998), 205–231.

[55] J. Sakarovitch, *Éléments de théorie des automates.* Vuibert, *à paraître.*

[56] A. Salomaa, *Jewels of formal language theory.* Computer Science Press, 1981.

[57] P.V. Silva, On free inverse monoid languages. *Theoret. Informatics and Appl.* **30** (1996), 349–378.

[58] I. Simon, Recognizable sets with multiplicities in the tropical semiring. *Proc. of MFCS'88, Lect. Notes in Comp. Sci.* **324** (1988), 107–120.

[59] I. Simon, The non deterministic complexity of a finite automaton. *Mots (M. Lothaire)*, Hermès (1990) 384–400.

[60] I. Simon, On semigroups of matrices over the tropical semiring. *R.A.I.R.O. Inf. Théorique*, (1994), 277–294.

[61] N.J.A. Sloane, *The On-Line Encyclopedia of Integer Sequences.* `http://www.research.att.com/~njas/sequences/`

[62] J.B. Stephen, Presentations of inverse monoids. *J. Pure Appl. Alg.* **63** (1990), 81–112.

[63] M. Szalay, On the maximal order in S_n and S_n^*. *Acta arithm.* **37** (1980), 321–331.