

Expressivité des automates pondérés circulaires et boustrophédons

Louis-Marie Dando

▶ To cite this version:

Louis-Marie Dando. Expressivité des automates pondérés circulaires et boustrophédons. Langage de programmation [cs.PL]. Université de Bordeaux, 2019. Français. NNT: 2019BORD0130 . tel-02350722

HAL Id: tel-02350722 https://theses.hal.science/tel-02350722

Submitted on 6 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.





THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par Louis-Marie DANDO

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Expressivité des automates pondérés circulaires et boustrophédons

Date de soutenance : 09 septembre 2019

Devant le jury composé de :

Prof. Jean-Marc Talbot,	Université Aix-Marseille	Président du jury
Prof. Pascal Caron,	Université de Rouen-Normandie	Rapporteur
Prof. Olivier Carton,	Université Paris Diderot	Rapporteur
Prof. Marie-Pierre BÉAL,	Université Paris-Est Marne-La-Vallée	Examinatrice
Prof. Anca Muscholl,	Université de Bordeaux	Examinatrice
Prof. Sylvain Lombardy,	Institut Polytechnique de Bordeaux .	Directeur

A.M.D.G

Résumé Cette thèse porte sur certaines extensions des automates pondérés, et étudie les séries qu'ils réalisent en fonction de la nature des poids.

Ces extensions se distinguent par les mouvements supplémentaires autorisés à la tête de lecture de l'automate. retour au début du mot pour les automates circulaires, changement de sens de lecture pour les automates boustrophédons. Dans le cas général, les automates pondérés circulaires sont plus puissants que les automates unidirectionnels classiques, et moins puissants que les boustrophédons.

On introduit de plus les expressions de Hadamard, qui sont une extension des expressions rationnelles et qui permettent de dénoter le comportement des automates circulaires. Les aspects algorithmiques de cette conversion sont étudiés dans le cas où les poids appartiennent à un semi-anneau rationnellement additif.

On montre que lorsque les poids sont des nombres rationnels, réels ou complexes, les automates circulaires sont aussi expressifs que les boustrophédons.

Enfin, si les poids forment un bi-monoïde localement fini, les automates boustrophédons ne sont pas plus expressifs que les automates pondérés classiques.

Mots-clés Automates pondérés, automates circulaires, séries rationnelles, séries de Hadamard, automates boustrophédons

Laboratoire d'accueil Laboratoire Bordelais de Recherche en Informatique

iv L.-M. DANDO

Remerciements

J'aimerais avant tout remercier très chaleureusement Sylvain Lombardy pour son encadrement tout au long de ces quatres années de thèse, pour sa patience infinie et ses conseils éclairés.

Je remercie Pascal Caron et Olivier Carton, qui ont bien voulu rapporter cette thèse. Merci également à Marie-Pierre Béal, Anca Muscholl et Jean-Marc Talbot, qui ont accepté de faire partie du jury.

J'aimerais également remercier tout particulièrement Géraud Sénizergues et Irène Durand qui ont accompagné mes premiers balbutiements dans la recherche lors de mon stage. Un grand merci également à mes professeurs de l'ENSEIRB-MatMeca, notamment Frédéric Herbreteau et David Renault qui à des titres divers ¹ m'ont accompagné durant ma scolarité bordelaise.

Je remercie mes co-bureaux Luis, Abdelhamid, Louis, et Truong, qui ont su me supporter quasiment tous les jours : bravo! Je remercie également Mohammed et Nathan pour leur amitié et la dénomination "blagues louis-marrantes" ². Mes remerciements vont aussi à Thibaut, Vassilis, Elsa, et tous les matheux de l'IMB. Je tiens également à remercier le LaBRI et tout son personnel, qui ont permis que ma thèse se déroule dans d'excellentes conditions matérielles.

J'aimerais également remercier pêle-mêle MM. Tolkien, Watterson, Pratchett, Carol, Lewis, Adams, Jordan, Sanderson, Astier, Whedon, Saki, Eddings, pour ce que leurs œuvres m'ont apporté.

Cette liste ne serait pas complète ³ si je ne mentionnais pas les pailloux, les doms, Thomas et Élisabeth (et Aurore), Antoine, Maider, Laude, Marie, Constance, Valentin, Marie-Marthe, Louise, les collègues de l'ENSEIRB et de la fac, ainsi que tous ceux qui ont rendu mon séjour à Bordeaux si agréable.

Enfin, je profite de ces quelques lignes pour embrasser toute ma famille. Papa, Maman, Gabriel et Margaux, Joseph, Henri, je vous remercie en dernier, vous qui n'êtes certes pas les derniers dans mon esprit ⁵: merci pour tout.

Bonne lecture à vous tous.

^{1.} Ou des tubes de l'été...

^{2.} Et la possibilité de rentrer dans un bar avec eux pour la blague.

^{3.} Elle aurait pu l'être, mais il ne faut pas dégouter les lecteurs avant même le début de la thèse avec des remerciements interminables. 4

^{4.} Et encore moins en abusant des notes de pied de page.

^{5.} Point bonus pour celui qui retrouve l'origine de cette citation.

vi L.-M. DANDO

Sommaire

			Page
In	${f trod}$	uction	1
1	Éle	ments fondamentaux	5
	1.1	Structures algébriques	. 5
	1.2	Somme infinie et étoile	. 7
	1.3	Ordre d'un monoïde	. 11
	1.4	Séries	. 12
	1.5	Expressions	. 14
2	Aut	tomates	17
	2.1	Automates unidirectionnels	. 17
	2.2	Extensions du modèle	. 20
	2.3	Constructions	. 28
3	Aut	tomates circulaires sur les semi-anneaux rationnellemen	nt
	add	litifs	37
	3.1	Équivalence séries de Hadamard - Automates circulaires	. 38
	3.2	Aspects algorithmiques: notions nécessaires	. 43
	3.3	Des expressions de Hadamard aux automates circulaires pondé-	
		rés	. 50
	3.4	Et vice-versa	
	3.5	Extension à \mathbb{Q} , \mathbb{R} et \mathbb{C}	. 71
4	Aut	tomates boustrophédons sur les corps commutatifs	83
	4.1	Construction de l'automate fortement valide	. 84
	4.2	D'un automate boustrophédon fortement valide à un automate	
		circulaire	
	4.3	2 -représentation sur \mathbb{Q} , \mathbb{R} , ou \mathbb{C}	. 111
5		tomates boustrophédons sur les semi-anneaux et les b	
		noïdes localement finis	117
	5.1	Les coupes	
	5.2	L'automate des coupes	. 121

SO	Λ	Λħ	1	1	ID	L
יור.	//\	// //	∕∎.	$\boldsymbol{\mathcal{A}}$	ın	г

5.3	L'équivalence boustrophédon-unidirectionnel	. 126
Index		135
Bibliog	graphie	137

viii L.-M. DANDO

Introduction

Cette thèse porte sur l'étude de l'expressivité en terme de séries de certaines familles d'automates pondérés en fonction de la structure des poids.

Parmi les outils utilisés pour la modélisation mathématique, le modèle des machines de calcul universelles dû à Turing est un des plus connus, et des plus étudiés. Ce modèle est très puissant, et permet de représenter de nombreux problèmes, notamment sur les langages. Cependant, la contrepartie est que de nombreux problèmes de décision sur ce type de machine sont indécidables.

Des machines plus simples ont été introduites, dont les propriétés peuvent être vérifiées algorithmiquement, c'est-à-dire automatiquement, en suivant toujours le même processus. Cependant les langages que ces machines peuvent traiter sont plus restreints.

On peut citer parmi ces machines le modèle des automates finis, modèle très étudié en raison – entre autres – de sa simplicité, et dont de très nombreuses propriétés sont connues :

la machine lit un mot en entrée, de gauche à droite, et ensuite prend une décision, soit d'accepter le mot, soit de le rejeter. Il est important de noter que cette machine ne modifie pas le mot d'entrée, contrairement aux machines de Turing.

Dans ce mémoire, nous allons étudier des modèles intermédiaires entre les automates et les machines de Turing, qui sont les automates circulaires et les automates boustrophédons. Ces modèles ne modifient pas non plus le mot d'entrée, mais peuvent le lire autant de fois que souhaité dans le cas des automates circulaires, voire changer de sens de lecture du mot pendant cette lecture dans le cas des automates boustrophédons.

Il a été prouvé que dans le cas où les automates ne sont pas pondérés, tous ces modèles sont équivalents quant à leur puissance, cependant plus le modèle est complexe, plus les machines vont être succintes pour exprimer la même chose.

Sauf mention contraire, dans ce mémoire, tous les automates sont munis de poids : au lieu de décider simplement "oui" ou "non", ces machines vont renvoyer un élément d'une structure algébrique. Par exemple, dans le cas d'un automate probabiliste, le calcul donnera un réel compris entre 0 et 1. Ces modèles sont utilisés dans l'industrie, l'analyse d'image, etc.

Dans ce mémoire, on montre que dans les semi-anneaux rationnellement

additifs, on peut construire un automate circulaire à partir d'une expression de Hadamard, et vice-versa, ainsi que certains résultats pour cette équivalence sur \mathbb{Q} , \mathbb{R} et \mathbb{C} .

On montre de plus que sur les corps commutatifs, bien qu'ils soient strictement plus puissants que les automates classiques, les automates boustrophédons et les automates circulaires sont équivalents.

L'essentiel des techniques de preuve utilisées repose sur la modification d'automates afin d'assurer qu'ils vérifient une propriété donnée. Il s'agit donc de preuves constructives.

Éléments fondamentaux

Le but du premier chapitre est de présenter l'essentiel des outils algébriques dont nous allons avoir besoin tout au long de ce mémoire. Y seront présentés en particulier l'ensemble des structures sur les poids dont nous munirons les automates, les langages et les séries, ainsi que les expressions, rationnelles comme celles de Hadamard.

Automates

Le deuxième chapitre sera consacré à la définition formelle des différents modèles d'automates étudiés au long de ce document : les automates unidirectionnels, le modèle classique, les automates circulaires, où l'on peut lire le mot autant de fois que nécessaire, les automates à navette, qui peuvent en plus lire le mot de droite à gauche, et les automates boustrophédons, qui peuvent changer de sens de lecture au cours du mot.

On y trouvera également quelques outils de manipulation des automates, permettant de transformer un automate tout en garantissant l'invariance de son comportement.

Automates circulaires sur les semi-anneaux rationnellement additifs

Le chapitre suivant étudie les liens entre expressions de Hadamard et automates circulaires : dans le cas des semi-anneaux rationnellement additifs, ces deux modèles réalisent les mêmes séries, et on donne un algorithme pour passer d'un modèle à l'autre. Ce résultat, qui étend le lien entre expressions rationnelles et automates unidirectionnels, permet aussi d'étendre les algorithmes effectuant ces transformations.

De plus, on donne quelques résultats sur les corps courants \mathbb{Q} , \mathbb{R} , et \mathbb{C} , qui ne sont pas rationnellement additifs : on montre entre autres comment passer d'un automate circulaire à un quotient de Hadamard d'expressions rationnelles, et vice-versa.

Les automates boustrophédons sur les corps commutatifs

Dans ce chapitre, on étudie les automates boustrophédons lorsque les poids forment un corps commutatif. On montre que les séries réalisées par ces automates sont des quotients de séries rationnelles. On montre ainsi, grâce à un résultat du chapitre précédent, que sur \mathbb{Q} , \mathbb{R} , et \mathbb{C} , les automates boustrophédons et les automates circulaires sont équivalents. La preuve de ce résultat repose sur le calcul par automate de l'étoile d'une matrice, qui n'est valide que sur une sous-classe des automates boustrophédons, les automates fortement valides. C'est pourquoi on montre tout d'abord que pour tout automate boustrophédon valide, on peut construire un automate boustrophédon équivalent fortement valide.

Les automates boustrophédons sur les semi-anneaux et les bi-monoïdes localement finis

Le dernier chapitre porte sur l'étude des automates boustrophédons lorsque les poids forment un bi-monoïde localement fini. Il s'agit d'une extension de la preuve de Shepherdson reposant sur la même technique : l'analyse des coupes de l'automates. Ces objets sont des "tranches" des chemins de l'automate, et sont étudiés en début de ce chapitre. La suite est consacrée à la construction d'un automate unidirectionnel équivalent à un automate boustrophédon.

Chapitre 1

Élements fondamentaux

"Au commencement était le Verbe"

– Évangile selon saint Jean,I,1

Dans ce chapitre sont rappelées un certain nombre de définitions dont nous allons avoir besoin dans la suite. On y fixe un certain nombre de notations, mais le lecteur déjà au fait des notions abordées peut passer sans perte de compréhension au chapitre suivant.

Structures algébriques

Rappel sur les langages

Définition 1.1. Un *alphabet* est un ensemble fini A de symboles, que l'on appelle *lettres*. Un *mot* est une suite finie de lettres, et la longueur de ce mot la longueur de cette suite. Un *langage* est un ensemble de mots.

On notera la longueur d'un mot u par |u|, ε le mot de longueur 0, u_i la $i^{\grave{e}me}$ lettre de ce mot, et A^* l'ensemble de tous les mots. On aura besoin également de deux marqueurs pour les extrémités du mot. On peut considérer que tout mot sur A^* de longueur n est de la forme $w = \vdash w_1 \ldots w_n \dashv$, avec pour tout $i, w_i \in A$. Par extension, on notera si besoin $w_0 = \vdash$ et $w_{n+1} = \dashv$. On a donc $\varepsilon = \vdash \dashv$.

On note u.v ou uv la concaténation du mot u et du mot v, c'est-à-dire la suite de lettres $u_1, \ldots, u_m, v_1, \ldots, v_n$, avec |u| = m et |v| = n.

On pose également, pour tout $i \in \mathbb{N}$, $u^0 = \varepsilon$ et $u^{i+1} = u^i \cdot u$.

Monoïdes

Définition 1.2. Un *monoïde* $(\mathbb{M}, ., 1)$ est un ensemble \mathbb{M} muni d'une opération binaire interne associative . qui possède un unique élément neutre, noté 1 ou $1_{\mathbb{M}}$.

Le monoïde (M, ., 1), parfois noté M, est dit commutatif si l'opération . est commutative.

On peut définir pour tout élément x de \mathbb{M} et pour tout i de \mathbb{N} la puis-sance $i^{\grave{e}me}$ de $x:x^0=1$ et $\forall i\in\mathbb{N},\ x^{i+1}=x.x^i.$

Un monoïde est dit *libre* s'il admet un unique sous-ensemble X tel que tout élément x de \mathbb{M} admet une unique décomposition en produit d'éléments de X.

Exemple 1

 $(\mathbb{N}, \times, 1)$ forme un monoïde, de même que $(\mathbb{N}, +, 0)$ ou bien que l'ensemble A^* muni de la concaténation, avec comme élément neutre ε . Ces deux derniers sont de plus des monoïdes libres.

Définition 1.3. Dans un monoïde, deux éléments x et y sont dits

- \mathcal{L} -équivalents s'il existe deux éléments du monoïdes z et z' tels que x=zy et y=z'x, et
- \mathcal{R} -équivalents s'il existe deux éléments du monoïdes z et z' tels que x = yz et y = xz'.

On notera $x \sim_{\mathcal{L}} y$ et $x \sim_{\mathcal{R}} y$ ces propriétés.

Définition 1.4. Soit $(M, *, \mathbb{1}_m)$ et $(N, ., \mathbb{1}_n)$ deux monoïdes.

L'application φ de M dans N est un morphisme de monoïde si pour tout x, y de M, $\varphi(x * y) = \varphi(x)\varphi(y)$, et que $\varphi(\mathbb{1}_m) = \mathbb{1}_n$.

Bi-monoïdes et semi-anneaux

Définition 1.5. Un bi-monoïde fort $(\mathbb{K}, +, ., 0, 1)$ est un ensemble muni de deux opérations internes : « + » la somme, et « . » le produit, avec 0 l'élément neutre de la somme, et 1 celui du produit ; et vérifiant les propriétés suivantes :

- $(\mathbb{K}, +, 0)$ est un monoïde commutatif,
- $(\mathbb{K}, .., 1)$ est un monoïde,
- 0 est absorbant pour le produit : $\forall x \in \mathbb{K}, \ 0.x = x.0 = 0.$

Définition 1.6. Un semi-anneau $(\mathbb{K}, +, ., 0, 1)$ est un bi-monoïde fort dont le produit est distributif par rapport à la somme, *i.e.* :

$$\forall x, y, z \in \mathbb{K}, \ x.(y+z) = x.y + x.z,$$
$$\forall x, y, z \in \mathbb{K}, \ (x+y).z = x.z + y.z.$$

 $(\mathbb{K}, +, ., 0, 1)$ est parfois noté simplement $(\mathbb{K}, +, .)$ ou \mathbb{K} .

Exemple 2

Les ensembles \mathbb{Q} , \mathbb{R} et \mathbb{C} , munis du produit et de la somme usuelle sont des semi-anneaux, tout comme $(\mathbb{N} \cup \{\infty\}, min, +)$, qui modélise les distances, et $(\mathbb{N} \cup \{\infty\}, +, min)$, qui modélise les flots. $(\mathbb{N}, +, min)$ est un bi-monoïde qui n'est pas un semi-anneau.

Lorsque l'ensemble \mathbb{K} est fini, on parle de *semi-anneau fini* : les parties d'un ensemble fini munies de l'intersection et de l'union, par exemple.

On considèrera également les corps, qui sont des semi-anneaux particuliers.

Définition 1.7. Un corps ($\mathbb{K}, +, ., 0, 1$) est un anneau (semi-anneau dans lequel tout élément admet un inverse pour la somme) dans lequel tout élément non nul admet un inverse pour le produit.

De même que pour les semi-anneaux, $(\mathbb{K}, +, ., 0, 1)$ est parfois noté simplement $(\mathbb{K}, +, .)$ ou \mathbb{K} .

Définition 1.8. Une *norme* sur un corps \mathbb{K} est une application $\|.\|$ de \mathbb{K} dans \mathbb{R}_+ , telle que :

- $\forall x \in \mathbb{K}, ||x|| = 0 \Rightarrow x = 0_{\mathbb{K}},$
- $\forall x, y \in \mathbb{K}, ||xy|| = ||x|| ||y||$, et
- $\forall x, y \in \mathbb{K}, ||x + y|| \le ||x|| + ||y||.$

Somme infinie et étoile

On souhaite de plus munir notre structure \mathbb{K} , un bi-monoïde fort ou un semi-anneau, d'un opérateur partiel de somme infinie, qui sera noté \sum_{∞} ou $\sum_{i\in I}$, avec I un ensemble dénombrable, et d'un opérateur partiel d'itération, l'étoile de Kleene.

Définition 1.9. Une famille dénombrable est une suite non ordonnée d'éléments d'un ensemble X. On peut donc la voir comme une fonction de X dans $\mathbb{N} \cup \{\infty\}$. Une famille $(s_i)_{i \in I}$ sera dite sommable dans \mathbb{K} un bi-monoïde fort si $\sum_{i \in I} s_i$ existe dans \mathbb{K} .

Afin que la définition du comportement des automates soit cohérente et que ces opérateurs vérifient les identités habituelles, on souhaite que la somme infinie vérifie un certain nombre de propriétés. En fonction de la structure, bi-monoïde fort ou semi-anneau, on souhaitera disposer de tout ou partie de ces propriétés en tant qu'axiomes.

Il s'agit des axiomes suivants :

 $\mathbf{Ax.1}\,$ Si la famille I est finie, l'opérateur $\sum\limits_{i\in I}$ et la somme de $\mathbb K$ coı̈ncident ;

Ax.2 (Commutativité)

Soit π une permutation sur I. Si la famille $(s_i)_{i\in I}$ est sommable de somme l, alors la famille $(s_{\pi(i)})_{i\in I}$ est sommable de somme l.

Ax.3 L'union finie de familles sommables est sommable, et en posant k familles sommables $((x_{i,j})_{i \in I_j})_{j \in [\![1,k]\!]}$ de somme s_j , on a

$$\sum_{\substack{j \in \llbracket 1,k \rrbracket \\ i \in I_j}} x_{i,j} = \sum_{j=1}^k s_j.$$

Ax.4 (1^e sommation par blocs).

Soit $(s_i)_{i\in I}$ une famille sommable, et $(I_j)_{j\in J}$ une partition de I.

Si pour tout j de J, $\sum_{i \in I_j} s_i$ est définie (on la notera r_j) alors $\sum_{j \in J} r_j$ est définie et vaut $\sum_{i \in I} s_i$.

De plus, dans les semi-anneaux, on souhaite que l'opérateur de somme infinie vérifie des axiomes ayant trait à la distributivité, que l'on n'exigera pas pour une somme infinie sur un bi-monoïde.

Ces axiomes sont ordonnés par croissance des contraintes qu'ils expriment : chacun est conséquence de l'axiome qui le suit.

Ax.5 Si la famille $(s_i)_{i\in I}$ est sommable, alors pour tout x de \mathbb{K} , $(x.s_i)_{i\in I}$ et $(s_i.x)_{i\in I}$ sont sommables, et

$$\sum_{i \in I} x.s_i = x \left(\sum_{i \in I} s_i \right), \tag{1.1}$$

$$\sum_{i \in I} s_i . x = \left(\sum_{i \in I} s_i\right) x,\tag{1.2}$$

(1.3)

Ax.6 Soit $(x_i)_{i \in I}$ et $(y_j)_{j \in J}$ deux familles sommables de somme x et y. Alors la famille $(x_iy_j)_{i,j \in I \times J}$ est sommable, et sa somme vaut xy.

Ax.7 (2^e sommation par blocs)

Soit $(s_i)_{i \in I}$ une famille, et $(I_j)_{j \in J}$ une partition de I.

Si pour tout j de J, $\sum_{i \in I_j} s_i$ est définie (on la notera r_j) et que $\sum_{j \in J} r_j$ est définie, alors $\sum_{i \in I} s_i$ est définie et vaut $\sum_{j \in J} r_j$.

À partir de la somme infinie, on peut définir l'étoile de Kleene d'un élément x de \mathbb{K} , notée $x^*: x^*$ est définie si et seulement si $\sum_{i=0}^{\infty} x^i$ est définie, et dans ce cas, $x^* = \sum_{i=0}^{\infty} x^i$.

On a donc, si les axiomes $\mathbf{Ax.3}$ et $\mathbf{Ax.5}$ sont vérifiés, pour tout élément xde K dont l'étoile est définie :

$$x^* = 1 + xx^* = 1 + x^*x.$$

Proposition 1. Soit x et y deux éléments d'un semi-anneau muni d'un opérateur de somme infinie vérifiant les 5 premiers axiomes, et d'une étoile. Si $(yx)^*$ est définie, alors $(xy)^*$ est définie et on a :

$$(xy)^* = 1 + x(yx)^*y. (1.4)$$

Démonstration. Conséquence de l'axiome Ax.5. En effet, si $(yx)^*$ est définie, alors aussi $\sum_{n\in\mathbb{N}} x(yx)^n y$. Cette dernière somme est égale à $\sum_{n\in\mathbb{N}} (yx)^{n+1}$. Ainsi, $x(yx)^*y = \sum_{n\in\mathbb{N}} (yx)^{n+1}$, et en rajoutant 1 de chaque coté de l'égalité, on obtient le résultat souhaité.

Proposition 2. Soit x et y deux éléments d'un semi-anneau muni d'un opérateur de somme infinie vérifiant les 7 axiomes, et d'une étoile. Si l'étoile de x et celle de x*y sont définies, alors on a :

$$(x+y)^* = (x^*y)^*x^* (1.5)$$

Démonstration. Cette preuve est inspirée de [Ésik et Kuich, 2002]. On considère φ un morphisme de $\{a,b\}^*$ dans \mathbb{K} , le semi-anneau, tel que $\varphi(a)=x$ et $\varphi(b)=y.$ Comme x^* est définie, la somme $\sum\limits_{w\in a^*}\varphi(w)$ est bien définie. Considérons les langages $L_n = (a^*b)^n$ et $L = (a^*b)^*$. On montre par récurrence sur n que $\sum_{w \in L_n} \varphi(w)$ est bien définie. Il est clair que cela est le cas lorsque n = 0. Supposons la propriété vraie pour n. Par l'axiome $\mathbf{Ax.5}$, $\sum_{n \in \mathbb{N}} \varphi(a^n)\varphi(b)$ est définie et vaut $\sum_{w \in a^*b} \varphi(w)$. Donc par l'axiome **Ax.6**, $\sum_{w \in L_{n+1}} \varphi(w)$ est définie et

vaut
$$\left(\sum_{w \in L_n} \varphi(w)\right) \left(\sum_{w \in a^*b} \varphi(w)\right)$$
.

vaut $\left(\sum_{w\in L_n}\varphi(w)\right)\left(\sum_{w\in a^*b}\varphi(w)\right)$. Posons $s_n=\sum_{w\in L_n}\varphi(w)=(x^*y)^n$. Comme $(x^*y)^*$ est définie, $\sum_{n\in\mathbb{N}}s_n$ est définie. nie d'après l'axiome **Ax.5**, et vaut $(x^*y)^*$. Par l'axiome **Ax.7**, $\sum_{w\in L}^{n\in\mathbb{Z}^n} \varphi(w)$ est dé-

finie et vaut $(x^*y)^*$. Par l'axiome $\mathbf{Ax.6}$, $(x^*y)^*x^* = \left(\sum_{w \in L} \varphi(w)\right) \left(\sum_{w \in a^*} \varphi(w)\right)$. Ainsi, $\sum_{w \in (a^*b)^*a^*} \varphi(w)$ est définie, que l'on peut écrire $\sum_{w \in (a+b)^*} \varphi(w)$.

Définition 1.10. Les équations (1.4) et (1.5) sont dites *identités de Conway*. Un semi-anneau est dit *de Conway* si l'étoile de chaque élément est définie et que ces deux identités sont respectées.

Le semi-anneau des langages rationnels muni de l'union et de la concaténation est un semi-anneau de Conway.

Ces axiomes ne sont que des propriétés que l'on souhaite voir vérifiées par la somme infinie, mais ne disent rien de la façon dont elle est définie. Par exemple, dans les treillis, où la somme est définie comme un supremum, cet opérateur est défini comme une borne supérieure. Dans \mathbb{Q} , \mathbb{R} ou \mathbb{C} , afin d'obtenir l'axiome $\mathbf{Ax.2}$, une famille est sommable si elle est absolument sommable 1 , au sens classique. On peut remarquer que l'axiome $\mathbf{Ax.7}$ n'y est pas vérifié. On pourrait également donner une définition de l'étoile qui ne dépend pas de la somme infinie : avec x dans \mathbb{Q} , \mathbb{R} ou \mathbb{C} , tel que |x| < 1, on a $x^* = \frac{1}{1-x}$. Nous n'utiliserons pas cette définition : dans ce mémoire, l'étoile est utilisée pour représenter la somme de chemins sur un automate.

On peut ordonner les semi-anneaux en fonction des axiomes qu'ils vérifient avec leur étoile usuelle.

Pour cette classification, il existe trois grandes familles de semi-anneaux : les semi-anneaux dans lesquels l'étoile est définie partout, les semi-anneaux pour lesquels l'étoile n'est définie que partiellement, et les semi-anneaux dans lesquels l'étoile n'est définie que pour le 0, comme par exemple $(\mathbb{N} \cup \{\infty\}, \max, +)$.

 \mathbb{Q} , \mathbb{R} , \mathbb{C} , et les corps commutatifs en général sont des semi-anneaux dans lesquels la somme infinie n'est définie que partiellement.

On appelle rationnellement additifs les semi-anneaux pour lesquels les axiomes 1 à 7 sont vrais et les séries géométriques sont définies, et donc dans lesquels l'étoile est définie partout [Ésik et Kuich, 2002]. L'ensemble des langages rationnels, muni de l'union et de la concaténation, est un semi-anneau rationnellement additif. Les treillis distributifs, $(\mathbb{N} \cup \{\infty\}, min, +)$ ou $\mathbb{Q}_+ \cup \{\infty\}$, sont aussi de tels semi-anneaux.

Sur un corps muni d'une norme, on considèrera qu'une famille $(x_i)_{i\in I}$ est sommable si la famille $(\|x_i\|)_{i\in I}$ est sommable.

Proposition 3. Dans un corps muni d'une norme, la somme infinie respecte les axiomes $\mathbf{Ax.1}$ à $\mathbf{Ax.6}$. L'étoile d'un élément x est définie si et seulement $si \|x\| < 1$.

Démonstration. Il s'agit des propriétés usuelles des séries absolument convergentes sur \mathbb{R} .

Remarque 1. Sur \mathbb{Q} , qui ne vérifie pas l'axiome $\mathbf{Ax.7}$, la proposition 2 n'est pas vérifiée : en posant $x=y=\frac{1}{2}$, on a x^* définie, qui vaut $\frac{2}{3}$, $(x^*y)^*$ définie, qui vaut $\frac{3}{4}$, et leur produit vaut donc $\frac{1}{2}$, mais l'étoile de x+y=1 n'est pas définie.

^{1.} Pour \mathbb{Q} , on demande de plus que sa somme appartienne à \mathbb{Q} .

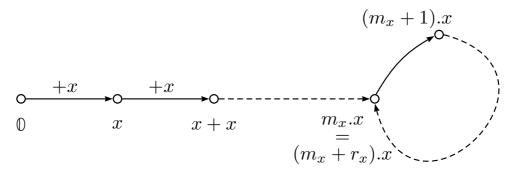


FIGURE 1.1 – Les éléments de $\{k.x \mid k \in \mathbb{N}\}$, ensemble fini.

Ordre d'un monoïde

On s'intéresse aux sommes infinies dans les bi-monoïdes localement finis. Un bi-monoïde muni d'un opérateur de somme infinie est dit *localement fini* si la clôture par le produit, la somme et la somme infinie de tout ensemble fini est finie. Cette définition est plus forte que la définition usuelle, pour laquelle la clôture par la somme infinie n'est pas exigée.

Remarque 2. Dans un souci de simplicité, on se placera dans le cas où \mathbb{K} est fini, l'extension des résultats à des bi-monoïdes forts localement finis étant aisée : ces résultats portant sur les automates, il suffit de considérer le bi-monoïde fort fini engendré par les poids utilisés dans l'automate.

Soit $(\mathbb{M}, +, \mathbb{O})$ un monoïde fini commutatif muni d'un opérateur de somme infinie, vérifiant les axiomes $\mathbf{Ax.1}$ à $\mathbf{Ax.3}$.

On notera par souci de simplification k.x la somme finie de k fois le même élément x de \mathbb{M} , avec k un entier naturel. On notera de même $\infty.x$ la somme infinie de x si celle-ci est définie.

Définition 1.11. Soit x un élement de \mathbb{M} . Son *ordre* est le couple minimal (m_x, r_x) d'entiers (avec r_x entier positif) qui vérifient :

$$\forall k \geqslant m_x, \ (k+r_x).x = k.x. \tag{1.6}$$

Cet ordre est toujours défini dans un monoïde localement fini. En effet, l'ensemble $\{k.x \mid k \in \mathbb{N}\}$ est fini, et la suite $s = (k.x)_{k \in \mathbb{N}}$ est ultimement periodique : si $s_i = s_j$, alors pour tout k entier naturel on a $s_{i+k} = i.x + k.x = j.x + k.x = s_{j+k}$. On peut ainsi voir r_x comme la distance minimale entre deux occurences égales dans s, et m_x comme le plus petit i tel que s_i apparait infiniment souvent dans s, ainsi que cela est représenté dans la Figure 1.1. On a également que si k est plus grand que m_x et que q est un multiple de r_x , alors (k+q).x = k.x

Définition 1.12. L'ordre d'un monoïde M est le couple

$$(\max\{m_x \mid x \in \mathbb{M}\}, \operatorname{ppmc}\{r_x \mid x \in \mathbb{M}\}), \tag{1.7}$$

avec ppmc(E) le plus petit multiple commun des éléments de E.

Définition 1.13. Pour tout k dans $\mathbb{N} \cup \{\infty\}$, on définit $k \mod(m, r)$ comme :

$$k \operatorname{mod}(m, r) = \begin{cases} k & \text{si } k < m \text{ ou si } k = \infty \\ m + ((k - m) \operatorname{mod} r) & \text{sinon.} \end{cases}$$
 (1.8)

En posant (m,r) l'ordre additif de \mathbb{M} , pour tout élément x de \mathbb{M} , on a $\forall k \geq m$, (k+r).x = k.x, et $\forall k$, $k.x = (k \mod(m,r)).x$. Ainsi, $k \mod(m,r)$ est compris entre 0 et m+r-1 ou bien est infini.

Dans la suite, on notera (m,r) l'ordre de \mathbb{M} , et $\mathcal{N}_{m,r}$ le semi-anneau $[0; m+r-1] \cup \{\infty\}$ où les opérations entre entiers finis sont modulo (m,r).

Définition 1.14. On appelle description d'une famille infinie I d'éléments de \mathbb{M} la fonction de \mathbb{M} dans $\mathcal{N}_{m,r}$ qui à chaque élément attribue son nombre d'occurences dans I modulo (m,r) si ce nombre est fini, et ∞ sinon

Par l'axiome $\mathbf{Ax.3}$, la somme des éléments de I et la somme des éléments de sa description avec leur multiplicité sont égales si celle-là est définie :

$$\sum_{i \in I} i = \sum_{x \in \mathbb{M}} d_I(x).x,\tag{1.9}$$

avec d_I la description de I.

Séries

L'ensemble des séries formelles sur A^* à coefficients dans \mathbb{K} est noté $\mathbb{K}\langle\langle A^*\rangle\rangle$. Une série s de $\mathbb{K}\langle\langle A^*\rangle\rangle$ est une application de A dans \mathbb{K} , et l'image d'un mot w par cette série, appelé coefficient de w, est noté $\langle s, w \rangle$. On note la série s comme la somme formelle

$$s = \sum_{w \in A^*} \langle s, w \rangle \ w.$$

On appelle terme constant de la série le coefficient du mot vide. S'il est nul, on dit que la série est propre. Si pour tout mot w, $\langle s, w \rangle \neq 0$, on dit que la série s a support plein.

On peut définir plusieurs opérations sur les éléments de $\mathbb{K}\langle\langle A^*\rangle\rangle$:

• La somme. La somme de deux séries est la série obtenue en effectuant la somme terme à terme des coefficients des deux séries.

$$s + t = \sum_{w \in A^*} (\langle s, w \rangle + \langle t, w \rangle) w.$$
 (1.10)

La série 0 (dans laquelle chaque coefficient est nul) est neutre pour cette opération.

• Le produit de Cauchy. Le *produit de Cauchy* est l'extension aux séries du produit de polynômes :

$$s \cdot t = \sum_{w \in A^*} \sum_{\substack{u, v \in A^* \\ uv = w}} (\langle s, u \rangle. \langle t, v \rangle) w. \tag{1.11}$$

La série 1 (dont tous les coefficients sont nuls sauf le terme constant qui vaut 1) est neutre pour le produit de Cauchy.

• Le produit de Hadamard. Le produit de Hadamard de deux séries est le produit terme à terme :

$$s \odot t = \sum_{w \in A^*} (\langle s, w \rangle. \langle t, w \rangle) w. \tag{1.12}$$

L'élément neutre pour cette opération est la série dont tous les coefficients sont égaux à 1, c'est la série caractéristique de A^* , et on la notera également A^* .

Les deux produits sont associatifs et distributifs par rapport à la somme. Ils permettent chacun de définir une puissance pour une série s:

$$s^{i} = \begin{cases} 1 & \text{si } i = 0, \\ s^{i-1} \cdot s & \text{sinon.} \end{cases} \qquad s^{\odot i} = \begin{cases} A^{*} & \text{si } i = 0, \\ s^{\odot i-1} \cdot s & \text{sinon.} \end{cases}$$
 (1.13)

Ces puissances permettent de définir deux itérateurs :

• L'étoile de Kleene. L'étoile de Kleene d'une série est la somme des puissances itérées de la série :

$$s^* = \sum_{w \in A^*} \sum_{i=0}^{\infty} \langle s^i, w \rangle w = \sum_{w \in A^*} \sum_{i=0}^{|w|} \langle s^i, w \rangle w.$$
 (1.14)

Cette somme est définie pour toutes les séries propres. Pour les séries non propres, cette somme est définie si et seulement si l'étoile de Kleene du terme constant de la série est définie. Dans ce cas, on peut montrer (cf. [Sakarovitch, 2009]) que $s^* = (s_0^* s_p)^* s_0^*$, avec $s = s_0 + s_p$ une série non propre, s_p une série propre, et $s_0 = \langle s, \varepsilon \rangle$.

• L'étoile de Hadamard. L'étoile de Hadamard d'une série est la somme des puissances de Hadamard itérées de la série

$$s^{\circledast} = \sum_{w \in A^*} \sum_{i=0}^{\infty} \langle s^{\odot i}, w \rangle w. \tag{1.15}$$

On constate que pour tout w, $\langle s^{\circledast}, w \rangle = \langle s, w \rangle^*$.

Remarque 3. Les étoiles ont été définies ici comme des opérateurs unaires. Il est possible – et on le fera plus tard – de les considérer comme des opérateurs binaires. C'est d'ailleurs la définition initiale de l'étoile de Kleene dans [Kleene, 1956]. On définit ces opérateurs binaires comme : $s * t = s^* \cdot t$ et $s \circledast t = s^* \odot t$.

Remarque 4. Dans le cas où $x \in \mathbb{C}$, \mathbb{R} ou \mathbb{Q} , l'étoile de x n'est définie que si |x| < 1, et on a $x^* = \frac{1}{1-x}$.

Remarque 5. Contrairement au cas booléen, l'étoile de Kleene n'est pas en général idempotente. En effet, $\left(-\frac{1}{2}\right)^* = \frac{2}{3}$ et $\left(-\frac{1}{2}\right)^{*^*} = 3$. En fait, on a déjà que $s^*.s^*$ est en général différent de s^* . Par exemple, si s=a, alors $s^*=1+a+aa+aaa+\ldots$ et $s^*.s^*=1+2a+3aaa+\ldots$

Un polynôme est une série de $\mathbb{K}\langle\langle A^*\rangle\rangle$ dont tous les coefficients sont nuls à partir d'une taille de mot suffisamment grande.

Les séries rationnelles sont la clôture des polynômes par les opérations rationnelles : la somme, le produit de Cauchy et l'étoile de Kleene. On notera $\mathbb{K}\operatorname{Rat} A^*$ l'ensemble des séries rationnelles sur A^* .

Les séries de Hadamard sont la clôture des séries rationnelles par les opérations terme à terme : la somme, le produit de Hadamard et l'étoile de Hadamard. On notera \mathbb{K} Had A^* l'ensemble des séries de Hadamard sur A^* .

Expressions

Les expressions sont des objets finis permettant de décrire inductivement des séries. On considère le plus souvent les expressions rationnelles, qui sont en général attribuées à Kleene. Leur version pondérée a été utilisée tout d'abord par [Caron et Flouret, 2003].

Définition 1.15. La grammaire suivante génère les expressions rationnelles pondérées :

$$R \to 0 \mid 1 \mid a \in A \mid kR, k \in \mathbb{K} \mid Rk, k \in \mathbb{K} \mid R + R \mid RR \mid R^*.$$
 (1.16)

On notera $\mathbb{K}\mathsf{RatExp}A$ l'ensemble des expressions rationnelles sur A à coefficients dans \mathbb{K} .

La grammaire suivante génère les expressions de Hadamard:

$$H \to R \mid kH, k \in \mathbb{K} \mid Hk, k \in \mathbb{K} \mid H+H \mid H \odot H \mid H^{\circledast}. \tag{1.17}$$

On notera $\mathbb{K}HadExpA$ l'ensemble des expressions rationnelles.

Définition 1.16. L'interprétation d'une expression de Hadamard est une série de Hadamard, définie inductivement comme suit :

$$[0] = 0_{\mathbb{K}},$$
 $[1] = 1_{\mathbb{K}},$ (1.18)

$$\forall a \in A, [a] = a,$$
 $[F + G] = [F] + [G],$ (1.19)

$$\forall k \in \mathbb{K}, [k \in \mathbb{K}] = k [\in \mathbb{K}], [k \in \mathbb{K}] = [\in \mathbb{K}],$$
 (1.20)

$$[\![\mathsf{FG}]\!] = [\![\mathsf{F}]\!] \cdot [\![\mathsf{G}]\!], \qquad [\![\mathsf{F}^*]\!] = [\![\mathsf{F}]\!]^*, \qquad (1.21)$$

$$\llbracket \mathsf{F} \odot \mathsf{G} \rrbracket = \llbracket \mathsf{F} \rrbracket \odot \llbracket \mathsf{G} \rrbracket, \qquad \qquad \llbracket \mathsf{F}^{\circledast} \rrbracket = \llbracket \mathsf{F} \rrbracket^{\circledast}. \tag{1.22}$$

Une expression de Hadamard E est dite *valide* si son interprétation est définie, *i.e.* si la série $\llbracket \mathsf{E} \rrbracket$ existe. Dans ce cas, pour tout w dans A^* , on note $\langle \mathsf{E}, w \rangle = \langle \llbracket \mathsf{E} \rrbracket, w \rangle$.

Les seules opérations pouvant ne pas être définies sont les itérations. Pour vérifier la validité d'une expression E, il faut étudier chacune des sous-expressions se trouvant sous une étoile dans E.

Dans \mathbb{Q} , \mathbb{R} ou \mathbb{C} muni de la somme et du produit usuels, E^* est valide si et seulement si $|\langle \mathsf{E}, \varepsilon \rangle| < 1$, et E^\circledast est valide si et seulement si pour tout $w \in A^*$, $|\langle \mathsf{E}, w \rangle| < 1$,

Nous verrons dans la suite que la validité d'une expression de Hadamard sur $\mathbb Q$ est indécidable.

Chapitre 2

Automates

"Chouette! Où est-ce qu'on va?"

— Pippin (Billy Boyd), La communauté de l'anneau 2001

Les objets mathématiques concernés par l'essentiel des résultats de ce mémoire sont les automates finis. Dans ce chapitre, on présentera les différents modèles d'automates que nous allons étudier, ainsi que quelques fonctions pour les manipuler.

Automates unidirectionnels

Les automates les plus simples sont les automates unidirectionnels. Ces machines, dont l'origine est en général ² attribuée à Kleene [Kleene, 1956], décrivent un langage sur un alphabet donné. Un automate prend en entrée un mot et renvoie vrai ou faux selon l'appartenance de ce mot au langage que cet automate décrit.

Pour ce faire, au fur et à mesure de la lecture du mot d'entrée, il passe par différents états. Pour changer d'état, l'automate emprunte une transition étiquetée par la lettre courante, et la consomme, jusqu'à arriver à la fin du mot. Si le dernier état est un état acceptant, ou terminal, alors le mot est accepté. Le mot est rejeté s'il n'existe aucun moyen d'arriver à un état final en fin du mot. Ainsi, l'automate de la Figure 2.1 décrit l'ensemble des mots qui comportent un nombre impair de a sur l'alphabet $\{a,b\}$.

La vision classique de ces objets est une vision *combinatoire*, c'est-à-dire que l'on considère le graphe étiqueté défini à partir de l'automate, et un calcul peut-être défini comme un chemin sur le graphe, qui va d'un état initial à un état final. L'étiquette du calcul est le mot lu le long des transitions empruntées

^{2.} Le lecteur intéressé par les origines historiques détaillées de ce modèle peut consulter [Perrin, 1995]

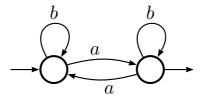


FIGURE 2.1 – Un automate décrivant les mots avec un nombre impair de a.

lors de ce calcul. De prime abord, on considère d'abord séparément les différents calculs de même étiquette.

On peut étendre ce modèle : au lieu de n'avoir que deux comportements pour un mot, l'accepter ou le refuser, on permet à l'automate de lui attribuer un poids. Pour ce faire, chaque transition est munie d'un poids, le poids d'un calcul sera le produit des transitions, et le poids d'un mot la somme des poids des calculs. On souhaite donc que les poids soient dotés d'une structure ayant deux opérations, la somme et le produit, ayant de bonnes propriétés. La structure minimale est celle du bi-monoïde, mais parfois, nous aurons besoin d'une structure plus puissante, celle du semi-anneau, qui rajoute la distributivité. C'est la structure usuelle de poids sur un automate, puisqu'elle permet entre autres des manipulations des représentations des automates en tant que matrices.

Dans la suite, et sauf mention contraire, on considère K un bi-monoïde fort.

Définition 2.1. Un \mathbb{K} -automate, ou encore automate pondéré sur \mathbb{K} , en anglais weighted automaton, sur l'alphabet A est un quadruplet (Q, E, I, T), avec :

- Q un ensemble fini d'états,
- E une fonction de $Q \times A \times Q$ vers K, la fonction de transition,
- I une fonction de Q vers \mathbb{K} , les poids initiaux,
- T une fonction de Q vers \mathbb{K} , les poids finaux.

En raisons des modèles étudiés dans la suite, on parlera de ces objets en tant qu'automates unidirectionnels.

Remarque 6. Il est possible d'étendre la définition à des automates infinis, c'est-à-dire ayant un nombre infini d'états. Les séries qui nous intéressent ici sont celles réalisées par des automates finis, d'où cette restriction. On se servira ponctuellement dans les preuves d'automates infinis.

Dans le cas pondéré, ce qui nous intéresse n'est pas tant la définition formelle de l'objet ou le graphe sous-jacent, mais la sémantique qui lui est associée : la façon de calculer le poids d'un mot dans cet automate. C'est l'objet des définitions suivantes.

Quelques définitions

Soit $\mathcal{A} = (Q, E, I, T)$ un \mathbb{K} -automate. On rappelle que le support d'une fonction est l'ensemble des éléments de son domaine de définition dont l'image est non nulle. L'ensemble des transitions est le support de E, les états initiaux le support de E, et les états finaux le support de E.

Une transition (p, a, q) se décompose en son état source p, noté $\sigma(p, a, q)$, son étiquette a, notée $\lambda(p, a, q)$, et son état cible q, noté $\tau(p, a, q)$.

On appellera transitions entrantes de l'état p les transitions dont l'état cible est p, et transitions sortantes celles dont l'état source est p,

Un chemin ρ de \mathcal{A} de longueur n est un triplet (p, e, q), avec p et q deux états de l'automate, et e une suite de transitions de la forme $(p_i, a_i, q_i)_{i \in [1, n]}$, avec $p_i, q_i \in Q$ et $a_i \in A$ tels que $p = p_1, q = q_n$ et que pour tout i dans [1, n-1], $q_i = p_{i+1}$ (si n = 0, alors p = q). Le poids de ce chemin est $\prod_{i=1}^n E(p_i, a_i, q_i)$. L'étiquette de ce chemin est le mot $w = a_1 \dots a_n$.

Un calcul de \mathcal{A} de poids k est un chemin (p, e, q) de poids k' tel que p est initial, q est final, et k = I(p)k'I(q).

Un mot w a pour poids k dans \mathcal{A} si la somme des poids des calculs dont l'étiquette est w vaut k. On note $\langle \mathcal{A}, w \rangle = k$.

Ainsi, un automate $r\'{e}alise$ une série : la série qui à chaque mot associe son poids dans l'automate. C'est le comportement de \mathcal{A} , noté

$$|\mathcal{A}| = \sum_{w \in A^*} \langle \mathcal{A}, w \rangle \ w.$$

On note $\mathbb{K}\mathsf{Aut}A$ l'ensemble des séries réalisables par un automate sur l'alphabet A.

Définition 2.2. Deux automates sont dits *équivalents* s'ils réalisent la même série.

Remarque 7. Dans le cas où \mathbb{K} est le semi-anneau booléen, on dit qu'un mot est accepté par \mathcal{A} s'il existe un calcul dans l'automate dont il est l'étiquette, i.e. $\langle \mathcal{A}, w \rangle = 1$. Le langage reconnu par un automate est l'ensemble des mots acceptés par celui-ci : c'est le support de la série qu'il réalise.

En général, le langage support de la série réalisée par l'automate est inclus dans le langage reconnu par l'automate support ¹. Si le semi-anneau est positif ², alors ces deux langages sont égaux.

Dans le cas des semi-anneaux en général, on a le théorème suivant, dit de Kleene-Schützenberger [Schützenberger, 1961], extension du résultat de Kleene sur les langages :

^{1.} Avec un automate $\mathcal{A}=(Q,E,I,T)$, c'est l'automate booléen (Q,E',I',T') tel que E'(p,a,q)=1 si et seulement si $E(p,a,q)\neq 0$, et de même pour les états initiaux et finaux.

^{2.} Dans le sens le plus faible : $x + y = 0 \Rightarrow x = y = 0$, et $xy = 0 \Rightarrow x = 0$ ou y = 0.

Théorème 2.1. Soit \mathbb{K} un semi-anneau. L'ensemble des séries réalisables par les \mathbb{K} -automates sont les séries \mathbb{K} -rationnelles (i.e. $\mathbb{K} \operatorname{Aut} A = \mathbb{K} \operatorname{Rat} A^*$).

Cette équivalence est constructive : à partir d'une expression rationnelle pondérée, on peut construire un automate réalisant la même série, et viceversa.

De nombreuses méthodes ont été décrites pour réaliser cette construction, soit dans le cas booléen, soit dans le cas pondéré. On en rappellera un certain nombre dans la section 3.3.

Extensions du modèle

On peut voir les automates comme des restrictions des machines de Turing. Une façon classique pour étendre le modèle est de lever certaines de ces restrictions. Nous allons ici présenter trois extensions classiques, qui lèvent petit à petit les limitations sur les mouvements de la tête de lecture.

Automates circulaires

Les automates circulaires (en anglais *rotating automata*) sont des automates qui peuvent relire le mot en entrée autant de fois qu'ils le souhaitent, et qui sont comme les K-automates classiques, en lecture seule.

Ainsi, on peut considérer qu'il s'agit de K-automates classiques munis de transitions spéciales qui ramènent la tête de lecture au début du mot.

Définition 2.3. Un automate circulaire sur l'alphabet A est un quintuplet (Q, E, R, I, T), où :

- Q, E, I, T sont définis de la même façon que dans le cas classique des \mathbb{K} -automates,
- R est une fonction de $Q \times Q$ vers K, la fonction de rembobinage.

On représentera R(p,q)=k comme une transition de poids k entre p et q étiquetée par la lettre \mathbf{r} , n'appartenant pas à l'alphabet A. On pourra appeler ces transitions des \mathbf{r} -transitions.

Ainsi, l'automate de la Figure 2.2 lit un mot non vide sur l'alphabet $\{a\}$, puis retourne au début du mot. Il attribue ensuite à toute lettre le poids $\frac{1}{2}$, sauf une, choisie de manière non déterministe, à laquelle il donne poids $\frac{1}{3}$. Ensuite, grâce à la **r**-transition, il peut retourner au début du mot pour recommencer cette dernière partie.

Remarque 8. Tout \mathbb{K} -automate peut-être vu comme un \mathbb{K} -automate circulaire, en posant R la fonction nulle : $\forall \mathbb{K}$, $\mathbb{K}\mathsf{Aut} A \subseteq \mathsf{R}\mathbb{K}\mathsf{Aut} A$. Un tel automate n'aura aucune transition étiquetée par \mathbf{r} .

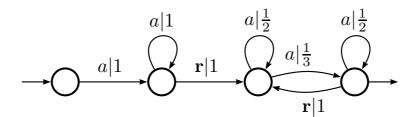


FIGURE 2.2 – Un automate circulaire pondéré.

Remarque 9. Il existe des semi-anneaux dans lesquels les automates circulaires sont strictement plus puissants que les automates unidirectionnels. Par exemple, dans le semi-anneau $\operatorname{Rat} A^*$ (ce qui est une façon de présenter les transducteurs), les automates circulaires peuvent réaliser la série qui à tout mot associe le carré de ce mot, or cette série n'est pas rationnelle. Dans la suite seront abordés d'autres exemples, notamment sur $\mathbb Q$, donc dans un semi-anneau commutatif.

Comportement et validité

Comme dans les automates unidirectionnels, le poids d'un calcul est le produit des poids des transitions rencontrées, en prenant en compte le poids initial et le poids final.

Dans un automate circulaire, on ne considère que des calculs d'étiquette $(w\mathbf{r})^i w$, avec i un entier.

Définition 2.4. Le comportement d'un automate circulaire \mathcal{A} est la série qui à tout mot w de A^* associe la somme des poids des calculs de \mathcal{A} d'étiquette $(w\mathbf{r})^i w$, avec i un entier. On note RKAutA l'ensemble des séries réalisables par les automates circulaires sur l'alphabet A.

Définition 2.5. Un automate circulaire \mathcal{A} est dit valide si pour tout mot w, la somme des poids des calculs de \mathcal{A} sur w est définie.

Automates à navette

Les automates à navette (en anglais *sweeping automata*) sont des automates qui peuvent lire le mot en entrée autant de fois qu'ils le souhaitent, en alternant le sens de lecture, d'abord de gauche à droite, puis de droite à gauche, *etc*.

Ainsi, on peut considerer qu'il s'agit de deux K-automates classiques, l'un lisant de droite à gauche et l'autre de gauche à droite, et l'on ne peut passer de l'un à l'autre qu'en extrémité de mot.

Définition 2.6. Un automate à navette sur l'alphabet A est un quintuplet (F, B, E, I, T), avec :

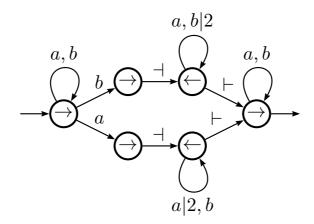


FIGURE 2.3 – Un automate à navette pondéré.

- $Q = F \cup B$, l'ensemble fini des états, avec F l'ensemble des états forward et B l'ensemble des états backward,
- E une fonction de

$$F \times A \times F \qquad \qquad \text{(la partie qui lit de gauche à droite)} \\ \cup F \times \{\exists\} \times B \qquad \qquad \text{(les demi-tours à la fin du mot)} \\ \cup B \times A \times B \qquad \qquad \text{(la partie qui lit de droite à gauche)} \\ \cup B \times \{\vdash\} \times F \qquad \qquad \text{(les demi-tours au début du mot)}$$

dans \mathbb{K} .

• I, T deux fonctions de F dans \mathbb{K} , I la fonction des poids initiaux et T celle des poids finaux.

Exemple 3

Pour l'automate de la Figure 2.3, on a F l'ensemble des états notés avec \rightarrow et B ceux notés par \leftarrow . De plus, par simplicité de lecture, on a omis le poids des transitions lorsque celui-ci était 1. Cet automate lit un mot non vide sur l'alphabet $\{a,b\}$, puis lit le mot à l'envers en associant le poids 2 à toute lettre identique à la dernière lettre. Il relit ensuite le mot.

Remarque 10. Tout automate circulaire peut-être vu comme un automate à navette. En effet, il suffit de simuler les transitions R(p,p') par l'ajout d'un état q et d'étendre $E: E(p,\vdash,q) = R(p,p'), E(q,\dashv,p') = 1$ et pour tout a de A, E(q,a,q) = 1. On a donc $\forall \mathbb{K}, \mathsf{RKAut} A \subsetneq \mathsf{SKAut} A$.

Remarque 11. Il existe des semi-anneaux dans lesquels les automates à navette sont strictement plus puissants que les automates circulaires. Par exemple, les transducteurs à navette peuvent réaliser la série qui à tout mot associe le miroir de ce mot. Cette série n'est pas réalisable par un automate circulaire (voir [Guillon, 2016] par exemple).

Comportement et validité

Comme dans les automates unidirectionnels, le poids d'un calcul est le produit des poids des transitions rencontrées, en prenant en compte le poids initial et le poids final.

Notons \overline{u} le miroir du mot u, c'est-à-dire le mot u lu à l'envers. Dans un automate à navette, on ne considère que les calculs d'étiquette $w(\exists \overline{w} \vdash w)^i$ avec i un entier.

Définition 2.7. Le comportement d'un automate à navette \mathcal{A} est la série qui à tout mot w de A^* associe la somme des poids des calculs de \mathcal{A} d'étiquette $w(\neg \overline{w} \vdash w)^i$, avec i un entier. On note SKAutA l'ensemble des séries réalisables par les automates à navette sur l'alphabet A.

Définition 2.8. Un automate à navette \mathcal{A} est dit *valide* si pour tout mot w, la somme des poids des calculs de \mathcal{A} sur w est définie.

Proposition 4. Si \mathbb{K} est commutatif, alors tout automate à navette est équivalent à un automate circulaire.

Démonstration. L'idée de la preuve, illustrée dans la Figure 2.4, est de prendre le "miroir" de la partie backward de l'automate à navette, que l'on peut simuler par un automate unidirectionnel, et de reconnecter les deux parties grâce à des **r**-transitions.

Soit $\mathcal{A} = (F, B, E, I, T)$ un automate à navette. Soit $\mathcal{B} = (Q, E', R, I, T)$ un automate circulaire, avec :

$$\bullet \ Q = F \cup B.$$

•
$$E'(p, a, q) = \begin{cases} E(p, a, q) & \text{si } p, q \in F, \\ E(q, a, p) & \text{si } p, q \in B, \\ 0 & \text{sinon}, \end{cases}$$

•
$$R(p,q) = \begin{cases} E(p,\vdash,q) & \text{si } p \in F, \\ E(p,\dashv,q) & \text{si } p \in B, \\ 0 & \text{sinon.} \end{cases}$$

Il y a bijection entre les chemins de \mathcal{A} et ceux de \mathcal{B} , et cette bijection conserve les poids : en effet, comme \mathbb{K} est commutatif, le poids d'un mot ne change pas en fonction du sens de lecture de celui-ci.

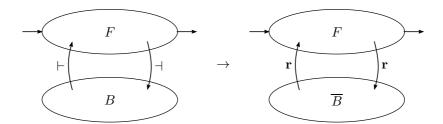
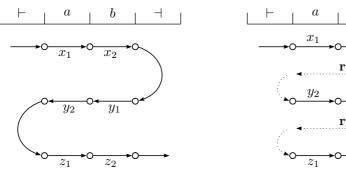


FIGURE 2.4 – La transformation d'un automate à navette en automate circulaire.



(a) Un calcul d'un automate à navette sur le mot ab.

(b) Un calcul d'un automate circulaire sur le mot ab.

FIGURE 2.5 – Deux calculs de poids $x_1x_2y_1y_2z_1z_2$.

À un chemin d'étiquette $w(\vdash \overline{w} \dashv w)^i$ dans \mathcal{A} , cette bijection associe un chemin d'étiquette $w(\mathbf{r}w)^{2i}$ dans \mathcal{B} . Ces deux chemins contribuent au poids du même mot w. La Figure 2.5 illustre la bijection.

Cette proposition va nous permettre de nous concentrer dans la suite sur les automates circulaires, et non les automates à navette, car les semi-anneaux manipulés seront souvent commutatifs. Le modèle est cependant présenté ici dans un souci d'exhaustivité.

Automates boustrophédons

On peut encore lever la restriction sur les mouvements de la tête de lecture afin de retrouver le mode de déplacement d'une machine de Turing : à tout moment durant la lecture du mot, la tête de lecture peut se déplacer sur la droite aussi bien que sur la gauche. Ces automates ont été introduits et prouvés équivalents aux automates unidirectionnels dans le cas booléen par [Rabin et Scott, 1959] et [Shepherdson, 1959]. Ce n'est généralement pas le cas lorsque ces automates sont pondérés. La définition donnée ici suit celle de [Birget, 1989.

Définition 2.9. Un \mathbb{K} -automate boustrophédon (en anglais two-way automaton) sur l'alphabet A est un quintuplet (F, B, E, I, T), avec :

- $Q = F \cup B$, l'ensemble fini des états, avec F l'ensemble des états forward et B l'ensemble des états backward,
- E une fonction de

$$Q \times A \times Q$$
 (les déplacements dans le mot)
 $\cup F \times \{\exists\} \times B$ (les demi-tours à la fin du mot)
 $\cup B \times \{\vdash\} \times F$ (les demi-tours au début du mot)

dans K, la fonction de transition,

- I une fonction de F vers \mathbb{K} , les poids initiaux,
- T une fonction de F vers \mathbb{K} , les poids finaux.

 $Nota\ bene$: Seuls les états de F peuvent être initiaux ou finaux.

On définit de plus une fonction sur les états : $\delta(p) = 1$ si p est dans F, et $\delta(p) = -1$ si p est dans B.

Définition 2.10. Soit $\mathcal{A} = (Q, E, I, T)$ un \mathbb{K} -automate boustrophédon.

Un chemin ρ de \mathcal{A} de longueur n est un triplet p, e, q, avec p et q deux états de l'automate, et e une suite de transitions de la forme $(p_i, a_i, q_i)_{i \in [1,n]}$, avec $p_i, q_i \in Q$ et $a_i \in A \cup \{\vdash, \dashv\}$ avec $p = p_1, q = q_n$ et tels que pour tout i dans [1, n-1], $q_i = p_{i+1}$.

L'étiquette d'un chemin $(p, (e_i)_{i \in [\![1,k]\!]}, q)$ existe et est le mot $w = w_1 \dots w_n$ $(w_i \in A \cup \{\vdash, \dashv\})$ s'il existe une application surjective π de $[\![1,k]\!]$ vers $[\![1,n]\!]$ telle que :

- $Im(\pi) = [1, n]$
- pour tout i dans [2, k], $\pi(i) = \pi(i-1) + \delta(\sigma(e_i))$
- pour tout i dans [1, k], $\lambda(s_i) = w_{\pi(i)}$.

Un calcul de \mathcal{A} d'étiquette w est un chemin $(p, (e_i)_{i \in \llbracket 1, k \rrbracket}, q)$ d'étiquette dans $(\vdash +\varepsilon)w(\varepsilon \vdash \dashv)$ tel que :

- p est initial,
- q est final,
- $\pi(1) = 1$,

- (a) Un automate boustrophédon.
- (b) Un calcul sur ba.

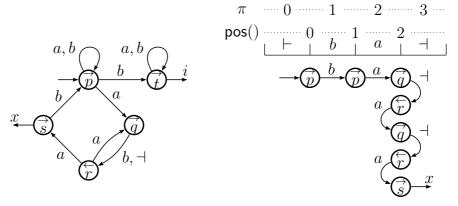


FIGURE 2.6 – Un automate boustrophédon et un calcul.

•
$$\pi(k) = n$$
.

On définit de plus une fonction pos de $[\![1,k]\!]$ dans $[\![0,n]\!]$ pour les états d'un chemin :

pour tout
$$j \in [1, k]$$
, $pos(j) = \pi(j)$ si $\delta(p_j) = 1$, et $pos(j) = \pi(j) - 1$ sinon.

Remarque 12. Dans la suite, on identifiera un chemin et la suite des états rencontrés par celui-ci.

Remarque 13. Tout automate à navette peut être vu comme un automate boustrophédon : $\forall \mathbb{K}$, $\mathsf{SKAut} A \subseteq 2\mathbb{K}\mathsf{Aut} A$.

Remarque 14. Il existe des semi-anneaux dans lesquels les automates boustrophédons sont strictement plus puissants que les automates à navette.

Exemple 4 (tiré de [Guillon, 2016])

On considère $\mathbb{K} = (\mathcal{P}(\mathbb{N}), \cup, +)$. La série s, telle que pour tout mot w, $\langle s, w \rangle = \{kn \mid k \in \mathbb{N}, ba^nb$ est un facteur de $w\}$, est réalisable par un automate boustrophédon, mais pas par un automate à navette. Il s'agit de la série qui à chaque mot w composé de k blocs de a, le i-ème bloc étant de longueur n_i , associe $\sum_{i=1}^k n_i \mathbb{N}$.

Comportement et validité

Définition 2.11. Le comportement d'un automate boustrophédon \mathcal{A} , noté |A|, est la série qui à tout mot w de A^* associe la somme des poids des calculs de \mathcal{A} d'étiquette w. On note $2\mathbb{K}\mathsf{Aut}A$ l'ensemble des séries réalisables par les automates boustrophédons sur l'alphabet A.

Définition 2.12. Un automate boustrophédon pondéré sur \mathbb{K} est dit *valide* si pour tout mot w, la somme des poids des calculs de \mathcal{A} d'étiquette w existe dans \mathbb{K} , donc si la série $|\mathcal{A}|$ est bien définie.

Dans certains cas, on peut avoir besoin d'une notion plus fine de validité : c'est le but de la définition suivante.

Définition 2.13. Un automate boustrophédon \mathcal{A} est dit fortement valide si pour tout mot w, pour tout états p, q de \mathcal{A} , pour toutes positions i, j dans w, la somme des poids des chemins sur w commençant en p à la position i et finissant en q dans la position j est définie.

Remarque 15. Tout automate unidirectionnel est fortement valide.

Variantes du modèle

Il existe d'autres définitions d'un automate boustrophédon. Les variantes principalement utilisées diffèrent de notre modèle soit dans la définition des calculs, soit dans la façon de représenter les mouvements de la tête de lecture (soit les deux). Le modèle utilisé en général sera le modèle présenté ci-dessus, mais il arrivera que l'on change de modèle localement afin d'avoir des calculs plus agréables à manipuler.

Les calculs peuvent être définis comme se finissant non à droite du mot comme dans notre définition, mais se finissant à gauche du mot, voire n'importe où. Plus formellement, dans les définitions 2.9 et 2.10, on remplace $\pi(k) = n$ par $\pi(k) = 1$ et T une fonction de B dans \mathbb{K} , ou bien par $\pi(k) \in [1, n]$ et T une fonction de Q dans \mathbb{K} . Ainsi, en fonction de la position de début des calculs, et de celle de la fin, un automate peut accepter un mot en examinant seulement un facteur de celui-ci.

Dans notre modèle, on représente de plus le mouvement de la tête de lecture dans un état, qui peut être *forward* ou *backward* selon que la tête se déplacera vers la gauche ou vers la droite du mot.

Un autre modèle usuel (e.g. [Rabin et Scott, 1959] ou [Shepherdson, 1959]) est d'avoir un seul type d'état, et d'avoir des transitions étiquetées par le mouvement de la tête de lecture : E est une fonction de $Q \times A \times \{\leftarrow, \rightarrow\} \times Q$ dans \mathbb{K} . On peut aussi voir E comme deux fonctions, E_+ et E_- .

Dans ce modèle, la fonction de transition est une fonction de $Q \times A \times Q \times \{-1,+1\}$ dans \mathbb{K} , un chemin est défini comme un triplet (p,e,q), avec p et q deux états, et e une suite de transitions de la forme $(p_i,a_i,q_i,m_i)_{[\![1,k]\!]}$ telle que $p=p_1, q=q_k$, et pour tout i de $[\![1,k-1]\!], q_i=p_{i+1}$. L'étiquette du chemin $(p,(p_i,a_i,q_i,m_i)_{i\in[\![1,k]\!]},q)$ existe et est le mot $w=w_1\ldots w_n$ s'il existe une application π de $[\![1,k]\!]$ vers $[\![1,n]\!]$ telle que :

•
$$Im(\pi) = [1, n],$$

• pour tout i dans $[1, k-1], w_{\pi(i+1)} = w_{\pi(i)+m_i}$.

Un calcul d'étiquette w est un chemin tel que p est initial et q est final d'étiquette dans $(\vdash +\varepsilon)w(\varepsilon + \dashv)$.

Tous ces modèles sont équivalents (voir [Pécuchet, 1985]). Nous montrerons en sous-section 2.3.6 comment passer d'un automate dans le modèle dont les transitions stockent les mouvements de la tête de lecture en un automate dans le modèle où cette information est dans les états, et vice-versa. Pour l'équivalence des autres modèles, il est immédiat que finir à gauche ou à droite lorsqu'on dispose d'un automate dans l'autre modèle, ou qui finit n'importe où, est faisable : il suffit de rajouter un état après tous les états finaux, qui emmène la tête de lecture à l'extrémité souhaitée.

Il peut être nécessaire de rajouter un état *forward* ensuite, si l'on rembobinait avec un état *backward*, afin qu'il soit final.

Constructions

Dans cette section, nous verrons diverses opérations et constructions sur les automates. En général, elles seront définies sur les K-automates classiques, mais on pourra facilement étendre ces définitions aux cas des automates circulaires, à navette, et boustrophédons.

Les transformations décrites dans cette partie préservent les calculs; leur correction ne dépend pas de la distributivité des poids : elles sont donc valides dans le cas où $\mathbb K$ est un bi-monoïde.

Revêtement

Les manipulations d'automates que nous allons étudier sont toutes basées sur le même concept : transformer un automate en un automate équivalent vérifiant de plus une certaine propriété. Dans le cas des automates unidirectionnel booléens, une méthode usuelle est l'utilisation d'un morphisme entre deux automates, c'est-à-dire une application entre les états, qui respecte les transitions.

Ces applications ne sont pas adaptées au cas pondéré. Nous étendons ici la définition du revêtement d'automates de [Sakarovitch, 2009] aux automates boustrophédons. Cette définition est inspirée de la notion de *covering* sur les graphes, due à [Stallings, 1983]. Le revêtement est lié à la notion plus classique de bisimulation. Plus précisément, on peut dire de deux automates qu'ils sont en bisimulation s'ils sont tous deux des revêtements d'un troisième.

Définition 2.14. Un revêtement (en anglais covering) d'un automate $\mathcal{A} = (Q_+, Q_-, E, I, T)$ est un automate $\mathcal{B} = (R_+, R_-, F, J, U)$ pour lequel il existe une surjection φ de $R = R_+ \cup R_-$ dans $Q = Q_+ \cup Q_-$ telle que :

- φ préserve le type des états : $\forall p \in R, p \in R_+ \Leftrightarrow \varphi(p) \in Q_+$.
- Si $I(q) \neq 0$, alors il existe un unique p antédédent de q par φ tel que $J(p) \neq 0$. Dans ce cas, I(q) = J(p).
- Pour tous états $q_1, q_2 \in Q$, pour toute lettre a, tels que $E(q_1, a, q_2) \neq 0$, pour tout antécédent p_1 de q_1 , il existe un unique p_2 antécédent de q_2 tel que $F(p_1, a, p_2) \neq 0$.
- Pour tous états $p_1, p_2 \in R$, pour toute lettre a, si $F(p_1, a, p_2) \neq 0$, $E(\varphi(p_1), a, \varphi(p_2)) = F(p_1, a, p_2)$.
- $U(p) = T(\varphi(p)).$

Proposition 5. Soit A un automate et B un automate, revêtement de A. Il existe une bijection qui respecte les poids entre les calculs de A et les calculs de B.

Démonstration. Considérons l'application ψ des transitions de \mathcal{B} vers celles de \mathcal{A} , qui envoie (p_1, a, p_2) sur $(\varphi(p_1), a, \varphi(p_2))$.

Soit un calcul de \mathcal{A} . Ce calcul a un antécédent, et il est unique. En effet, l'état initial admet un unique antécédent, initial dans \mathcal{B} . Et pour toute transition, si l'on fixe l'antécédent de la source, alors l'antécédent de la cible est unique.

C'est donc bien une bijection, et elle respecte bien les poids.

Corollaire 1. Un automate est valide si et seulement si un revêtement de celui-ci est valide. Le cas échéant, ils sont équivalents.

Circulation

Définition 2.15. Soit $\mathcal{A} = (Q, E, I, T)$ un \mathbb{K} -automate. Une *circulation* sur cet automate est une fonction de Q dans \mathbb{K} . Appliquer une circulation c sur \mathcal{A} revient à construire un automate $\mathcal{B} = (Q, F, J, U)$ tel que, pour tout état $p \in Q$:

- $\bullet \ I(p) = J(p).c(p),$
- U(p) = c(p).T(p),
- Pour tout état q, c(p).E(p, a, q) = F(p, a, q).c(q).

Proposition 6. Le comportement d'un automate est invariant par circulation.

Démonstration. Soit w un mot de A^* , et $\rho = (p_1, (e_i)_{i \in [\![1,k]\!]}, p_{k+1})$ un calcul de \mathcal{A} sur w. Soit \mathcal{P}_w le poids de ce calcul. Avec $e_i = (p_i, w_i, p_{i+1})$, on a :

$$\mathcal{P}_w = I(p_1) \cdot \prod_{i=1}^k E(p_i, w_i, p_{i+1}) \cdot T(p_{k+1}). \tag{2.1}$$

Par définition de \mathcal{B} , on a donc :

$$\mathcal{P}_w = J(p_1).c(p_1).\prod_{i=1}^k E(p_i, w_i, p_{i+1}).T(p_{k+1})$$

:

$$= J(p_1). \prod_{i=1}^r F(p_i, w_i, p_{i+1}).c(p_{r+1}). \prod_{i=r+1}^k E(p_i, w_i, p_{i+1}).T(p_{k+1})$$

:

$$= J(p_1) \cdot \prod_{i=1}^{k} F(p_i, w_i, p_{i+1}) \cdot c(p_{k+1}) \cdot T(p_{k+1})$$

$$= J(p_1) \cdot \prod_{i=1}^{k} F(p_i, w_i, p_{i+1}) \cdot U(p_{k+1})$$

C'est bien le poids du même calcul dans \mathcal{B} . Ainsi, le poids d'un calcul ne change pas par circulation, et non plus celui d'un mot. Les deux automates sont donc équivalents.

Remarque 16. Si \mathbb{K} est inversible, alors pour tout automate \mathcal{A} , pour toute circulation c sur cet automate, il existe un unique automate \mathcal{B} tel qu'il soit le résultat de l'application de c sur \mathcal{A} .

Ces deux constructions, le revêtement et la circulation, vont nous permettre de prouver l'équivalence avant et après les transformations que nous allons décrire maintenant.

Automate caractéristique

Définition 2.16. Un automate est dit *caractéristique* si toute transition a un poids égal à 1.

Proposition 7. Si K est localement fini, alors toute série réalisée par un K-automate peut être réalisée par un K-automate caractéristique.

Démonstration. Soit $\mathcal{A} = (Q, E, I, T)$ un automate unidirectionnel pondéré sur \mathbb{K} localement fini (cf. section 1.3 page 11). On effectue le produit de l'automate \mathcal{A} avec la partie finie \mathbb{K}' engendrée par les poids de l'automate.

Plus précisément, considérons l'automate $\mathcal{B}=(Q\times\mathbb{K}',E',I',T')$ défini comme suit :

- E'((p, x), a, (q, y)) = E(p, a, q) si et seulement si y = xE(p, a, q).
- I'(p, x) = x si I(p) = x, 0 sinon,
- T'(p, x) = T(p).

Montrons que cet automate est un revêtement de \mathcal{A} . Posons φ la surjection qui à tout état (p, x) de \mathcal{B} associe p de \mathcal{A} .

Si I(q) n'est pas nul, alors il existe un unique antécédent par φ dont l'image par I' n'est pas nulle, c'est (q, I(q)), et on a bien I'(q, I(q)) = I(q).

Soit p, q, a, k tels que $E(p, a, q) = k \neq 0$, et (p, x) un antécédent de p. Alors il existe un seul état (q, y) tel que $E'((p, x), a, (q, y)) \neq 0$. Il s'agit de (q, xk).

De plus, pour tous les états (p, x) et (q, y) de \mathcal{B} , si $E'((p, x), a, (q, y)) \neq 0$, alors y = xE(p, a, q) et E'((p, x), a, (q, y)) = E(p, a, q).

Comme de plus on a que T'(p, x) = T(p), l'automate \mathcal{B} est bien un revêtement de \mathcal{A} .

Soit maintenant l'automate $\mathcal{C} = (Q \times \mathbb{K}'', E'', I'', T'')$:

- E''((p, x), a, (q, y)) = 1 si x.E(p, a, q) = y, et 0 sinon,
- I''(p, x) = 1 si I(p) = x, 0 sinon,
- T''(p, x) = xT(p).

On a bien appliqué sur \mathcal{B} la circulation c telle que c(p,x)=x. En effet, on a bien :

- I'(p, x) = x = I''(p, x)c(p, x),
- T''(p,x) = xT(p) = c(p,x)T'(p,x), et
- si E'((p, x), a, (q, y)) = k alors k = E(p, a, q), y = xk, et E''((p, x), a, (q, y)) = 1.

L'automate \mathcal{C} est donc équivalent à \mathcal{A} .

Automates positifs sur \mathbb{Q} et \mathbb{R}

Définition 2.17. Un \mathbb{K} -automate est dit *standard* s'il existe un seul état initial p, avec I(p) = 1, et tel que p n'ait aucune transition entrante.

On a la proposition classique suivante :

Proposition 8. Tout K-automate est équivalent à un K-automate standard.

Définition 2.18. Un \mathbb{Q} -automate ou \mathbb{R} -automate standard est dit *positif* si toute transition a un poids positif.

Lemme 1. Toute série réalisable par un \mathbb{R} -automate boustrophédon est réalisable par un \mathbb{R} -automate boustrophédon standard positif.

Démonstration. Soit $\mathcal{A} = (Q_+, Q_-, E, I, T)$ un \mathbb{R} -automate boustrophédon réalisant une série s, que l'on peut considérer standard par la proposition s. Considérons l'automate suivant $\mathcal{B} = (Q_+ \times \{1, -1\}, Q_- \times \{1, -1\}, E', I', T')$:

- I'(p,1) = I(p) si I(p) est positif, I'(p,-1) = I(p) sinon.
- E'((p, x), a, (q, x)) = E(p, a, q) si E(p, a, q) est positif,
- E'((p, x), a, (q, y)) = -E(p, a, q) si E(p, a, q) est négatif, et que $x \neq y$,
- T'(p, x) = xT(p).

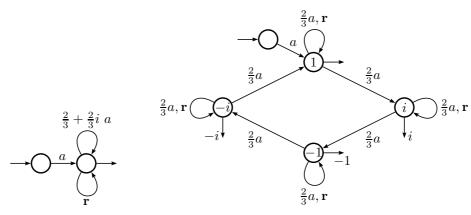
Cet automate est la circulation d'un revêtement de \mathcal{A} , il lui est donc équivalent. À part peut-être les poids finaux, tous les poids sont positifs. Comme \mathcal{A} est standard, et en posant i son état initial, \mathcal{B} n'a qu'un seul état initial, (i,1), initial de poids 1. \mathcal{B} est donc standard.

Proposition 9. Toute série s réalisée par un \mathbb{R} -automate boustrophédon est décomposable en deux séries s_+, s_- , chacune réalisable par un \mathbb{R}_+ -automate positif boustrophédon standard telles que $s=s_+-s_-$.

Démonstration. Soit $\mathcal{A} = (Q_+, Q_-, E, I, T)$ un \mathbb{R} -automate boustrophédon réalisant une série s, et \mathcal{B} l'automate construit dans la preuve du lemme 1 ci-dessus.

On peut considérer deux automates, \mathcal{B}_+ et \mathcal{B}_- , copies de \mathcal{B} sauf pour la fonction finale, restreinte aux éléments de poids final positif pour \mathcal{B}_+ et aux éléments de poids final négatif pour \mathcal{B}_- . De plus, on multiplie par -1 tous les poids finaux de \mathcal{B}_- , afin d'obtenir un automate dont tous les poids sont positifs.

La série réalisée par \mathcal{A} est immédiatement la différence de la série réalisée par \mathcal{B}_+ et de celle réalisée par \mathcal{B}_- . Ces deux automates sont pondérés sur \mathbb{R}_+ .



- (a) Un automate circulaire sur \mathbb{C} .
- (b) Cet automate n'est pas valide.

FIGURE 2.7 – Contre-exemple de l'extension à \mathbb{C} de la proposition 9.

Remarque 17. Cette construction peut s'étendre à \mathbb{C} dans le cas des automates unidirectionnels, pour obtenir une série réelle et une série imaginaire pure. Cependant, comme il n'y a pas bijection des calculs, la construction ne conserve pas nécessairement la validité dans le cas des \mathbb{C} -automates circulaires, ainsi qu'illustré dans l'exemple 5.

Exemple 5

L'automate de la Figure 2.7a est valide, cependant si l'on construit un automate en suivant l'idée de la construction de la proposition 9, on obtient l'automate de la Figure 2.7b, qui n'est pas valide. En effet, pour le mot a, par exemple, il existe exactement 2^{n+1} calculs d'étiquette $(a\mathbf{r})^n a$, chacun dont le module du poids vaut $\left(\frac{2}{3}\right)^{n+1}$. La somme des modules des poids des calculs d'étiquette $(a\mathbf{r})^n a$ est donc $\left(\frac{4}{3}\right)^{n+1}$. Cela signifie que la somme des modules des poids des calculs sur a est donc $\sum_{n=0}^{\infty} \left(\frac{4}{3}\right)^n$, qui n'est pas définie. L'automate n'est donc pas valide.

δ -normalisation

Définition 2.19. Un automate boustrophédon (F, B, E, I, T) est dit δ -normalisé si F et B sont respectivement partionnés en $F = F_+ \cup F_-$ et $B = B_+ \cup B_-$, tels que :

- tous les états finaux sont dans F_+ ,
- pour tout état p, pour toute transition (p, a, q), q appartient à F si et seulement si p appartient à $F_+ \cup B_+$.

Proposition 10. Tout automate boustrophédon est équivalent à un automate δ -normalisé.

Démonstration. Soit $\mathcal{A} = (F, B, E, I, T)$ un automate boustrophédon sur \mathbb{K} .

L'idée est de faire deux copies des états de l'automate, l'une où l'on ne garde que les transitions provenant des états *forward*, l'autre celles provenant des états *backward*.

Soit $\mathcal{B} = (F \times \{1, 2\}, B \times \{1, 2\}, E', I', T')$ un automate sur \mathbb{K} avec :

- I'(p, i) = I(p), si i = 1, 0 sinon.
- $\bullet \ T'(p,i) = T(p).$
- E'((p,i), a, (q,1)) = E(p, a, q) si p appartient à F, 0 sinon.
- E'((p,i), a, (q,2)) = E(p, a, q) si p appartient à B, 0 sinon.

Montrons que cet automate est un revêtement de \mathcal{A} . Considérons φ la surjection qui à tout état (p, i) associe l'état p.

Soit q tel que I(q) est non nul. Alors il existe un unique antécédent de q qui soit initial, c'est (p, 1), de poids initial I(p).

Soit p, q et k non nul tel que $E(p, a, q) = k \neq 0$. Soit (p, i) un antécédent de p. Alors il existe un unique antécédent de q, noté (q, j), tel que E'((p, i), a, (q, j)) est différent de 0, c'est (q, 1) si p appartient à F, (q, 2) sinon.

Soit (p, i), (q, j) deux états de $F \times \{1, 2\} \cup B \times \{1, 2\}$. Si E'((p, i), a, (q, j)) est non nul, alors E'((p, i), a, (q, j)) = E(p, a, q).

De plus, pour tout $p \in F \cup B$, T'(p,i) = T(p). Il s'agit donc bien d'un revêtement.

Équivalence des modèles des automates boustrophédons

Nous allons montrer que le modèle que nous utilisons et le modèle dans lequel les mouvements de la tête de lecture dépendent des transitions sont équivalents, en donnant une transformation permettant de passer d'un modèle à un autre.

Soit $\mathcal{A} = (Q_+, Q_-, E, I, T)$ un automate boustrophédon sur \mathbb{K} . L'automate $\mathcal{B} = (Q_+ \cup Q_-, E_+, E_-, I, T)$ lui est équivalent, en posant $E_+(p, a, q) = E(p, a, q)$ si $q \in Q_+$, et $E_-(p, a, q) = E(p, a, q)$ si $q \in Q_-$. La transformation inverse ne marche que si les états n'admettent qu'un seul type de transition entrante. On peut au besoin dupliquer les états pour lesquels c'est nécessaire, afin de séparer les transitions entrantes forward des transitions entrantes backward.

Exemple 6

Les Figures 2.8a et 2.9a présentent deux automates, obtenus l'un à partir de l'autre, selon la transformation ci-dessus. On peut voir l'effet de cette tranformation sur les calculs, avec les Figures 2.8b et 2.9b.

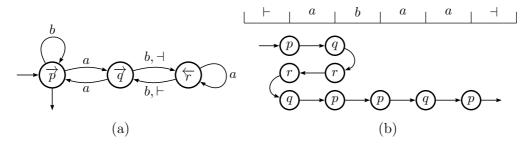


FIGURE 2.8 – Un automate boustrophédon \mathcal{A}_0 et un de ses calculs.

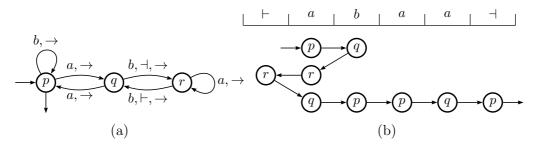


FIGURE 2.9 – Un automate boustrophédon \mathcal{B}_0 équivalent à \mathcal{A}_0 et un de ses calculs.

La transformation décrite ci-dessus est naturellement un revêtement. Il existe donc une bijection entre les calculs de \mathcal{A} et ceux de \mathcal{B} , la validité est donc conservée. Les deux modèles ont donc bien la même expressivité.

36 L.-M. Dando

Chapitre 3

Automates circulaires sur les semi-anneaux rationnellement additifs

"Ce qu'on a fait, on le refait, l'histoire est comme un cercle immense."

> - Offenbach, La Grande-Duchesse de Gérolstein

Ce chapitre présente les résultats obtenus pour les automates circulaires pondérés par des semi-anneaux rationnellement additifs, qui ont été pour l'essentiel publiés dans [Dando et Lombardy, 2017]. On rappelle tout d'abord en section 3.1 que sur de tels semi-anneaux, les automates circulaires et les séries de Hadamard sont équivalents [Lombardy, 2016]. Il est donc naturel de chercher des algorithmes pour passer d'une représentation à une autre : c'est le but de ce chapitre, dans lequel on présente une façon d'étendre les algorithmes de conversion entre les séries rationnelles et les automates unidirectionnels.

Cette extension consiste à transformer une expression de Hadamard en une expression rationnelle sur un alphabet étendu, et de transformer cette expression en un automate unidirectionnel, en suivant un des algorithmes classiques, tels que la construction de l'automate des dérivées, ou celle de l'automate des positions. On montre que si l'on considère ensuite l'automate obtenu comme un automate circulaire, on réalise encore la même série. Différents algorithmes pour cette conversion sont décrits dans la section 3.3.

La transformation réciproque est moins directe, car il faut pouvoir retransformer l'expression rationnelle étendue en une expression de Hadamard, ce qui ne peut se faire que si cette expression est d'une certaine forme. Les détails de cette transformation sont expliqués dans la section 3.4. La dernière section s'intéresse au cas des automates pondérés sur \mathbb{Q} , \mathbb{R} ou \mathbb{C} . Bien que ces derniers ne soient pas rationnellement additifs, ils présentent des propriétés intéressantes : on montre que les séries de Hadamard sont exactement les quotients de Hadamard de séries rationnelles (proposition 28). On montre également que le problème de l'équivalence de deux automates circulaires valides est décidable (théorème 3.3).

La section 3.2 introduit les différents éléments dont nous allons avoir besoin dans la suite. Dans tout ce chapitre, nous noterons \mathbb{K} un semi-anneau rationnellement additif.

Équivalence séries de Hadamard - Automates circulaires

Dans les semi-anneaux rationnellement additifs, les automates circulaires, et même les boustrophédons, ont un comportement toujours défini [Lombardy, 2016]. Cela provient essentiellement du fait que dans un semi-anneau rationnellement additif, l'étoile d'un élément est toujours définie. On montre que ce comportement est une série de Hadamard, et réciproquement.

Proposition 11. Toute série de Hadamard sur \mathbb{K} , un semi-anneau rationnel-lement additif, est le comportement d'un automate circulaire sur \mathbb{K} : \mathbb{K} Had $A^* \subseteq \mathbb{R}$ \mathbb{K} AutA.

 $D\'{e}monstration$. Une série de Hadamard est dans la clôture des séries rationnelles par les opérations terme à terme. Montrons donc le résultat par induction. Soit s une série de Hadamard.

• La série s est rationnelle.

Au chapitre 2, nous avons vu que le théorème 2.1 dit de Kleene-Schützenberger ([Schützenberger, 1961]) affirmait que toute série rationnelle est le comportement d'un K-automate unidirectionnel. Comme ces automates forment un sous-ensemble des automates circulaires, la proposition est vraie dans ce cas.

Dans la suite, on supposera la proposition vraie pour s_1 et s_2 , deux séries de Hadamard. On notera $\mathcal{A}_1 = (Q, E, R, I, T)$ et $\mathcal{A}_2 = (Q', E', R', I', T')$ les automates circulaires dont les séries s_1 et s_2 sont les comportements, tels qu'illustrés dans la Figure 3.1.

De plus, on utilisera les abus de notation suivants :

Si X est un sous-ensemble de Y, et f une fonction de X dans \mathbb{K} , alors on étend naturellement f à Y avec $f(y) = 0_{\mathbb{K}}$ quel que soit $y \in Y \setminus X$.

Si X' est également un sous-ensemble de Y, avec f' une fonction de X' dans \mathbb{K} , alors on note (f+f')(y)=f(y)+f'(y) pour tout y dans Y.

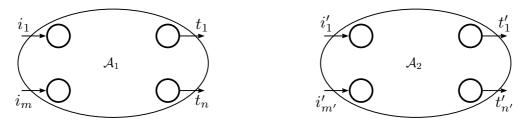


FIGURE 3.1 – Deux automates, A_1 réalisant s_1 , et A_2 réalisant s_2 .

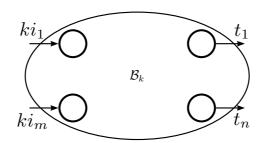


FIGURE 3.2 – L'automate réalisant ks_1 .

 \bullet La série s est le produit de s_1 par un élément k du semi-anneau.

L'automate $\mathcal{B}_k = (Q, E, R, J, T)$ réalise la série s, en posant pour tout état q de Q, J(q) = kI(q). Cette construction est représentée dans la Figure 3.2. (Si le semi-anneau n'est pas commutatif, et que $s = s_1 k$, alors on multiplie les sorties, et non les entrées, de l'automate).

• La série s est la somme des séries s_1 et s_2 .

L'automate $\mathcal{A}_1 \cup \mathcal{A}_2 = (Q \cup Q', E + E', R + R', I + I', T + T')$ réalise la série s. En effet, pour tout mot w, l'ensemble des calculs pour w est l'union disjointe de l'ensemble des calculs pour w sur \mathcal{A} et de ceux sur \mathcal{B} . Ainsi, le poids de w dans $\mathcal{A} \cup \mathcal{B}$ est bien la somme du poids de w dans s_1 et dans s_2 .

Cette construction est représentée dans la Figure 3.3.

• La série s est le produit de Hadamard de s_1 et s_2 .

L'automate $\mathcal{A}_1 \odot \mathcal{A}_2 = (Q \cup Q', E + E', R + R' + S, I, T')$, avec S(p, q) = T(p).I'(q) pour tout p dans Q et q dans Q', réalise la série s. En effet, pour tout mot w, pour tout paire de calculs $(p, (e_i), q)$ de \mathcal{A}_1 et $(p', (e'_j), q')$ de \mathcal{A}_2 , il existe un calcul $(p, (f_k), q')$ dans $\mathcal{A}_1 \odot \mathcal{A}_2$, où la suite f est définie ainsi :

$$f_i = e_i \text{ pour } i \in [1, n], f_{n+1} = (q, p'), f_{n+1+j} = e'_j \text{ pour } j \in [1, m],$$

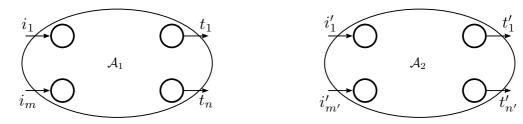


FIGURE 3.3 – L'automate réalisant $s_1 + s_2$.

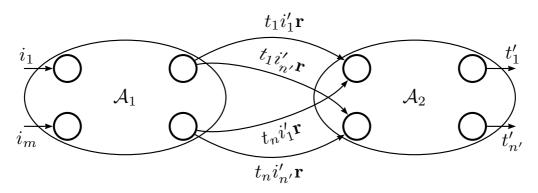


FIGURE 3.4 – L'automate réalisant $s_1 \odot s_2$.

avec n la longeur de la suite e, et m celle de e'. Le poids de ce nouveau calcul est bien le produit des poids des calculs initiaux.

De plus, tout calcul de $\mathcal{A}_1 \odot \mathcal{A}_2$ est nécessairement de cette forme. Ainsi, on a bien $\mathcal{A}_1 \odot \mathcal{A}_2(w) = \mathcal{A}_1(w) \cdot \mathcal{A}_2(w)$.

Cette construction est représentée dans la Figure 3.4.

• La série s est l'étoile de Hadamard de la série s_1 .

Soit l'automate $\mathcal{A}_1^{\circledast} = (Q \cup \{i\}, E + F, R + S, I + \chi_i, T + \chi_i)$, avec F et S telles que $\forall a \in A, F(i, a, i) = 1_{\mathbb{K}}, \ \forall (p, q) \in Q, S(p, q) = T(p).I(q)$, et χ_i la fonction caractéristique de i.

Cette construction est représentée dans la Figure 3.5.

Cet automate réalise bien la série s. En effet, en posant w un mot de A^* , pour tout l entier positif, pour chaque liste de calculs $(p^{(j)}, (e_m^{(j)}), q^{(j)})_{j \in [\![1,l]\!]}$ dans \mathcal{A} , il existe un calcul $(p^{(1)}, (f_k), q^{(l)})$ dans $\mathcal{A}_1^{\circledast}$ telle que la suite f est constituée de la concaténation des suites $e^{(j)}$ dans l'ordre, les suites $e^{(j)}$ et $e^{(j+1)}$ étant séparées par la \mathbf{r} -transition $(q^{(j)}, p^{(j+1)})$.

Le poids d'un tel calcul est naturellement le produit des poids des l calculs de la liste. Si l'on groupe ces calculs en fonction de l, on remarque donc que le poids de ces groupes est la puissance l de la série s.

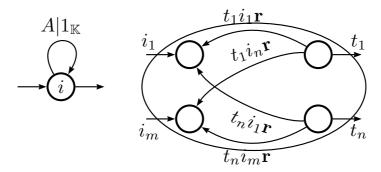


FIGURE 3.5 – L'automate réalisant s_1^{\circledast} .

Or, un calcul sur $\mathcal{A}_1^{\circledast}$ peut être de deux formes : soit une telle suite avec l-1 **r**-transitions de S, soit une boucle sur l'état i.

Ce dernier calcul est unique pour w, et il est de poids 1 : l'automate restreint à i calcule la puissance 0 de la série s.

Ainsi, le poids d'un mot w dans $\mathcal{A}_1^{\circledast}$ est de 1 plus le poids de w dans \mathcal{A}_1 à la puissance l, et ce pour chaque entier positif l.

On a donc bien $\langle \mathcal{A}_1^{\circledast}, w \rangle = \langle \mathcal{A}_1, w \rangle^*$.

On a également la propriété dans l'autre sens :

Proposition 12. Soit \mathbb{K} un semi-anneau rationnellement additif. Tout \mathbb{K} -automate circulaire est valide, et son comportement est une série de Hadamard : $\mathsf{RKAut}A \subseteq \mathsf{KHad}A^*$.

Démonstration. Soit $\mathcal{A} = (Q, E, R, I, T)$ un \mathbb{K} -automate circulaire, avec Q ordonné de taille n, que l'on notera par simplicité [1; n].

Pour montrer la proposition, nous allons tout d'abord montrer par induction qu'une certaine somme est définie, puis exprimer le comportement de \mathcal{A} en fonction de cette somme.

Nous aurons besoin de quelques définitions : soit w un mot de A^* , $k \in \mathbb{N}$, et π un chemin de \mathcal{A} d'étiquette $u = (w\mathbf{r})^k$.

Pour tout $j \in [1, k]$, on note s_j l'état rencontré après avoir lu le $j^{\text{ème}}$ **r** dans le chemin π . On définit la fonction **ordre** sur un chemin comme le plus grand des états s_j rencontré le long de ce chemin.

Plus précisément,

$$\mathsf{ordre}(\pi) = \begin{cases} 0 \text{ si } k = 0 \text{ ou } k = 1, \\ \max\{s_j \mid j \in [\![1,k-1]\!]\} \text{ sinon}. \end{cases}$$

Notez que dans le cas général, on ne tient compte ni de s_0 ni de s_k , qui sont respectivement le premier et le dernier état de π .

Soit $\mathcal{P}(w, i, p, q)$ l'ensemble des chemins dont l'étiquette est dans $(w\mathbf{r})^*$, d'ordre au plus i, et dont le premier état est p et le dernier q. Soit S(w, i, p, q) la somme des poids des chemins de $\mathcal{P}(w, i, p, q)$.

Montrons par induction sur i que pour toute paire d'états (p,q), la somme S(w,i,p,q) est bien définie.

- Supposons que i = 0. Alors, $\operatorname{ordre}(\pi) = 0$, et soit π est de taille 0, soit il est de taille |w| + 1. Dans les deux cas, $\mathcal{P}(w, 0, p, q)$ est fini quels que soient p et q, et la somme S(w, 0, p, q) est bien définie.
- Supposons que pour tout j < i, pour tout p et q, S(w, j, p, q) est bien définie.

On peut décrire de manière unique $\mathcal{P}(w,i,p,q)$ en considérant la concaténation de chemins comme un produit :

$$\mathcal{P}(w, i, p, q) = \mathcal{P}(w, i - 1, p, q)$$

$$\cup \mathcal{P}(w, i - 1, p, i).\mathcal{P}(w, i - 1, i, i)^*.\mathcal{P}(w, i - 1, i, q).$$
(3.1)

On a donc:

$$S(w, i, p, q) = S(w, i - 1, p, q) + S(w, i - 1, p, i).S(w, i - 1, i, i)^*.S(w, i - 1, i, q).$$
(3.2)

Dans un semi-anneau rationnellement additif, l'étoile de Kleene est toujours définie, donc S(w, i, p, q) est bien définie.

Ainsi, par induction, la somme S(w, i, p, q) est bien définie quels que soient p et q. Montrons maintenant que le comportement de \mathcal{A} est une série de Hadamard.

Notons tout d'abord F(w, p, q) la somme des chemins de p à q d'étiquette w. Cette somme existe car ces chemins sont en nombre fini.

On peut définir deux séries à partir des deux sommes F et S: la série s(i,p,q) telle que $\langle s(i,p,q),w\rangle = S(w,i,p,q)$, et la série f(p,q) telle que $\langle f(p,q),w\rangle = F(w,p,q)$. On peut remarquer que les séries f(p,q) et s(0,p,q) sont rationnelles pour tout couple (p,q).

De plus, par l'équation (3.2), pour tout i dans [1, n], on a :

$$s(i, p, q) = s(i - 1, p, q) + s(i - 1, p, i) \odot s(i - 1, i, i)^{\circledast} \odot s(i - 1, i, q).$$
(3.3)

Ainsi, par une induction immédiate, s(n, p, q) est une série de Hadamard.

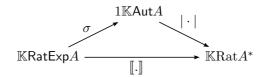


FIGURE 3.6 – Un algorithme σ de conversion.

De plus, on peut exprimer le poids du mot w dans A en fonction de F et de S. Ces deux sommes étant bien définies, le poids $\langle A, w \rangle$ du mot w est également bien défini, et est égal à :

$$\langle \mathcal{A}, w \rangle = \sum_{p,q,r \in Q} I(p).S(w, n, p, q).F(w, q, r).T(r). \tag{3.4}$$

On peut donc exprimer le comportement de A:

$$|\mathcal{A}| = \sum_{p,q,r \in Q} I(p).s(n,p,q) \odot f(q,r).T(r), \tag{3.5}$$

qui est une série de Hadamard.

Les séries réalisées par les automates circulaires sont donc exactement les séries de Hadamard (cf. [Lombardy, 2016]).

On remarque que la preuve de la propriété précédente peut s'étendre à \mathbb{Q} , \mathbb{R} , ou \mathbb{C} sur les automates valides :

Proposition 13. Soit \mathbb{K} égal à \mathbb{Q} , \mathbb{R} , ou \mathbb{C} . Le comportement de tout automate circulaire valide sur \mathbb{K} est une série de Hadamard sur \mathbb{K} .

En effet, sur \mathbb{R} ou \mathbb{C} , si $(x_i)_{i\in I}$ est une famille sommable, alors pour toute partition (I_j) de I, toute famille $(x_i)_{i\in I_j}$ est sommable. On a donc bien tous les S(w, i, p, q) de la preuve de la proposition 12 définis.

Cet argument n'est pas suffisant sur \mathbb{Q} , car ce dernier n'est pas complet. Cependant, on obtient quand même le résultat que le comportement d'un \mathbb{Q} -automate valide est une série de Hadamard sur \mathbb{R} , dont tous les coefficients sont obtenus grâce à des opérations rationnelles sur des éléments de \mathbb{Q} : la série obtenue est donc une série de Hadamard sur \mathbb{Q} .

Aspects algorithmiques : notions nécessaires

Un algorithme σ de conversion d'une expression rationnelle en un automate unidirectionnel est une application rendant commutatif le graphe de la Figure 3.6.

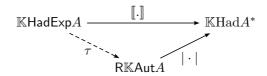


FIGURE 3.7 – On cherche un algorithme τ .

Dans ce chapitre, nous cherchons à construire un algorithme τ convertissant une expression de Hadamard en un automate circulaire, de façon à rendre commutatif le graphe de la Figure 3.7.

Le but de cette section est de définir les outils dont nous allons avoir besoin afin d'effectuer cette construction.

Comment voir les automates circulaires

Un automate circulaire peut être vu comme un automate unidirectionnel sur un alphabet doté d'une lettre spéciale, \mathbf{r} , qui sert à "rembobiner" le mot d'entrée. Le calcul des poids des mots est effectué suivant la définition 2.3. On notera $A_{\mathbf{r}} = A \cup \{\mathbf{r}\}$.

Dans cette optique, définissons les deux fonctions qui permettent de passer de la sémantique unidirectionnelle à la sémantique circulaire.

Définition 3.1. Soit $\mathcal{A} = (Q, E, R, I, T)$ un automate circulaire sur A. On note $\mathbf{1w}(\mathcal{A}) = (Q, E', I, T)$ l'automate unidirectionnel sur $A_{\mathbf{r}}$ tel que :

- E'(p, a, q) = E(p, a, q) pour tout $a \in A$, et
- $E'(p, \mathbf{r}, q) = R(p, q)$.

De même, avec $\mathcal{B} = (Q, E, I, T)$ un automate unidirectionnel sur $A_{\mathbf{r}}$, on note $\mathcal{B}^{\mathbf{r}}$ voire $\mathsf{rot}(\mathcal{B})$ l'automate circulaire (Q, E', R, I, T), avec :

- E'(p, a, q) = E(p, a, q) pour tout $a \in A$, et
- $R(p,q) = E(p, \mathbf{r}, q)$.

On a naturellement $\mathcal{A} = \mathsf{rot}(\mathsf{1w}(\mathcal{A}))$ et $\mathcal{B} = \mathsf{1w}(\mathsf{rot}(\mathcal{B}))$, avec \mathcal{A} un automate circulaire sur A et \mathcal{B} un automate unidirectionnel sur $A_{\mathbf{r}}$.

On a également immédiatement pour tout mot w de A^* :

$$\langle \mathcal{A}, w \rangle = \sum_{i=0}^{\infty} \langle 1 w(\mathcal{A}), w(\mathbf{r}w)^i \rangle,$$
 (3.6)

$$\langle \mathcal{B}^{\mathbf{r}}, w \rangle = \sum_{i=0}^{\infty} \langle \mathcal{B}, w(\mathbf{r}w)^i \rangle.$$
 (3.7)

On aimerait cependant savoir l'effet de ces opérations sur la série reconnue par l'automate en paramètre : c'est le but de la définition suivante.

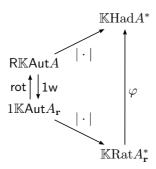


FIGURE 3.8 – La fonction φ respecte 1w et rot.

Définition 3.2. Soit φ la fonction linéaire de $\mathbb{K}\langle\langle A_{\mathbf{r}}^* \rangle\rangle$ vers $\mathbb{K}\langle\langle A^* \rangle\rangle$ telle que

$$\forall s \in \mathbb{K}\langle\langle A_{\mathbf{r}}^* \rangle\rangle, \qquad \varphi\bigg(\sum_{u \in A_{\mathbf{r}}^*} \langle s, u \rangle u\bigg) = \sum_{w \in A^*} \bigg(\sum_{u \in w(\mathbf{r}w)^*} \langle s, u \rangle\bigg) w. \tag{3.8}$$

Cette fonction est partiellement définie. Plus précisément, $\varphi(s)$ est définie si et seulement si pour tout mot w de A^* , la famille $(\langle s, u \rangle)_{u \in w(\mathbf{r}w)^*}$ est sommable.

La fonction φ associe à une série s sur $A_{\mathbf{r}}$ la série sur A pour laquelle le poids d'un mot est la somme des poids des mots de la forme $w(\mathbf{r}w)^i$ dans s.

On a la proposition suivante:

Proposition 14. Soit \mathbb{K} un semi-anneau rationnellement additif, et \mathcal{A} un \mathbb{K} -automate circulaire valide sur A. L'image par φ de $|\mathbf{1w}(\mathcal{A})|$ est définie et vaut $|\mathcal{A}|$.

Démonstration. Pour chaque mot w, la somme des poids des calculs de 1w(A) étiquetés par un mot de $w(\mathbf{r}w)^*$ est bien définie : il s'agit d'une somme finie. Comme \mathbb{K} est rationnellement additif, la somme des poids des calculs de 1w(A) des mots de $w(\mathbf{r}w)^*$ existe et est égale à la somme des poids des calculs de A étiquetés par w.

On obtient immédiatement le corollaire suivant :

Corollaire 2. Soit \mathbb{K} un semi-anneau rationnellement additif et \mathcal{A} un \mathbb{K} -automate unidirectionnel sur A_r .

Si
$$\varphi(|\mathcal{A}|)$$
 existe, alors \mathcal{A}^r est valide et $|A^r| = \varphi(|\mathcal{A}|)$.

Le graphe de la Figure 3.8 est donc commutatif. De plus, comme toute série rationnelle peut être décrite comme le comportement d'un automate unidirectionnel et que toute série de Hadamard est le comportement d'un automate circulaire, on obtient le corollaire suivant :

Corollaire 3. L'image par φ des séries \mathbb{K} -rationnelles sur A_r est l'ensemble des séries de Hadamard sur A pondérées par \mathbb{K} : $\varphi(\mathbb{K}\operatorname{Rat} A_r^*) = \mathbb{K}\operatorname{Had} A^*$.

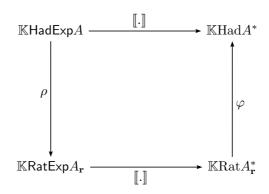


FIGURE $3.9 - \rho$, inverse syntaxique de φ ?

Démonstration. Par le théorème de Kleene-Schützenberger, toute série \mathbb{K} -rationnelle s sur $A_{\mathbf{r}}$ est le comportement d'un automate unidirectionnel \mathcal{A} sur $A_{\mathbf{r}}$. Par le corollaire 2, $\mathsf{rot}(\mathcal{A})$ est valide et l'image de s par φ est donc le comportement de $\mathsf{rot}(\mathcal{A})$, qui est une série de Hadamard, par la proposition 12.

De plus, toute série de Hadamard s' sur A est, par la proposition 11, le comportement d'un automate circulaire valide \mathcal{B} . Par la proposition 14, l'image de $\mathbf{1w}(\mathcal{B})$ par φ est s'.

Expressions pré-Hadamard

À partir de maintenant, et ce jusqu'à la fin de ce chapitre, nous considèrerons que l'étoile de Hadamard n'est plus un opérateur unaire sur les expressions, mais qu'il s'agit d'un opérateur binaire.

Il est défini comme suit :

$$\llbracket F \circledast G \rrbracket = \llbracket F \rrbracket^{\circledast} \odot \llbracket G \rrbracket$$

Remarque 18. Il est immédiat que la grammaire introduite en section 1.5 et celle avec une étoile binaire sont équivalentes. L'utilisation d'une itération de Hadamard binaire nous permettra dans la suite une écriture plus aisée des définitions. De plus, l'étoile de Kleene a été introduite dans [Kleene, 1956] en tant qu'opérateur binaire.

On souhaite désormais avoir un inverse de φ : une transformation syntaxique ρ qui transforme une expression de Hadamard E sur A en une expression rationnelle sur $A_{\mathbf{r}}$ de telle sorte que $[\![\mathsf{E}]\!] = \varphi([\![\rho(\mathsf{E})]\!])$, et que le graphe de la Figure 3.9 soit commutatif.

On définit cet inverse inductivement :

$$\rho(\mathsf{E}) = \mathsf{E} \text{ si } \mathsf{E} \in \mathbb{K} \mathsf{RatExp} A, \tag{3.9}$$

$$\forall k \in \mathbb{K}, \ \rho(k\mathsf{F}) = k\rho(\mathsf{F}), \tag{3.10}$$

46 L.-M. Dando

$$\rho(\mathsf{F}k) = \rho(\mathsf{F})k,\tag{3.11}$$

$$\rho(\mathsf{F} + \mathsf{G}) = \rho(\mathsf{F}) + \rho(\mathsf{G}),\tag{3.12}$$

$$\rho(\mathsf{F} \odot \mathsf{G}) = \rho(\mathsf{F})\mathbf{r}\rho(\mathsf{G}),\tag{3.13}$$

$$\rho(\mathsf{F} \circledast \mathsf{G}) = (\rho(\mathsf{F})\mathbf{r})^* \rho(\mathsf{G}). \tag{3.14}$$

Pour montrer que ρ est bien un inverse de φ , en passant par les interprétations, montrons tout d'abord le lemme suivant :

Lemme 2. Soit E une expression de Hadamard sur A. Pour tout mot w de A^* ,

$$\langle \mathsf{E}, w \rangle = \sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), w(\mathbf{r}w)^i \rangle.$$
 (3.15)

Démonstration. Il est important de noter que les manipulations de somme infinies ne sont ici valides que parce que \mathbb{K} est un semi-anneau rationnellement additif.

La preuve s'effectue par induction sur E.

• E est une expression rationnelle.

On a donc $\rho(\mathsf{E}) = \mathsf{E}$, et donc

$$\sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), (w\mathbf{r})^i w \rangle = \sum_{i=0}^{\infty} \langle \mathsf{E}, (w\mathbf{r})^i w \rangle$$
 (3.16)

$$= \langle \mathsf{E}, w \rangle + \sum_{i=1}^{\infty} \langle \mathsf{E}, (w\mathbf{r})^{i} w \rangle \tag{3.17}$$

$$= \langle \mathsf{E}, w \rangle. \tag{3.18}$$

Supposons maintenant le résultat vrai pour deux expressions de Hadamard F et G.

• E = F + G, E = kF ou E = Fk.

Le résultat est immédiat par linéarité.

• $E = F \odot G$.

$$\langle \mathsf{E}, w \rangle = \langle \mathsf{F} \odot \mathsf{G}, w \rangle$$

$$= \langle \mathsf{F}, w \rangle \langle \mathsf{G}, w \rangle = \left(\sum_{i=0}^{\infty} \langle \rho(\mathsf{F}), (w\mathbf{r})^{i} w \rangle \right) \left(\sum_{j=0}^{\infty} \langle \rho(\mathsf{G}), (w\mathbf{r})^{j} w \rangle \right)$$

$$= \sum_{i,j \in \mathbb{N}^{2}} \langle \rho(\mathsf{F}), (w\mathbf{r})^{i} w \rangle \langle \rho(\mathsf{G}), (w\mathbf{r})^{j} w \rangle$$

$$= \sum_{k=0}^{\infty} \sum_{n=0}^{k} \langle \rho(\mathsf{F}), (w\mathbf{r})^{k-n} w \rangle \langle \rho(\mathsf{G}), (w\mathbf{r})^{n} w \rangle$$

$$= \sum_{k=0}^{\infty} \langle \rho(\mathsf{F}) \mathbf{r} \rho(\mathsf{G}), (w\mathbf{r})^{k+1} w \rangle$$

$$= \sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), (w\mathbf{r})^{i} w \rangle$$

$$(3.19)$$

• $E = F \circledast G$.

On a donc $\langle \mathsf{E}, w \rangle = \langle \mathsf{F}, w \rangle^* \langle \mathsf{G}, w \rangle$. Pour montrer le résultat, et par souci de simplification d'écriture, on étend ici les expressions rationnelles avec la puissance associée au produit de Cauchy, et les expressions de Hadamard avec celle associée au produit de Hadamard. Pour tout entier k, on notera respectivement E^k et $\mathsf{E}^{\odot k}$ les puissances $k^{\grave{e}mes}$ d'une expression E .

On a donc $\llbracket \mathsf{E}^{\odot k} \rrbracket = \llbracket \mathsf{E} \rrbracket^{\odot k}$, et $\langle \mathsf{E}^{\odot k}, w \rangle = \langle \mathsf{E}, w \rangle^k$. On peut montrer facilement par induction sur k, avec le résultat sur $\mathsf{F} \odot \mathsf{G}$, que

$$\langle \mathsf{E}^{\odot k}, w \rangle = \langle \mathsf{E}, w \rangle^k = \sum_{i=0}^{\infty} \langle (\rho(\mathsf{E})\mathbf{r})^k, (w\mathbf{r})^i \rangle.$$
 (3.20)

On peut donc réécrire $\langle \mathsf{F}, w \rangle^*$:

$$\langle \mathsf{F}, w \rangle^* = \sum_{k=0}^{\infty} \langle \mathsf{F}, w \rangle^k = \sum_{k=0}^{\infty} \langle \mathsf{F}^{\odot k}, w \rangle$$

$$= \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^k, (w\mathbf{r})^i \rangle$$

$$= \sum_{i=0}^{\infty} \sum_{k=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^k, (w\mathbf{r})^i \rangle$$

$$= \sum_{i=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^*, (w\mathbf{r})^i \rangle.$$
(3.21)

Ce qui nous permet d'obtenir le résultat suivant pour $\langle E, w \rangle$:

$$\langle \mathsf{E}, w \rangle = \left(\sum_{i=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^*, (w\mathbf{r})^i \rangle \right) \left(\sum_{j=0}^{\infty} \langle \rho(\mathsf{G}), w(\mathbf{r}w)^j \rangle \right)$$

$$= \sum_{i,j \in \mathbb{N}^2} \langle (\rho(\mathsf{F})\mathbf{r})^*, (w\mathbf{r})^i \rangle \langle \rho(\mathsf{G}), w(\mathbf{r}w)^j \rangle$$

$$= \sum_{n=0}^{\infty} \sum_{k=0}^{n} \langle (\rho(\mathsf{F})\mathbf{r})^*, (w\mathbf{r})^{n-k} \rangle \langle \rho(\mathsf{G}), w(\mathbf{r}w)^k \rangle$$

$$= \sum_{n=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^* \rho(\mathsf{G}), w(\mathbf{r}w)^n \rangle$$

$$= \sum_{n=0}^{\infty} \langle \rho(\mathsf{E}), w(\mathbf{r}w)^n \rangle.$$
(3.22)

Ainsi, le lemme est prouvé pour toute expression de Hadamard.

On peut donc maintenant montrer la proposition suivante :

Proposition 15. Soit E une expression de Hadamard. On a $\llbracket \mathsf{E} \rrbracket = \varphi(\llbracket \rho(\mathsf{E}) \rrbracket)$. Démonstration. On a $\llbracket \rho(\mathsf{E}) \rrbracket = \sum_{u \in A^*_{\mathbf{r}}} \langle \rho(\mathsf{E}), u \rangle u$, par définition de l'interprétation.

Donc

$$\varphi(\llbracket \rho(\mathsf{E}) \rrbracket) = \varphi\left(\sum_{u \in A_{\mathbf{r}}^*} \langle \rho(\mathsf{E}), u \rangle u\right)
= \sum_{w \in A^*} \left(\sum_{u \in (w\mathbf{r})^* w} \langle \rho(\mathsf{E}), u \rangle\right) w
= \sum_{w \in A^*} \sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), (w\mathbf{r})^i w \rangle w
= \sum_{w \in A^*} \langle \mathsf{E}, w \rangle w
= \llbracket \mathsf{E} \rrbracket.$$
(3.23)

Ainsi, ρ fonctionne bien comme un inverse de φ , et le graphe de la Figure 3.9 est commutatif.

Caractérisons maintenant les expressions dans l'image de ρ .

Définition 3.3. La grammaire suivante génère les expressions pré-Hadamard, en notant \mathbb{K} RatExpA l'ensemble des expressions rationnelles sur l'alphabet A, pondérées sur \mathbb{K} :

$$P \to E \in \mathbb{K}RatExpA \mid kP, k \in \mathbb{K} \mid Pk, k \in \mathbb{K} \mid P+P \mid PrP \mid (Pr)^*P.$$
 (3.24)

Proposition 16. L'image par ρ des expressions rationnelles sur l'alphabet A est l'ensemble des expressions pré-Hadamard sur A_r .

 $D\acute{e}monstration$. Soit E une expression de Hadamard. Il est immédiat par induction que $\rho(\mathsf{E})$ est une expression pré-Hadamard.

De plus, toute expression pré-Hadamard est image par ρ d'une expression de Hadamard. En effet, toutes les productions de la grammaire à la définition 3.3 correspondent à un membre droit des équations (3.9) à (3.14) qui définissent ρ .

On peut donc définir un inverse pour ρ sur son image, et l'on notera cet inverse ρ^{-1} . Cet inverse envoie donc E, une expression pré-Hadamard sur $A_{\mathbf{r}}$, vers une expression de Hadamard F telle que $\rho(\mathsf{F}) = \mathsf{E}$.

Aspects algorithmiques : des expressions de Hadamard aux automates circulaires pondérés...

Nous allons désormais présenter comment adapter les algorithmes de tranformation d'expressions rationnelles en automates unidirectionnels au problème de tranformation d'expressions de Hadamard en automates circulaires. On présente tout d'abord une extension générique, puis son application à des algorithmes classiques.

Extension générique de la synthèse d'automates

Il existe de nombreux algorithmes permettant de construire, à partir d'une expression rationnelle, un automate réalisant la même série, *i.e.* le comportement de l'automate et l'interprétation de l'expression désignent la même série.

À l'aide des fonctions introduites dans la section précédente, nous pouvons étendre un algorithme σ de transformation d'expression rationnelle en automate unidirectionnel afin qu'il transforme une expression de Hadamard en automate circulaire.

Théorème 3.1. Soit σ un algorithme (donc une application) convertissant une expression rationnelle en un automate unidirectionnel équivalent.

L'application $rot \circ \sigma \circ \rho$ envoie une expression de Hadamard sur un automate circulaire réalisant la même série.

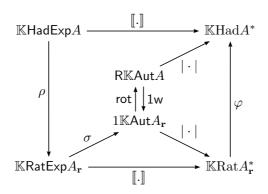


FIGURE 3.10 – Extension d'un algorithme σ .

Démonstration. Soit E une expression de Hadamard.

$$\begin{array}{lll} \mid (\sigma(\rho(\mathsf{E})))^{\mathbf{r}} \mid & = & \varphi(\mid \sigma(\rho(\mathsf{E})) \mid) & \quad \text{par le corollaire 2} \\ & = & \varphi(\llbracket \rho(\mathsf{E}) \rrbracket) & \quad \text{par d\'efinition de } \sigma \\ & = & \llbracket \mathsf{E} \rrbracket & \quad \text{par la proposition 15} \end{array}$$

Le graphe de la Figure 3.10 est donc commutatif.

La complexité de la fonction rot est constante, puisqu'il ne s'agit que d'une autre façon d'interpreter un même objet. La complexité de la transformation d'une expression de Hadamard en une expression rationnelle est linéaire en la taille de cette expression. Comme la complexité de la transformation σ d'une expression rationnelle en un automate est nécessairement au moins linéaire, la complexité de rot $\circ \sigma \circ \rho$ est donc celle de σ .

Cependant, lorsque la complexité de la conversion est calculée en fonction du nombre de lettres dans l'expression rationnelle, pour avoir la complexité de la conversion expression de Hadamard - automate circulaire, il faut ajouter au nombre de lettres de l'expression le nombre d'opérateurs de Hadamard utilisés.

Le théorème 3.1 est applicable à n'importe quel algorithme de conversion des expressions rationnelles en automates unidirectionnels. Nous allons, dans les sous-sections suivantes, appliquer ce théorème à trois algorithmes connus : la dérivation, les *follow*, et une méthode inspirée de Thompson.

Dérivation

La dérivation des expressions rationnelles pondérées a été introduite dans [Lombardy et Sakarovitch, 2005], et est une extension aux séries sur \mathbb{K} de l'algorithme d'Antimirov [Antimirov, 1996] sur les langages.

Il faut tout d'abord définir une fonction auxiliaire, introduite dans la proposition suivante :

Proposition 17. Soit E une expression rationnelle. Le poids du mot vide dans [E], $\langle E, \varepsilon \rangle$, peut être calculé par la fonction récursive suivante, Null, qui va des expressions rationnelles sur \mathbb{K} dans \mathbb{K} :

$$\begin{aligned} & \operatorname{Null}(0) = 0_{\mathbb{K}}, \\ & \operatorname{Null}(1) = 1_{\mathbb{K}}, \\ & \forall a \in A, \operatorname{Null}(a) = 0_{\mathbb{K}}, \\ & \forall k \in \mathbb{K}, \operatorname{Null}(k\mathsf{F}) = k \operatorname{Null}(\mathsf{F}), \\ & \operatorname{Null}(\mathsf{F}k) = \operatorname{Null}(\mathsf{F}) k, \\ & \operatorname{Null}(\mathsf{F} + \mathsf{G}) = \operatorname{Null}(\mathsf{F}) k + \operatorname{Null}(\mathsf{G}) \\ & \operatorname{Null}(\mathsf{F}.\mathsf{G}) = \operatorname{Null}(\mathsf{F}) \operatorname{Null}(\mathsf{G}), \\ & \operatorname{Null}(\mathsf{F}^*) = \operatorname{Null}(\mathsf{F})^*. \end{aligned}$$

La preuve est immédiate.

La dérivée d'une expression rationnelle est une combinaison linéaire d'expressions, que l'on nommera un polynôme d'expressions. La somme formelle dans ces polynômes sera notée \boxplus , afin d'éviter la confusion avec la somme des expressions rationnelles. De plus, on utilisera des crochets pour factoriser les opérations qui se distribuent sur les polynômes. Ainsi, on notera $[P \boxplus P'].Q$ le polynôme $P.Q \boxplus P'.Q$.

Définition 3.4. La *dérivée* d'une expression \mathbb{K} -rationnelle E sur A par une lettre a est un polynôme d'expressions noté $\frac{\partial}{\partial a}\mathsf{E}$. Il est défini inductivement comme :

$$\frac{\partial}{\partial a}0 = 0, \qquad \qquad \frac{\partial}{\partial a}1 = 0,$$

$$\frac{\partial}{\partial a}(\mathsf{E} + \mathsf{F}) = \frac{\partial}{\partial a}\mathsf{E} \boxplus \frac{\partial}{\partial a}\mathsf{F}, \qquad \forall b \in A, \ \frac{\partial}{\partial a}b = \begin{cases} 1 \text{ si } a = b, \\ 0 \text{ sinon}, \end{cases}$$

$$\forall k \in \mathbb{K}, \ \frac{\partial}{\partial a}k\mathsf{E} = k\frac{\partial}{\partial a}\mathsf{E}, \qquad \forall k \in \mathbb{K}, \frac{\partial}{\partial a}\mathsf{E}k = \left[\frac{\partial}{\partial a}\mathsf{E}\right]k, \qquad (3.26)$$

$$\frac{\partial}{\partial a}(\mathsf{EF}) = \left[\frac{\partial}{\partial a}\mathsf{E}\right].\mathsf{F} \boxplus \mathsf{Null}(\mathsf{E})\frac{\partial}{\partial a}\mathsf{F},$$

$$\frac{\partial}{\partial a}(\mathsf{E}^*) = \mathsf{Null}(\mathsf{E}^*)\left[\frac{\partial}{\partial a}\mathsf{E}\right].\mathsf{E}^*.$$

L'idée de la construction de l'automate des dérivées est d'associer à chacune de ces expressions générées un état de l'automate, avec l'état initial celui associé à l'expression originale. Entre les états F et G, il y a une transition étiquetée par a de poids k si $\langle \frac{\partial}{\partial a} \mathsf{F}, \mathsf{G} \rangle = k$. Il a été montré dans [Lombardy et Sakarovitch, 2005] qu'il n'y a qu'un nombre fini d'expressions générées par l'application successive de la dérivation à une expression, et que l'automate est donc fini. De plus, le comportement de cet automate est l'interprétation de l'expression d'origine.

On peut appliquer le théorème 3.1 à l'algorithme de dérivation. Appliquonsle tout d'abord sur un exemple.

Exemple 7

On cherche un automate circulaire qui réalise la même série que $E_0 = ((\frac{1}{2}(a +$ $(ab)^*b(a+b)^*$) $(ab)^*$, une expression de Hadamard.

Calculons tout d'abord $\rho(\mathsf{E}_0)$. En posant $\mathsf{F}_1 = (\frac{1}{2}(a+b))^*b(a+b)^*$, on a $E_1 = \rho(E_0) = (F_1 \mathbf{r})^* (ab)^*$.

On applique maintenant l'algorithme de construction d'automate à E_1 , une expression rationnelle.

Les dérivées de E_1 sont calculées dans le tableau 3.1.

Expression	$\frac{\partial}{\partial a}$	$\frac{\partial}{\partial b}$	$\frac{\partial}{\partial \mathbf{r}}$	
$E_1 = (F_1\mathbf{r})^*(ab)^*$				
$E_2 = F_1 \mathbf{r} E_1$	$\frac{1}{2}E_2$	$\frac{1}{2}E_2 \boxplus E_3$	0	(3.27)
	E_3	E_3	E_1	
$E_4 = b(ab)^*$	0	E_5	0	
$E_5 = (ab)^*$	E_4	0	0	

Table 3.1 – Les expressions dérivées de E₁.

Cela nous permet de construire l'automate des dérivées de E₁, en Figure 3.11. Cet automate, interprété comme un automate circulaire, réalise une série de Hadamard, qui est l'interprétation de E₀. On peut donc le voir comme l'automate des dérivées de E_0 .

Pour étendre le concept de dérivation aux expressions de Hadamard, on étend tout d'abord la définition de la fonction Null.

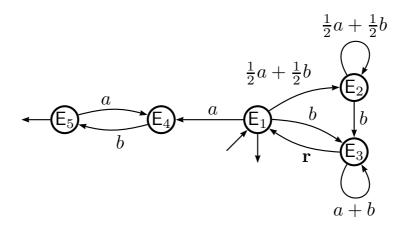


FIGURE 3.11 – L'automate des dérivées de $\mathsf{E}_1 = \rho(\mathsf{E}_0)$.

Définition 3.5. Soit E une expression de Hadamard. Si E est une expression rationnelle, alors Null(E) est définie à la définition 3.4. Sinon, on la définit inductivement comme suit :

$$\begin{split} \forall k \in \mathbb{K}, \quad & \mathsf{Null}(k\mathsf{E}) &= k \, \mathsf{Null}(\mathsf{E}), \\ & \quad & \mathsf{Null}(\mathsf{E}k) &= \mathsf{Null}(\mathsf{E}) \, k, \\ & \quad & \mathsf{Null}(\mathsf{E} \odot \mathsf{F}) &= 0, \\ & \quad & \mathsf{Null}(\mathsf{E} \circledast \mathsf{F}) &= \mathsf{Null}(\mathsf{F}), \\ & \quad & \quad & \mathsf{Null}(\mathsf{E} + \mathsf{F}) &= \mathsf{Null}(\mathsf{E}) + \mathsf{Null}(\mathsf{F}). \end{split}$$

Remarque 19. Cette extension de la fonction Null ne respecte pas la propriété Null(E) = $\langle E, \varepsilon \rangle$. Cependant, on a Null(E) = Null($\rho(E)$) = $\langle \rho(E), \varepsilon \rangle$, qui est l'égalité dont on aura besoin.

On peut désormais étendre la dérivation aux expressions de Hadamard.

Définition 3.6. Pour toute lettre a de A, la dérivée d'une expression de Hadamard est inductivement définie comme suit : si E est une expression rationnelle,

alors la dérivée est définie à la définition 3.4. Sinon :

$$\forall k \in \mathbb{K}, \ \frac{\partial}{\partial a} k \mathsf{E} = k \frac{\partial}{\partial a} \mathsf{E},$$

$$\frac{\partial}{\partial a} \mathsf{E} k = \left[\frac{\partial}{\partial a} \mathsf{E} \right] k,$$

$$\frac{\partial}{\partial a} (\mathsf{E} + \mathsf{F}) = \frac{\partial}{\partial a} \mathsf{E} \boxplus \frac{\partial}{\partial a} \mathsf{F},$$

$$\frac{\partial}{\partial a} (\mathsf{E} \odot \mathsf{F}) = \left[\frac{\partial}{\partial a} \mathsf{E} \right] \odot \mathsf{F},$$

$$\frac{\partial}{\partial a} (\mathsf{E} \circledast \mathsf{F}) = \left[\frac{\partial}{\partial a} \mathsf{E} \right] \odot (\mathsf{E} \circledast \mathsf{F}) \boxplus \frac{\partial}{\partial a} \mathsf{F}.$$

$$(3.29)$$

De plus, il nous faut définir un nouvel opérateur de dérivation pour le produit de Hadamard.

Si E est une expression rationnelle, alors $\frac{\partial}{\partial \odot} E = 0$. Sinon,

$$\frac{\partial}{\partial \odot}(k\mathsf{E}) = k \frac{\partial}{\partial \odot} \mathsf{E},$$

$$\frac{\partial}{\partial \odot}(\mathsf{E}k) = \left[\frac{\partial}{\partial \odot} \mathsf{E}\right] k,$$

$$\frac{\partial}{\partial \odot}(\mathsf{E} + \mathsf{F}) = \frac{\partial}{\partial \odot} \mathsf{E} \boxplus \frac{\partial}{\partial \odot} \mathsf{F},$$

$$\frac{\partial}{\partial \odot}(\mathsf{E} \odot \mathsf{F}) = \mathsf{Null}(\mathsf{E}) \mathsf{F} \boxplus \left[\frac{\partial}{\partial \odot} \mathsf{E}\right] \odot \mathsf{F},$$

$$\frac{\partial}{\partial \odot}(\mathsf{E} \circledast \mathsf{F}) = \left[\frac{\partial}{\partial \odot} \mathsf{E}\right] \odot (\mathsf{E} \circledast \mathsf{F}) \boxplus \mathsf{Null}(\mathsf{E}) (\mathsf{E} \circledast \mathsf{F}) \boxplus \frac{\partial}{\partial \odot} \mathsf{F}.$$
(3.30)

Ce résultat vient du théorème 3.1 et de la proposition suivante.

Proposition 18. Soit E une expression de Hadamard. Alors, en étendant naturellement ρ^{-1} aux polynômes d'expressions, on a les égalités suivantes :

$$\mathsf{Null}(\mathsf{E}) = \mathsf{Null}(\rho(\mathsf{E})), \tag{3.31}$$

$$\forall a \in A, \ \frac{\partial}{\partial a} \mathsf{E} = \rho^{-1} \left(\frac{\partial}{\partial a} \rho(\mathsf{E}) \right),$$
 (3.32)

$$\frac{\partial}{\partial \odot} \mathsf{E} = \rho^{-1} \left(\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{E}) \right). \tag{3.33}$$

Démonstration. Cette proposition se prouve par induction sur E.

Si E est une expression rationnelle, alors $\rho(\mathsf{E}) = \rho^{-1}(\mathsf{E}) = \mathsf{E}$, et on a bien $\frac{\partial}{\partial \mathbf{r}}\mathsf{E} = \frac{\partial}{\partial \odot}\mathsf{E} = 0$. Sinon, soient deux expressions de Hadamard F et G vérifiant la proposition :

• $\mathsf{E} = \mathsf{F} \odot \mathsf{G}$. On a $\mathsf{Null}(\rho(\mathsf{E})) = \mathsf{Null}(\rho(\mathsf{F})) \mathsf{Null}(\mathbf{r}) \mathsf{Null}(\rho(\mathsf{G})) = 0 = \mathsf{Null}(\mathsf{E})$, et :

$$\frac{\partial}{\partial a}(\mathsf{F} \odot \mathsf{G}) = \left[\frac{\partial}{\partial a}\mathsf{F}\right] \odot \mathsf{G} = \left[\rho^{-1}\left(\frac{\partial}{\partial a}\rho(\mathsf{F})\right)\right] \odot \mathsf{G}$$

$$= \rho^{-1}\left(\left[\frac{\partial}{\partial a}\rho(\mathsf{F})\right]\mathbf{r}\rho(\mathsf{G})\right)$$

$$= \rho^{-1}\left(\frac{\partial}{\partial a}\left(\rho(\mathsf{F})\mathbf{r}\rho(\mathsf{G})\right)\right)$$

$$= \rho^{-1}\left(\frac{\partial}{\partial a}\rho(\mathsf{E})\right).$$
(3.34)

On a également pour la dérivée par rapport au produit de Hadamard :

$$\frac{\partial}{\partial \odot}(\mathsf{F} \odot \mathsf{G}) = \mathsf{Null}(\mathsf{F}) \, \mathsf{G} \, \boxplus \left[\frac{\partial}{\partial \odot} \mathsf{F} \right] \odot \mathsf{G}$$

$$= \mathsf{Null}(\rho(\mathsf{F})) \, \mathsf{G} \, \boxplus \left[\rho^{-1} \left(\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{F}) \right) \right] \odot \mathsf{G}$$

$$= \rho^{-1} \left(\mathsf{Null}(\rho(\mathsf{F})) \rho(\mathsf{G}) \, \boxplus \left[\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{F}) \right] \mathbf{r} \rho(\mathsf{G}) \right)$$

$$= \rho^{-1} \left(\left[\left[\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{F}) \right] \mathbf{r} \, \boxplus \, \mathsf{Null}(\rho(\mathsf{F})) \frac{\partial}{\partial \mathbf{r}} \mathbf{r} \right] \rho(\mathsf{G}) \right)$$

$$= \rho^{-1} \left(\left[\frac{\partial}{\partial \mathbf{r}} (\rho(\mathsf{F}) \mathbf{r}) \right] \rho(\mathsf{G}) \right)$$

$$= \rho^{-1} \left(\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{E}) \right).$$

 $\bullet \ \mathsf{E} = \mathsf{F} \circledast \mathsf{G}. \ \mathrm{On} \ \mathrm{a} \ \mathsf{Null}(\rho(\mathsf{E})) = (\mathsf{Null}(\rho(\mathsf{F})) \mathsf{Null}(\mathbf{r}))^* \mathsf{Null}(\mathsf{G}) = \mathsf{Null}(\mathsf{G}) =$

Null(E), et:

$$\frac{\partial}{\partial a}(\mathsf{F} \circledast \mathsf{G}) = \left[\frac{\partial}{\partial a}\mathsf{F}\right] \odot (\mathsf{F} \circledast \mathsf{G}) \boxplus \frac{\partial}{\partial a}\mathsf{G}$$

$$= \rho^{-1} \left(\left[\frac{\partial}{\partial a}\rho(\mathsf{F})\right] \mathbf{r}(\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G})\right) \boxplus \frac{\partial}{\partial a}\mathsf{G}$$

$$= \rho^{-1} \left((\mathsf{Null}(\rho(\mathsf{F})\mathbf{r}))^* \left[\frac{\partial}{\partial a}(\rho(\mathsf{F})\mathbf{r})\right] (\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G})\right) \boxplus \frac{\partial}{\partial a}\mathsf{G}$$

$$= \rho^{-1} \left(\left[\frac{\partial}{\partial a}(\rho(\mathsf{F})\mathbf{r})^*\right] \rho(\mathsf{G}) \boxplus \mathsf{Null}((\rho(\mathsf{F})\mathbf{r})^*) \frac{\partial}{\partial a}\rho(\mathsf{G})\right)$$

$$= \rho^{-1} \left(\frac{\partial}{\partial a}((\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G}))\right)$$

$$= \rho^{-1} \left(\frac{\partial}{\partial a}\rho(\mathsf{E})\right).$$
(3.36)

Quant à la dérivée par rapport au produit de Hadamard, elle vaut :

$$\begin{split} \frac{\partial}{\partial \odot}(\mathsf{F} \circledast \mathsf{G}) &= \left[\frac{\partial}{\partial \odot}\mathsf{F}\right] \odot \mathsf{E} \boxplus \mathsf{Null}(\mathsf{F}) \mathsf{E} \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\ &= \rho^{-1} \left(\left[\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{F})\right] \mathbf{r} \rho(\mathsf{E}) \right) \boxplus \rho^{-1} \left(\mathsf{Null}(\rho(\mathsf{F})) \frac{\partial}{\partial \mathbf{r}} \mathbf{r} \rho(\mathsf{E}) \right) \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\ &= \rho^{-1} \left(\left[\left[\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{F})\right] \mathbf{r} \boxplus \mathsf{Null}(\rho(\mathsf{F})) \frac{\partial}{\partial \mathbf{r}} \mathbf{r} \right] \rho(\mathsf{E}) \right) \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\ &= \rho^{-1} \left(\left[\frac{\partial}{\partial \mathbf{r}} (\rho(\mathsf{F}) \mathbf{r})\right] \rho(\mathsf{E}) \right) \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\ &= \rho^{-1} \left(\left(\mathsf{Null}(\rho(\mathsf{F}) \mathbf{r}) \right)^* \left[\frac{\partial}{\partial \mathbf{r}} (\rho(\mathsf{F}) \mathbf{r})\right] (\rho(\mathsf{F}) \mathbf{r})^* \rho(\mathsf{G}) \boxplus \frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{G}) \right) \\ &= \rho^{-1} \left(\left[\frac{\partial}{\partial \mathbf{r}} (\rho(\mathsf{F}) \mathbf{r})^*\right] \rho(\mathsf{G}) \boxplus \mathsf{Null} \left((\rho(\mathsf{F}) \mathbf{r})^* \right) \frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{G}) \right) \\ &= \rho^{-1} \left(\frac{\partial}{\partial \mathbf{r}} (\rho(\mathsf{F}) \mathbf{r})^* \rho(\mathsf{G}) \right) \\ &= \rho^{-1} \left(\frac{\partial}{\partial \mathbf{r}} \rho(\mathsf{E}) \right). \end{split} \tag{3.37}$$

Ainsi, on a pu étendre aux expressions de Hadamard l'algorithme des dérivées, afin d'obtenir un automate circulaire équivalent.

Automate des follow

La définition pour les expressions rationnelles et construisant l'automate des *follow* se trouve dans [Ilie et Yu, 2003].

Il est appelé ainsi parce qu'il repose sur le calcul d'une fonction qui, étant donnée une expression et une lettre de cette expression, calcule les lettres qui peuvent la suivre — follow en anglais — dans l'expression. À chaque lettre, ou plutôt à chaque position de lettre dans l'expression, correspondra un état, et la fonction follow donnera les transitions sortantes.

Plus précisément, on considère, pour une expression rationnelle E , la liste des occurences des lettres. Chacune de ces occurences est appellée *position*, et l'ensemble de ces positions est noté $\mathsf{pos}(\mathsf{E})$.

Pour définir l'automate des follow, nous aurons besoin de quatre fonctions : Null, des expressions vers \mathbb{K} , définie plus haut(cf. définition 3.6), First, Follow, et Last, des expressions dans les combinaisons linéaires — formelles — de positions. La fonction Follow a de plus une position en argument. On notera $\langle L, p \rangle$ le coefficient d'une position p dans une combinaison linéaire de positions L.

Ces fonctions sont définies comme suit, avec a une lettre de A:

First(a) = position de
$$a$$
,
First(0) = First(1) = $0_{\mathbb{K}}$,
First(F + G) = First(F) \boxplus First(G),
First(FG) = First(F) \boxplus Null(F)First(G), (3.38)
 $\forall k \in \mathbb{K}$, First(k F) = k First(F),
First(F k) = First(F),
First(F*) = (Null(F))*First(F),
Last(0) = Last(1) = $0_{\mathbb{K}}$,
Last(a) = position de a ,
Last(F + G) = Last(F) \boxplus Last(G),
Last(FG) = Last(G) \boxplus Last(F)Null(G), (3.39)
 $\forall k \in \mathbb{K}$, Last(k F) = Last(F),
Last(F k) = Last(F),
Last(F*) = Last(F)(Null(F))*,

$$\begin{aligned} \operatorname{Follow}(0,p) &= \operatorname{Follow}(1,p) &= \operatorname{Follow}(a,p) = 0_{\mathbb{K}}, \\ \forall k \in \mathbb{K}, \ \operatorname{Follow}(k\mathsf{F},p) &= \operatorname{Follow}(\mathsf{F}k,p) = \operatorname{Follow}(\mathsf{F},p), \\ \operatorname{Follow}(\mathsf{F}+\mathsf{G},p) &= \begin{cases} \operatorname{Follow}(\mathsf{F},p) & \text{si } p \in \operatorname{pos}(\mathsf{F}), \\ \operatorname{Follow}(\mathsf{G},p) & \text{si } p \in \operatorname{pos}(\mathsf{G}), \end{cases} \\ \operatorname{Follow}(\mathsf{FG},p) &= \begin{cases} \operatorname{Follow}(\mathsf{F},p) \boxplus \langle \operatorname{Last}(\mathsf{F}),p \rangle \operatorname{First}(\mathsf{G}), \ \text{si } p \in \operatorname{pos}(\mathsf{F}), \\ \operatorname{Follow}(\mathsf{G},p) & \text{si } p \in \operatorname{pos}(\mathsf{G}), \end{cases} \\ \operatorname{Follow}(\mathsf{F}^*,p) &= \operatorname{Follow}(\mathsf{F},p) \boxplus \langle \operatorname{Last}(\mathsf{F}),p \rangle (\operatorname{Null}(\mathsf{F}))^* \operatorname{First}(\mathsf{F}). \end{aligned} \tag{3.40}$$

On rajoute de plus une position i_0 , la position initiale, à pos(E), telle que $Follow(E, i_0) = First(E)$ et $\langle Last(E), i_0 \rangle = Null(E)$.

On définit maintenant l'automate des positions, qui nous servira à définir l'automate des follow:

Définition 3.7. On appelle automate des positions de l'expression rationnelle E l'automate (Q, E, I, T) suivant :

- Q est l'ensemble des positions de E,
- $E(p, a, q) = \langle \mathsf{Follow}(\mathsf{E}, p), q \rangle$ si la lettre à la position q est un a,
- I est la fonction caractéristique de i_0 ,
- $T(p) = \mathsf{Last}(p)$.

On peut réduire la taille de cet automate en en prenant un quotient : c'est la définition suivante.

Définition 3.8. L'automate des follow de l'expression rationnelle E est un quotient de l'automate des positions de E : si Follow(E, p) = Follow(E, q) et $\langle \mathsf{Last}(\mathsf{E}), p \rangle = \langle \mathsf{Last}(\mathsf{E}), q \rangle$, alors on fusionne les états p et q.

L'automate des positions est ainsi un revêtement de l'automate des follow.

Exemple 8

On considère l'expression E_1 définie dans l'exemple 7. C'est une expression rationnelle, avec 8 positions.

Par souci de simplicité, on ajoute des indices aux lettres, pour repérer les positions : $\mathsf{E}_1 = ((\frac{1}{2}(a_1+b_2))^*b_3(a_4+b_5)^*\mathbf{r}_6)^*(a_7b_8)^*$

Les fonctions Follow et Last calculées dans le tableau 3.2 induisent une relation d'équivalence sur les positions : $\{\{i_0, 6\}, \{1, 2\}, \{3, 4, 5\}, \{7\}, \{8\}\}\}$; on peut donc construire l'automate des follow de E_1 , en Figure 3.12.

Cet automate, interprété comme un automate circulaire, réalise $\llbracket \mathsf{E}_0 \rrbracket$.

position	$Follow(E_1, \mathrm{pos})$	$\langle Last(E_1), \mathrm{pos} \rangle$	
i_0	$\frac{1}{2} 1 \boxplus \frac{1}{2} 2 \boxplus 3 \boxplus 7$	1	
1 (a)	$\frac{1}{2} 1 \boxplus \frac{1}{2} 2 \boxplus 3$	0	
2 (b)	$\frac{1}{2} 1 \boxplus \frac{1}{2} 2 \boxplus 3$	0	
3 (b)	$4 \boxplus 5 \boxplus 6$	0	(3.41)
4 (a)	$4 \boxplus 5 \boxplus 6$	0	(0.41)
5 (b)	$4 \boxplus 5 \boxplus 6$	0	
6 (r)	$\frac{1}{2} 1 \boxplus \frac{1}{2} 2 \boxplus 3 \boxplus 7$	1	
7 (a)	8	0	
8 (b)	7	1	

Table 3.2 – Les fonctions Follow et Last de l'expression E_1 .

Pour étendre le concept de l'automate des *follow* aux automates circulaires, il faut étendre la définition des fonctions qu'il utilise.

Définition 3.9. On étend la définition des équations (3.38) à (3.40) aux expressions de Hadamard, où l'on considère les positions non plus uniquement des lettres, mais des lettres et des opérateurs de Hadamard. On notera \odot_i pour indiquer que l'opérateur est en position i.

$$\begin{aligned} \operatorname{First}(\mathsf{F} \odot_i \mathsf{G}) &= \operatorname{First}(\mathsf{F}) \boxplus \operatorname{Null}(\mathsf{F}) \, i, \\ \operatorname{First}(\mathsf{F} \circledast_i \mathsf{G}) &= \operatorname{First}(\mathsf{F}) \boxplus \operatorname{Null}(\mathsf{F}) \, i \boxplus \operatorname{First}(\mathsf{G}), \end{aligned} \tag{3.42}$$

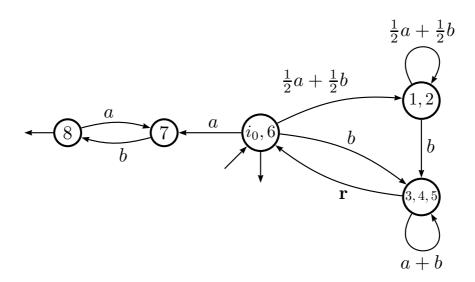


FIGURE 3.12 – L'automate des follow de $\mathsf{E}_1 = \rho(\mathsf{E}_0)$.

Exemple 9

Les positions de E_0 sont indiquées dans le tableau (3.45) suivant :

$$E_0 = ((\frac{1}{2}(a+b))^* b (a+b)^*) \otimes (ab)^*$$
position 1 2 3 4 5 6 78

Comme à chaque utilisation d'un opérateur de Hadamard dans E correspond un \mathbf{r} dans $\rho(\mathsf{E})$, il y a une bijection naturelle entre les positions de E et celles de $\rho(\mathsf{E})$.

La proposition suivante montre la correction de la définition 3.9 :

Proposition 19. Pour toute expression de Hadamard E on a :

$$First(E) = First(\rho(E)),$$
 (3.46)

$$\mathsf{Last}(\mathsf{E}) = \mathsf{Last}(\rho(\mathsf{E})),\tag{3.47}$$

$$\forall p \in \mathsf{pos}(\mathsf{E}), \; \mathsf{Follow}(\mathsf{E}, p) = \mathsf{Follow}(\rho(\mathsf{E}), p).$$
 (3.48)

Démonstration. La preuve est par induction sur E. Nous montrons l'equation sur Follow, la preuve pour First et Last s'effectuant de la même manière.

Il est immédiat qu'elle est vérifiée pour E si c'est une expression rationnelle, ou pour E de la forme $k\mathsf{F}$, $\mathsf{F}k$ ou $\mathsf{F}+\mathsf{G}$, avec k dans \mathbb{K} , et F et G des expressions vérifiant la propriété.

Supposons maintenant que $E = F \odot_i G$.

On a, pour tout p dans pos(F),

$$\begin{split} \mathsf{Follow}(\rho(\mathsf{E}),p) &= \mathsf{Follow}(\rho(\mathsf{F})\mathbf{r}_i\rho(\mathsf{G}),p), \\ &= \mathsf{Follow}(\rho(\mathsf{F}),p) \boxplus \langle \mathsf{Last}(\rho(\mathsf{F})),p\rangle \mathsf{First}(\mathbf{r}_i\rho(\mathsf{G})), \\ &= \mathsf{Follow}(\mathsf{F},p) \boxplus \langle \mathsf{Last}(\mathsf{F}),p\rangle i, \\ &= \mathsf{Follow}(\mathsf{F}\odot_i\mathsf{G},p). \end{split}$$

On a également

$$\begin{split} \mathsf{Follow}(\rho(\mathsf{E}),i) &= \mathsf{Follow}(\rho(\mathsf{F})\mathbf{r}_i\rho(\mathsf{G}),i), \\ &= \langle \mathsf{Last}(\mathbf{r}_i),i\rangle \mathsf{First}(\rho(\mathsf{G})), \\ &= \mathsf{First}(\mathsf{G}), \\ &= \mathsf{Follow}(\mathsf{F}\odot_i\mathsf{G},i). \end{split}$$

Et pour tout p dans pos(G),

$$\begin{split} \mathsf{Follow}(\rho(\mathsf{E}),p) &= \mathsf{Follow}(\rho(\mathsf{G}),p), \\ &= \mathsf{Follow}(\mathsf{G},p), \\ &= \mathsf{Follow}(\mathsf{F}\odot_i\mathsf{G},p). \end{split}$$

Ainsi, pour tout $p \in pos(E)$, on a $Follow(\rho(E), p) = Follow(E, p)$. On procède de la même manière avec $E = F \circledast_i G$.

Pour tout p dans pos(F):

$$\begin{split} \mathsf{Follow}(\rho(\mathsf{E}),p) &= \mathsf{Follow}((\rho(\mathsf{F})\mathbf{r}_i)^*\rho(\mathsf{G}),p), \\ &= \mathsf{Follow}((\rho(\mathsf{F})\mathbf{r}_i)^*,p) \boxplus \langle \mathsf{Last}((\rho(\mathsf{F})\mathbf{r}_i)^*),p\rangle \mathsf{First}(\rho(\mathsf{G})), \\ &= \mathsf{Follow}(\rho(\mathsf{F})\mathbf{r}_i,p) \\ &= \mathsf{Follow}(\rho(\mathsf{F}),p) \boxplus \langle \mathsf{Last}(\rho(\mathsf{F})),p\rangle i, \\ &= \mathsf{Follow}(\mathsf{F},p) \boxplus \langle \mathsf{Last}(\mathsf{F}),p\rangle i, \\ &= \mathsf{Follow}(\mathsf{F}\circledast_i\mathsf{G},p). \end{split}$$

De même, avec p = i, on a :

$$\begin{split} \mathsf{Follow}(\rho(\mathsf{E}),i) &= \mathsf{Follow}((\rho(\mathsf{F})\mathbf{r}_i)^*\rho(\mathsf{G}),i), \\ &= \mathsf{Follow}((\rho(\mathsf{F})\mathbf{r}_i)^*,i) \boxplus \langle \mathsf{Last}((\rho(\mathsf{F})\mathbf{r}_i)^*),i\rangle \mathsf{First}(\rho(\mathsf{G})), \\ &= \mathsf{Follow}(\rho(\mathsf{F})\mathbf{r}_i,i) \boxplus \langle \mathsf{Last}(\rho(\mathsf{F})\mathbf{r}_i),i\rangle \mathsf{First}(\rho(\mathsf{F})\mathbf{r}_i) \\ &\boxplus \langle \mathsf{Last}(\rho(\mathsf{F})\mathbf{r}_i),i\rangle \mathsf{First}(\mathsf{G}), \\ &= \langle \mathsf{Last}(\mathbf{r}_i),i\rangle \left(\mathsf{First}(\rho(\mathsf{F})) \boxplus \mathsf{Null}(\rho(\mathsf{F}))\mathsf{First}(\mathbf{r}_i)\right) \boxplus \mathsf{First}(\mathsf{G}), \\ &= \mathsf{First}(\mathsf{F}) \boxplus \mathsf{Null}(\mathsf{F})i \boxplus \mathsf{First}(\mathsf{G}), \end{split}$$

$$= \mathsf{First}(\mathsf{F} \circledast_i \mathsf{G}),$$
$$= \mathsf{Follow}(\mathsf{F} \circledast_i \mathsf{G}, p).$$

Et pour tout p dans pos(G),

$$\begin{split} \mathsf{Follow}(\rho(\mathsf{E}),p) &= \mathsf{Follow}((\rho(\mathsf{F})\mathbf{r}_i)^*\rho(\mathsf{G}),p), \\ &= \mathsf{Follow}(\rho(\mathsf{G}),p) \boxplus \langle \mathsf{Last}((\rho(\mathsf{F})\mathbf{r}_i)^*),p\rangle \mathsf{First}(\rho(\mathsf{G})), \\ &= \mathsf{Follow}(\mathsf{G},p) \\ &= \mathsf{Follow}(\mathsf{F}\circledast_i\mathsf{G},p) \end{split}$$

Ainsi,
$$Follow(\rho(E), p) = Follow(E, p)$$
.

Automate Thompson-like

Les définitions d'automates permettent parfois d'étiqueter une transition avec ε . Un algorithme de conversion utilise de manière extensive de telles transitions : il s'agit de la construction de Thompson [Thompson, 1968]. On étudie ici une variante de cet algorithme, inspirée par la ZPC-structure de [Ziadi et al., 1997], basée sur son extension aux multiplicités dans [Champarnaud et al., 2003].

Informellement, pour une expression rationnelle E, et similairement à l'automate de Thompson classique, l'automate a un unique état initial de poids 1, et un état final t de poids 1, distinct de i. De plus, i n'a pas de transition rentrante, t n'a pas de transition sortante, et si un état possède deux transitions entrantes ou deux transitions sortantes distinctes, ces transitions sont étiquetées par ε .

Cependant, contrairement à la construction classique, i peut être final, et il n'y a pas d' ε -transition de i à t: le poids final de i est donc le poids de ε dans E. L'état i est le seul état différent de t qui peut être final. Cette construction garantit l'absence de circuit d' ε -transitions, contrairement à la construction usuelle.

On pose dans un souci de concision $\chi_{\rm E}$ la fonction caractéristique de l'ensemble E, c'est-à-dire la fonction qui à x associe 1 si x est dans E, et 0 sinon. La définition formelle est récursive :

Définition 3.10. L'automate Thompson-like (Q, E, I, T) de l'expression rationnelle F, noté $\mathcal{T}(\mathsf{F})$, est défini récursivement. On pose a une lettre, F_1 et F_2 deux expressions rationnelles, $\mathcal{T}(\mathsf{F}_1) = (Q_1, E_1, \chi_{i_1}, T_1)$, d'états finaux t_1 de poids 1, et i_1 de poids c_1 , $\mathcal{T}(\mathsf{F}_2) = (Q_2, E_2, \chi_{i_2}, T_2)$ d'états finaux t_2 de poids 1, et i_2 de poids c_2 . a Voici maintenant la définition récursive de cet automate. Pour tous les cas, on a $I = \chi_{\{i\}}$; ensuite :

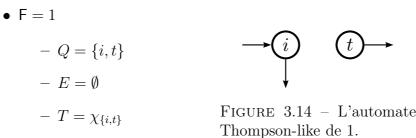
•
$$F = 0$$

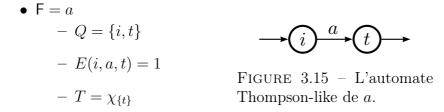
3.3. Des expressions de Hadamard aux automates circulaires pondérés...

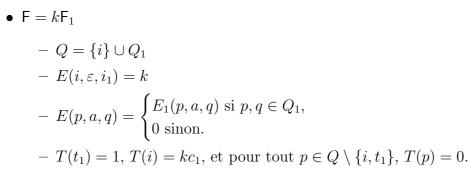
$$-Q = \{i, t\}$$

$$-E = \emptyset$$
FIGURE 2.12

FIGURE 3.13 – L'automate $T = \chi_{\{t\}}$ Thompson-like de 0.







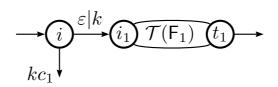


FIGURE 3.16 – L'automate Thompson-like de kF_1 .

•
$$\mathsf{F} = \mathsf{F}_1 k$$

Ici, on a $i_1 = i$.
- $Q = \{t\} \cup Q_1$
- $E(t_1, \varepsilon, t) = k$
- $E(p, a, q) = \begin{cases} E_1(p, a, q) & \text{si } p, q \in Q_1 \\ 0 & \text{sinon} \end{cases}$

$$-T(i_1) = c_1 k, T(t) = 1$$
, et pour tout $p \in Q \setminus \{i_1, t\}, T(p) = 0$.

$$\xrightarrow[c_1 k]{c_1} \underbrace{\mathcal{T}(\mathsf{F}_1)} \underbrace{t_1}^{\varepsilon|k} \underbrace{t}$$

FIGURE 3.17 – L'automate Thompson-like de F_1k .

•
$$\mathsf{F} = \mathsf{F}_1 \mathsf{F}_2$$

- $Q = \{i, t\} \cup Q_1 \cup Q_2$
- $E(i, \varepsilon, i_2) = c_1$
- $E(t_1, \varepsilon, t) = c_2$
- $E(i, \varepsilon, i_1) = E(t_1, \varepsilon, i_2) = E(t_2, \varepsilon, t) = 1$
- $E(p, a, q) = \begin{cases} E_1(p, a, q) & \text{si } p, q \in Q_1, \\ E_2(p, a, q) & \text{si } p, q \in Q_2, \\ 0 & \text{sinon} \end{cases}$
- $T(i) = c_1 c_2, T(t) = 1$, et pour tout $p \in Q \setminus \{i, t\}, T(p) = 0$.

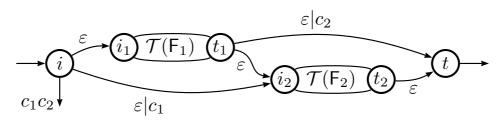


FIGURE 3.18 - L'automate Thompson-like de F_1F_2 .

•
$$\mathsf{F} = \mathsf{F}_1^*$$

- $Q = \{i, t\} \cup Q_1$
- $E(i, \varepsilon, i_1) = E(t_1, \varepsilon, i_1) = E(t_1, \varepsilon, t) = c_1^*$
- $E(p, a, q) = \begin{cases} E_1(p, a, q) & \text{si } p, q \in Q_1, \\ 0 & \text{sinon} \end{cases}$
- $T(i) = c_1^*, T(t) = 1$, et pour tout $p \in Q \setminus \{i, t\}, T(p) = 0$.

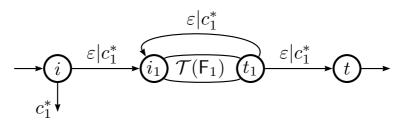


FIGURE 3.19 – L'automate Thompson-like de F_1^* .

En utilisant le théorème 3.1, on peut étendre cette définition aux expressions de Hadamard :

Définition 3.11. L'automate *Thompson-like* (Q, E, R, I, T) de l'expression de Hadamard F, noté $\mathcal{T}(\mathsf{F})$, est défini récursivement. On pose a une lettre, F_1 et F_2 deux expressions de Hadamard, $\mathcal{T}(\mathsf{F}_1) = (Q_1, E_1, \chi_{i_1}, T_1)$, d'états finaux t_1 de poids 1, et i_1 de poids c_1 , $\mathcal{T}(\mathsf{F}_2) = (Q_2, E_2, \chi_{i_2}, T_2)$ d'états finaux t_2 de poids 1, et i_2 de poids i_2 . Voici maintenant la définition récursive de cet automate. Pour tous les cas, on a $i_1 = i_2$ ensuite :

• F est une expression rationnelle : $\mathcal{T}(\mathsf{F}) \text{ est d\'efini \`a la d\'efinition } 3.10.$

•
$$\mathsf{F} = \mathsf{F}_1 \odot \mathsf{F}_2$$

- $Q = \{i, t, i_{\mathbf{r}}, t_{\mathbf{r}}\} \cup Q_1 \cup Q_2$

- $E(i, \varepsilon, i_1) = E(t_1, \varepsilon, i_{\mathbf{r}}) = E(t_{\mathbf{r}}, \varepsilon, i_2) = E(t_2, \varepsilon, t) = 1$

- $E(i, \varepsilon, i_{\mathbf{r}}) = c_1$

- $E(t_{\mathbf{r}}, \varepsilon, t) = c_2$

- $E(p, a, q) = \begin{cases} E_1(p, a, q) & \text{si } p, q \in Q_1, \\ E_2(p, a, q) & \text{si } p, q \in Q_2, \\ 0 & \text{sinon} \end{cases}$

- $R(i_{\mathbf{r}}, t_{\mathbf{r}}) = 1$

- $R(p, q) = \begin{cases} R_1(p, q) & \text{si } p, q \in Q_1, \\ R_2(p, q) & \text{si } p, q \in Q_2, \\ 0 & \text{sinon} \end{cases}$

- $T = \chi_{\{t\}}$
 $\varepsilon | c_1$

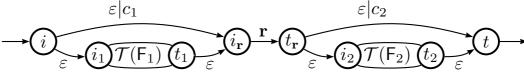


FIGURE 3.20 – L'automate Thompson-like de $F_1 \odot F_2$.

•
$$\mathsf{F} = \mathsf{F}1 \circledast \mathsf{F}_2$$

$$- Q = \{i, t, i_\mathbf{r}, t_\mathbf{r}, p_1, p_2\} \cup Q_1 \cup Q_2$$

$$- E(i, \varepsilon, p_1) = E(i, \varepsilon, i_2) = E(t_\mathbf{r}, \varepsilon, p_1) = E(t_\mathbf{r}, \varepsilon, p_2) = E(p_1, \varepsilon, i_1) = E(p_2, \varepsilon, i_2) = E(t_1, \varepsilon, i_\mathbf{r}) = E(t_2, \varepsilon, t) = 1$$

$$- E(p_1, \varepsilon, i_\mathbf{r}) = c_1$$

$$-E(p_{2}, \varepsilon, t) = c_{2}$$

$$-E(p, a, q) = \begin{cases} E_{1}(p, a, q) & \text{si } p, q \in Q_{1}, \\ E_{2}(p, a, q) & \text{si } p, q \in Q_{2}, \end{cases}$$

$$-R(i_{\mathbf{r}}, t_{\mathbf{r}}) = 1$$

$$-R(p, q) = \begin{cases} R_{1}(p, q) & \text{si } p, q \in Q_{1}, \\ R_{2}(p, q) & \text{si } p, q \in Q_{2}, \end{cases}$$

$$-T(t) = 1, T(i) = c_{2}, \text{ et pour tout } p \in Q \setminus \{i, t\}, T(p) = 0.$$

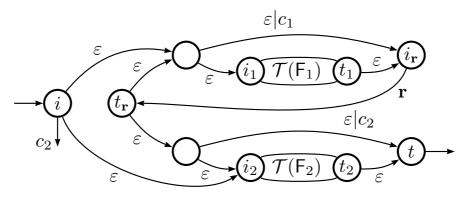


FIGURE 3.21 – L'automate Thompson-like de $F_1 \otimes F_2$.

Ces automates sont bien ceux que l'on aurait obtenus en appliquant la transformation aux expressions $\rho(\mathsf{F}_1 \odot \mathsf{F}_2)$ et $\rho(\mathsf{F}_1 \circledast \mathsf{F}_2)$. On peut donc transformer une expression de Hadamard en un automate circulaire. On cherche maintenant à effectuer la transformation dans l'autre sens.

...Et vice-versa

Pour effectuer la transformation inverse, à savoir transformer un automate circulaire en expression de Hadamard, on adopte la même philosophie : on considère l'automate comme un automate unidirectionnel sur un alphabet étendu, on applique un algorithme qui transforme les automates unidirectionnels en expressions rationnelles sur cet alphabet, et on transforme ces expressions rationnelles en expressions de Hadamard.

Pour cette dernière étape, on veut utiliser ρ^{-1} . Or l'image de ρ est un sousensemble des expressions rationnelles : il s'agit des expressions pré-Hadamard de la définition 3.3.

On souhaite donc s'assurer que l'algorithme utilisé ne produise que des expressions pré-Hadamard, et obtenir le théorème suivant :

Théorème 3.2. Soit τ un algorithme (donc une application) convertissant un automate unidirectionnel en une expression pré-Hadamard équivalente. L'application $\rho^{-1} \circ \tau \circ 1$ w envoie un automate circulaire sur une expression pré-Hadamard réalisant la même série.

Nous allons ici appliquer ce théorème avec l'algorithme d'élimination d'états, dont on va détailler le principe.

On convertit tout d'abord l'automate $\mathcal{A} = (Q, E, I, T)$ en un graphe simple orienté, dont on étiquette les arcs par des expressions rationnelles :

- Ses sommets sont les états de l'automate, plus deux nouveaux sommets, i_0 et t_0 .
- Pour chaque transition (p, a, q) de poids k non nul, l'arc (p, q) existe et est étiqueté par ka. S'il y a plusieurs transitions de p à q, l'étiquette de l'arc (p, q) est la somme des $k_i a_i$, avec la transition (p, a_i, q) de poids k_i non nul.
- Si l'état p est initial de poids k, l'arc (i_0, p) est étiqueté par k.
- Si l'état q est final de poids k, l'arc (q, t_0) est étiqueté par k.

Ensuite, l'idée est d'éliminer un par un chacun des sommets différents de i_0 et de t_0 , jusqu'à ce qu'il ne reste que ces deux sommets : tant qu'il reste un sommet p différent de i_0 et de t_0 , pour toute paire de sommets q, r tels que q est un prédécesseur de p et r un successeur, on met à jour l'étiquette de l'arc (q, r) d'après le processus décrit en Figure 3.22. On supprime ensuite le sommet p.



FIGURE 3.22 – Processus d'élimination de l'état p.

Lorsque l'on a éliminé tous les sommets sauf i_0 et t_0 , on retourne l'étiquette de l'arc (i_0, t_0) , qui est une expression rationnelle sur $A_{\mathbf{r}}$, dont l'interprétation est la même série que le comportement de \mathcal{A} . La forme de cette expression dépend fortement de l'ordre choisi pour éliminer les états. Mais une élimination quelconque ne nous permet pas toujours d'obtenir une expression pré-Hadamard.

Exemple 10

L'élimination d'états, sur l'automate de la Figure 3.23, donne soit $\varepsilon + b(a + (\frac{1}{2}\mathbf{r} + a)b)^*(\frac{1}{2}\mathbf{r} + a)$, soit $(ba^*(\frac{1}{2}\mathbf{r} + a))^*$. Aucune de ces expressions n'est une expression pré-Hadamard.

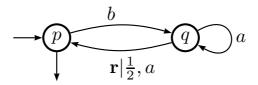


FIGURE 3.23 – Un automate unidirectionnel sur A_r .

Pour assurer l'obtention d'une expression pré-Hadamard, il nous faut travailler avec des automates spéciaux : les automates **r**-locaux. Après leur définition, nous montrons que tout automate unidirectionnel peut être transformé en un automate équivalent **r**-local, et enfin nous appliquons l'algorithme à un automate **r**-local.

Automate r-local

Définition 3.12. Un automate unidirectionnel sur l'alphabet $A_{\mathbf{r}}$ est dit r-local s'il existe une partition $\{Q_1, Q_2\}$ de l'ensemble des états Q telle qu'il n'y ait aucun état initial dans Q_2 , et que l'étiquette d'une transition soit \mathbf{r} si et seulement si la destination de cette transition est dans Q_2 .

Cette propriété ne restreint pas l'expressivité des automates. En effet, on a la proposition suivante :

Proposition 20. Tout automate unidirectionnel sur l'alphabet A_r admet un revêtement r-local.

Démonstration. Soit $\mathcal{A} = (S, F, J, U)$ un automate unidirectionnel sur l'alphabet $A_{\mathbf{r}}$. On pose $\mathcal{B} = (Q, E, I, T)$ un automate unidirectionnel sur l'alphabet $A_{\mathbf{r}}$ tel que $Q = Q_1 \cup Q_2$, avec Q_1 et Q_2 deux copies de S: la copie dans Q_1 de l'état p de S sera nommée p_1 , et p_2 celle dans Q_2 . De plus, pour tout p, q dans S et i dans $\{1, 2\}$, on pose:

$$I(p_1) = J(p), I(p_2) = 0, T(p_i) = U(p),$$

 $\forall a \in A, E(p_i, a, q_1) = F(p, a, q),$
 $\forall a \in A, E(p_i, a, q_2) = 0,$
 $E(p_i, \mathbf{r}, q_2) = F(p, \mathbf{r}, q),$
 $E(p_i, \mathbf{r}, q_1) = 0.$

Cette construction est illustrée dans la Figure 3.24.

L'automate \mathcal{B} est bien un revêtement de \mathcal{A} , avec la fonction de revêtement qui envoie p_1 et p_2 sur p. Il est de plus immédiat que \mathcal{B} est \mathbf{r} -local.

- (a) Un automate unidirectionnel A.
- (b) Un revêtement \mathbf{r} -local de \mathcal{A} .

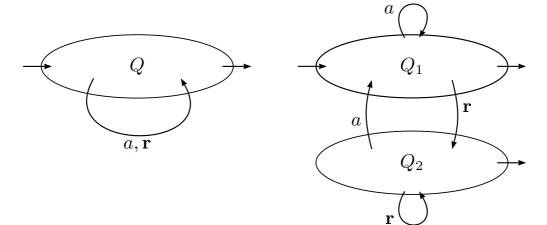


FIGURE 3.24 – Schéma de la construction de $\mathcal B$ r-local.

Élimination d'états dans un automate r-local

Appliquons donc maintenant l'algorithme d'élimination d'états sur un automate **r**-local, afin d'obtenir des expressions pré-Hadamard.

Dans cet algorithme, l'ordre d'élimination des sommets joue un rôle important dans la forme du résultat : on présente ici une variante de l'algorithme qui nous permet d'obtenir des expressions pré-Hadamard.

Elle se déroule comme suit : on élimine tout d'abord les états dont les transitions entrantes sont différentes de **r**. De plus, à chaque nouvelle expression créée par l'algorithme, on applique les règles de réécriture suivantes, afin de toujours manipuler des expressions pré-Hadamard :

$$\begin{aligned} \mathsf{E}\mathbf{r} + \mathsf{F}\mathbf{r} &\to (\mathsf{E} + \mathsf{F})\mathbf{r} \\ \mathsf{E}(k\mathbf{r}) &\to (\mathsf{E}k)\mathbf{r} \\ k(\mathsf{E}\mathbf{r}) &\to (k\mathsf{E})\mathbf{r} \\ \mathsf{E}(\mathsf{F}\mathbf{r}) &\to (\mathsf{E}\mathsf{F})\mathbf{r}. \end{aligned} \tag{3.49}$$

On a donc la proposition suivante:

Proposition 21. La sortie de la variante de l'algorithme d'élimination d'états appliqué à un automate \mathbf{r} -local \mathcal{A} est une expression pré-Hadamard $\mathsf{E}_{\mathcal{A}}$ telle que $[\![\mathsf{E}_{\mathcal{A}}]\!] = |\mathcal{A}|$.

Démonstration. Ni l'ordre d'élimination des états ni les diverses règles de réécriture ne changent l'interprétation de l'expression obtenue. Celle-ci est donc bien égale au comportement de l'automate.

Assurons-nous donc que l'on obtient bien une expression pré-Hadamard. Soit $\mathcal{A} = (Q_1 \cup Q_2, E, I, T)$ un automate **r**-local, avec Q_1 et Q_2 la partition

des états témoignant de la \mathbf{r} -localité de l'automate. On notera R l'ensemble des sommets restant après chaque élimination.

On peut séparer l'application de l'algorithme en deux étapes : supprimer les états de Q_1 en premier, puis ceux de Q_2 ensuite.

Les propriétés suivantes sont des invariants lors de la première étape, pour tout $p \in R$:

$$\forall q_1 \in R \setminus Q_2, p \xrightarrow{\mathsf{E}} q_1 \implies \mathsf{E} \text{ est une expression rationnelle},$$

 $\forall q_2 \in Q_2, \ p \xrightarrow{\mathsf{E}} q_2 \implies \mathsf{E} = \mathsf{Fr} \text{ avec } \mathsf{F} \text{ une expression rationnelle}.$

En effet, les deux propositions sont vraies au début par définition de l'automate. Si l'on supprime un état entre p et q_1 , comme il s'agit nécessairement d'un état de Q_1 , alors les expressions E, F, G, et H de la Figure 3.22 sont toutes rationnelles, et également EF*G + H. S'il s'agit d'un état entre p et q_2 , alors grâce aux réécritures de l'équation (3.49), on obtient une expression rationnelle suivie de \mathbf{r} .

Lors de la deuxième étape, les propriétés suivantes sont invariantes, pour tout $p \in R$:

$$\forall q_2 \in Q_2 \cap R, \ p \xrightarrow{\mathsf{E}} q_2 \implies \mathsf{E} = \mathsf{Fr} \text{ avec } \mathsf{F} \text{ une expression pré-Hadamard,}$$

$$p \xrightarrow{\mathsf{E}} t_0 \implies \mathsf{E} \text{ est une expression pré-Hadamard.}$$

À la fin de l'algorithme, l'étiquette de l'arc (i_0, t_0) est bien une expression pré-Hadamard.

Ainsi, nous disposons d'un algorithme τ vérifiant l'hypothèse du théorème 3.2 : en l'appliquant, nous obtenons un algorithme transformant un automate circulaire en une expression de Hadamard.

Proposition 22. Soit A un automate circulaire et E l'expression pré-Hadamard obtenue en appliquant la variante de l'élimination d'états sur un revêtement r-local de 1 w A. Alors $\rho^{-1}(E)$ est une expression de Hadamard, et

$$\llbracket \rho^{-1}(E) \rrbracket = |\mathcal{A}|.$$

Extension à \mathbb{Q} , \mathbb{R} et \mathbb{C}

On s'intéresse dorénavant à des semi-anneaux qui, bien que non rationnel-lement additifs, présentent des propriétés intéressantes, et qui sont usuels : \mathbb{Q} , \mathbb{R} et \mathbb{C} .

Le but de cette section est de montrer que la clôture des séries rationnelles par le quotient de Hadamard est l'ensemble des séries de Hadamard, ainsi que d'étudier quelques problèmes de décision sur la validité des modèles. On peut tout d'abord remarquer que $\mathbb{Q}_+ \cup \{\infty\}$ et $\mathbb{R}_+ \cup \{\infty\}$ sont des semi-anneaux rationnellement additifs, avec pour tout $x \geq 1$, $x^* = \infty$. On rappelle que dans ces semi-anneaux, $\forall x$ tel que x < 1, on a $x^* = \frac{1}{1-x}$.

Ainsi, on a le résultat suivant :

Proposition 23. Soit s une série réalisée par un \mathbb{Q} -automate circulaire valide. Alors s est une série de Hadamard sur \mathbb{Q} .

Démonstration. Par la proposition 9, on a s_+ et s_- deux séries, chacune réalisable par un \mathbb{Q}_+ -automate positif valide, telles que $s=s_+-s_-$. Comme $\mathbb{Q}_+\cup\{\infty\}$ est rationnellement additif, par la proposition 12, ces deux séries sont des séries de Hadamard sur \mathbb{Q}_+ . De plus, comme l'automate réalisant s est valide, ∞ n'est le poids d'aucun mot. Ainsi, s est une série de Hadamard sur \mathbb{Q} .

Les séries sur \mathbb{C}

Pour étudier les séries sur \mathbb{C} , décrites par des expressions rationnelles ou de Hadamard, considérons tout d'abord une nouvelle opération sur les séries : le quotient de Hadamard.

Définition 3.13. Une série t sur \mathbb{C} est dite à support plein si son support ne s'annule jamais : $\forall w \in A^*, \ \langle t, w \rangle \neq 0$.

On peut définir pour une telle série son inverse de Hadamard : il s'agit de la série notée $t^{\textcircled{1}}$:

$$\forall w \in A^*, \ \langle t^{\textcircled{1}}, w \rangle = \frac{1}{\langle t, w \rangle}.$$

Comme t est à support plein, $t^{\scriptsize\textcircled{\tiny 1}}$ est bien définie, et est l'inverse de t pour le produit de Hadamard. On peut maintenant définir le quotient de Hadamard de deux séries s et t sur \mathbb{C} , noté $\circ \frac{s}{t}$, avec t une série à support plein : c'est la série telle que

$$\forall w \in A^*, \ \langle \circ \frac{s}{t}, w \rangle = \langle s \odot t^{\bigcirc}, w \rangle = \frac{\langle s, w \rangle}{\langle t, w \rangle}.$$

Problèmes de clôture

On s'intéresse maintenant au comportement des classes de séries définies auparavant vis-à-vis de l'application des différents opérateurs.

Proposition 24 ([Schützenberger, 1962]). Dans un semi-anneau commutatif, le produit de Hadamard de deux séries rationnelles est une série rationnelle.

L'idée de la preuve constructive de [Schützenberger, 1962] est d'effectuer le produit direct des automates, et de prouver que ce produit réalise bien le produit de Hadamard des séries réalisées par les deux automates initiaux.

Corollaire 4. La puissance ième pour le produit de Hadamard d'une série rationnelle est encore une série rationnelle.

Ce n'est pas nécessairement le cas de l'étoile de Hadamard. En effet :

Proposition 25. Il existe des séries de Hadamard qui ne sont pas rationnelles.

Démonstration. Soit s la série rationnelle décrite par l'expression $\left(\frac{a}{2}\right)^* \frac{a}{2} \left(\frac{a}{2}\right)^*$. On a donc pour tout entier k, $\langle s, a^k \rangle = \frac{k}{2^k}$, et $\langle s^{\circledast}, a^k \rangle = \frac{2^k}{k-2^k}$.

Les coefficients d'une série rationnelle sur une variable vérifient une relation récurrente linéaire (cf. par exemple [Berstel et Reutenauer, 2010]). Ce n'est pas le cas pour s^{\circledast} : elle n'est donc pas rationnelle.

Lemme 3. Dans un semi-anneau commutatif, toute série de Hadamard peut s'exprimer comme le quotient de Hadamard de deux séries rationnelles.

 $D\acute{e}monstration.$ Soit s une série de Hadamard. Montrons le résultat par induction sur s.

- s est rationnelle. Alors $s = o \frac{s}{A^*}$
- s = kt ou s = tk, avec $t = o\frac{p}{q}$. Alors on a $s = o\frac{kp}{q}$ ou $s = o\frac{pk}{q}$, selon le cas.
- $s = t \odot t'$, avec $t = \circ \frac{p}{q}$ et $t' = \circ \frac{p'}{q'}$. On a $s = \circ \frac{p \odot p'}{q \odot q'}$, qui sont toutes deux rationnelles, grace à la proposition 24.
- $s=t^{\circledast}$, avec $t=\frac{p}{q}$. Comme l'étoile de Hadamard de t est définie, pour tout mot w, on a $\langle p,w\rangle<|\langle q,w\rangle|$. En particulier, on a $\langle q-p,w\rangle\neq 0$. De plus, la série q-p est naturellement rationnelle. Ainsi, $\frac{1}{q-p}$ est

bien définie, et on a
$$s = o \frac{q}{q - p}$$
.

On souhaite également montrer que le quotient de Hadamard de deux séries rationnelles est une série de Hadamard. Pour cela, nous avons besoin de la série $\mathsf{Geom}(\lambda)$, qui à tout mot w associe $\frac{1}{\lambda^{|w|+1}}$, ainsi que du lemme suivants :

Lemme 4. Soit t une série \mathbb{Q} -rationnelle telle que pour tout mot w, on ait $\langle t, w \rangle > 0^{1}$. Alors il existe un nombre rationnel positif λ tel que la série $t' = A^* - t \odot \mathsf{Geom}(\lambda)$ est une série \mathbb{Q}_+ -rationnelle, dont chaque coefficient *est dans* [0; 1[.

 $D\'{e}monstration$. La série t est rationnelle, il existe donc un automate unidirectionnel la réalisant. Par le lemme 1, on peut supposer cet automate standard (un seul état initial de poids 1) et positif (toutes les transitions ont un poids positif).

Soit $\mathcal{A} = (Q, E, \chi_i^2, T)$ cet automate, avec $i \in Q$ le seul état initial.

Posons $M = \max(\max_{p,a} \sum_{q} E(p, a, q), \max_{p} T(p))$ et $\lambda = 1/M$. On définit $\mathcal{B} = (Q \cup \{\bot\}, E', \chi_i, T')$, avec \bot un nouvel état, tel que pour tout p, q de $Q \cup \{\bot\}$ et pour toute lettre a,

$$E'(p, a, q) = \begin{cases} E(p, a, q).\lambda & \text{si } p, q \in Q, \\ 1 - \sum_{r \in Q} E(p, a, r).\lambda & \text{si } p \in Q \text{ et } q = \bot, \\ 1 & \text{si } p = q = \bot, \\ 0 & \text{sinon.} \end{cases}$$
(3.50)

En fonction de T', l'automate \mathcal{B} peut réaliser différentes séries :

1. $T'(p) = T(p) \cdot \lambda$ pour tout p de Q et $T'(\bot) = 0$.

Dans ce cas, il y a une bijection naturelle entre les chemins de \mathcal{B} et ceux de \mathcal{A} . Le poids de chaque chemin d'étiquette w dans \mathcal{B} est égal au poids du chemin correspondant dans \mathcal{A} , multiplié par $\lambda^{|w|+1}$.

Ainsi,
$$\langle \mathcal{B}, w \rangle = \langle t, w \rangle . \lambda^{|w|+1}$$
.

2. T'(p) = 1 pour tout $p \text{ de } Q \cup \{\bot\}$;

Pour toute lettre a, pour tout état p, la somme des poids des transitions sortantes de p avec étiquette a est égale à 1. Comme il n'y a qu'un seul état initial et qu'il est de poids 1, le poids de chaque mot est de 1 dans \mathcal{B} : $\langle \mathcal{B}, w \rangle = 1.$

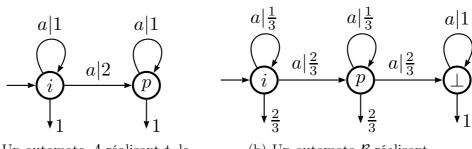
3. $T'(p) = 1 - T(p) \cdot \lambda$ pour tout état p de Q et $T'(\bot) = 1$.

Dans ce cas, le comportement de \mathcal{B} est la différence des deux précédents comportements: $\langle \mathcal{B}, w \rangle = 1 - \langle t, w \rangle \lambda^{|w|+1}$. Cette valeur est bien dans [0; 1[.

Dans le dernier cas, tous les poids de \mathcal{B} sont positifs; donc \mathcal{B} est un \mathbb{Q}_+ automate réalisant une série \mathbb{Q}_+ -rationnelle.

^{1.} La série t a donc un support plein.

^{2.} la fonction caractéristique de $\{i\}$



- (a) Un automate \mathcal{A} réalisant t, la série de exemple 11.
- (b) Un automate \mathcal{B} réalisant $A^* - t \odot \mathsf{Geom}(\frac{1}{2}).$

FIGURE 3.25 – Schéma de la construction du lemme 4.

Exemple 11

Soit t la série réalisée par l'automate \mathcal{A} décrit dans la Figure 3.25a. On a $\forall k \in \mathbb{N}, \langle t, a^k \rangle = 2k + 1$. Appliquons la construction de la preuve du lemme 4 à cette série. On a M=3, et donc $\lambda=\frac{1}{3}$. L'automate \mathcal{B} de la construction est en Figure 3.25b.

Proposition 26. Pour toute série \mathbb{Q} -rationnelle s à support plein, il existe un nombre rationnel positif λ tel que l'étoile de Hadamard de $A^* - s \odot s \odot \mathsf{Geom}(\lambda)$ est définie et

$$\circ \frac{A^*}{s} = (A^* - s \odot s \odot \mathsf{Geom}(\lambda))^{\circledast} \odot s \odot \mathsf{Geom}(\lambda). \tag{3.51}$$

Démonstration. Si s est une série Q-rationnelle à support plein, $t = s \odot s$ est une série rationnelle dont tous les coefficients sont strictement positifs. Par le lemme 4, il existe un nombre rationnel positif λ tel que la série t'= $A^* - t \odot \mathsf{Geom}(\lambda)$ est une série \mathbb{Q}_+ -rationnelle, dont chaque coefficient est dans [0; 1[.

Son étoile de Hadamard est donc définie, et on a pour tout mot w:

$$\langle (A^* - t \odot \mathsf{Geom}(\lambda))^{\circledast}, w \rangle = \langle A^* - t \odot \mathsf{Geom}(\lambda), w \rangle^*$$
 (3.52)

$$= (1 - \langle t \odot \mathsf{Geom}(\lambda), w \rangle)^* \tag{3.53}$$

$$= \frac{1}{1 - (1 - \langle t \odot \mathsf{Geom}(\lambda), w \rangle)}$$

$$= \frac{1}{\langle t \odot \mathsf{Geom}(\lambda), w \rangle}$$
(3.54)

$$= \frac{1}{\langle t \odot \mathsf{Geom}(\lambda), w \rangle} \tag{3.55}$$

$$= \langle \circ \frac{A^*}{t \odot \mathsf{Geom}(\lambda)}, w \rangle \tag{3.56}$$

(3.57)

On peut donc écrire:

$$(A^* - t \odot \mathsf{Geom}(\lambda))^{\circledast} \odot s \odot \mathsf{Geom}(\lambda) = \circ \frac{A^*}{t \odot \mathsf{Geom}(\lambda)} \odot s \odot \mathsf{Geom}(\lambda)$$

$$(3.58)$$

$$= \circ \frac{A^*}{s \odot s \odot \mathsf{Geom}(\lambda)} \odot s \odot \mathsf{Geom}(\lambda)$$

$$(3.59)$$

$$= \circ \frac{A^*}{s}.$$

$$(3.60)$$

Les résultats du lemme 4 et de la proposition 26 s'étendent naturellement à \mathbb{R} . On ne peut les étendre directement à \mathbb{C} . Cependant, si avec s une série sur \mathbb{C} , on note \overline{s} la série qui associe à tout mot w le conjugué du complexe $\langle s, w \rangle$, on peut faire les remarques suivantes :

- Si s est \mathbb{C} -rationnelle, alors c'est aussi le cas pour \overline{s} .
- Si s est \mathbb{C} -rationnelle, $s \odot \overline{s}$ est une série \mathbb{C} -rationnelle dont tous les coefficients sont des réels positifs. Elle est à support plein si et seulement si s l'est.

Comme \mathbb{C} est une extension de Fatou de \mathbb{R} , par le lemme de Fatou ([Fliess, 1974]), on obtient que $s \odot \overline{s}$ est une série \mathbb{R} -rationnelle,

Ainsi, si s est à support plein, on peut appliquer le lemme 4 à la série $t = s \odot \overline{s}$: il existe un réel positif λ tel que $A^* - t \odot \mathsf{Geom}(\lambda)$ est une série dont l'étoile de Hadamard existe. On obtient donc :

Proposition 27 (proposition 26 sur \mathbb{C}). Pour toute série \mathbb{C} -rationnelle s à support plein, il existe un nombre réel positif λ tel que l'étoile de Hadamard de $A^* - s \odot \overline{s} \odot \text{Geom}(\lambda)$ est définie et

$$\circ \frac{A^*}{s} = (A^* - s \odot \overline{s} \odot \mathsf{Geom}(\lambda))^{\circledast} \odot \overline{s} \odot \mathsf{Geom}(\lambda). \tag{3.61}$$

Proposition 28. Sur \mathbb{C} , les séries de Hadamard sont exactement les quotients de Hadamard des séries rationnelles.

Démonstration. Conséquence immédiate du lemme 3 et de la proposition 27.

Ainsi, on pourra décrire une série de Hadamard sur \mathbb{C} grâce à deux représentations équivalentes : une paire de séries rationnelles, ou bien une expression de Hadamard. De même que pour que l'interprétation d'une expression de Hadamard soit définie il faut que la série existe, pour que la paire de série rationnelles définisse effectivement une série de Hadamard, il faut que la deuxième soit à support plein.

Proposition 29. L'ensemble des séries de Hadamard n'est pas clos par le produit de Cauchy.

 $D\'{e}monstration$. Soit s la série définie dans la preuve de la proposition 25. En étudiant le produit de Cauchy de s avec elle-même, on a :

$$\langle s.s, a^k \rangle = \sum_{i=0}^k \frac{1}{(i+1)} \cdot \frac{1}{(k-i+1)}$$

$$= \sum_{i=0}^k \frac{1+i-i+k-k+1-1}{(i+1)(k-i+1)}$$

$$= \sum_{i=0}^k \frac{i+1}{(i+1)(k-i+1)} + \frac{k-i+1}{(i+1)(k-i+1)} + \frac{-k-1}{(i+1)(k-i+1)}$$

$$= \sum_{i=0}^k \frac{1}{(k-i+1)} + \sum_{i=0}^k \frac{1}{(i+1)} - (k+1) \sum_{i=0}^k \frac{1}{(i+1)(k-i+1)}$$

$$= 2 \sum_{i=0}^k \frac{1}{(i+1)} - (k+1) \langle s.s, a^k \rangle$$

$$= \frac{2}{(k+2)} \sum_{i=1}^{k+1} \frac{1}{i}$$
(3.62)

Donc pour des entiers k suffisamment grands, on a :

$$\langle s.s, a^k \rangle = \frac{2}{k+2} \sum_{i=0}^k \frac{1}{i+1} \sim \frac{2 \ln k}{k}.$$
 (3.63)

Supposons maintenant que s.s est une série de Hadamard. C'est donc le quotient de deux séries rationnelles. Les coefficients de chacune de ces deux séries doivent satisfaire une relation récurrente linéaire : leur quotient ne sera donc jamais équivalent à $\frac{2\ln k}{k}$, et l'on obtient une contradiction.

Décidabilité

Sur ces objets, différents problèmes de décision peuvent être exprimés. On en présente ici quelques-uns. Les résultats d'indécidabilité sont exprimés sur

 \mathbb{Q} , mais s'étendent à \mathbb{C} , ou du moins à tout sous-semi-anneau représentable de \mathbb{C} . Toutefois, pondérer des automates par des nombres non calculables par machine de Turing n'est pas pertinent en général. Si nécessaire, on peut cependant les manipuler formellement.

Proposition 30 (cf. [Berstel et Reutenauer, 2010, Sakarovitch, 2009]). On ne peut décider si une série Q-rationnelle a un support plein.

Remarque 20. On peut décider si une série \mathbb{R}_+ -rationnelle a un support plein : il suffit de vérifier si l'automate support reconnait A^* . Ce problème étant décidable grâce à une simple minimisation, le résultat suit immédiatement.

Intéressons-nous maintenant à quelques problèmes de validité des représentations. Les deux représentations se comportent de la même manière dans le cas de \mathbb{Q} .

Proposition 31. La validité de la représentation d'une série de Hadamard $sur \mathbb{Q}$ est indécidable.

Démonstration. Même si les deux représentations sont équivalentes, nous allons prouver ce résultat dans les deux cas, en réduisant au problème de la décision de la validité deux probèmes indécidables.

- Dans le cas d'une représentation par une paire de séries rationnelles, la représentation est valide si et seulement si la série au dénominateur a un support plein, ce qui est indécidable, par la proposition 30.
- Dans le cas d'une représentation grâce à une série de Hadamard, on réduit un problème sur les automates probabilistes, ce qui nous donne le résultat sur les \mathbb{Q}_+ -automates. Le problème de l'acceptation d'un mot avec une probabilité supérieure ou égale à $\frac{1}{2}$ par un automate probabiliste avec poids dans $\{0,\frac{1}{2},1\}$ est indécidable [Paz, 1971]. Soit donc s la série sur \mathbb{Q}_+ réalisée par un automate de cette forme. L'expression $(2s)^{\circledast}$ est valide si et seulement si pour tout mot w, son poids dans 2s est dans] -1; 1[. Comme l'automate est positif, cela revient à exiger que le poids par s de chaque mot soit strictement plus petit que $\frac{1}{2}$, ce qui est indécidable.

De manière surprenante, dans le cas des séries sur \mathbb{Q}_+ , les deux problèmes vus ci-dessus ont des décidabilités différentes.

Proposition 32. La validité des expressions de Hadamard avec poids dans \mathbb{Q}_+ est indécidable.

Démonstration. La preuve dans le cas des séries sur \mathbb{Q} est en fait une preuve sur \mathbb{Q}_+ .

78 L.-M. Dando

Proposition 33. Soit $s = o \frac{t}{t'}$ une série de Hadamard avec t' une série $\mathbb{Q}+$ rationnelle. La validité de s est décidable.

Démonstration. Cela découle immédiatement de la décidabilité du support plein des séries sur \mathbb{Q}_+ .

Proposition 34. L'équivalence des expressions de Hadamard valides sur \mathbb{C} est décidable.

Démonstration. Soit E et E' deux expressions de Hadamard valides. On peut construire, grâce à la proposition 28 , quatre séries rationnelles s, s', t, t', telles que $\llbracket \mathsf{E} \rrbracket = \circ \frac{s}{t}$ et $\llbracket \mathsf{E}' \rrbracket = \circ \frac{s'}{t'}$. Pour montrer l'équivalence de E et de E', il suffit de montrer l'équivalence de $s \odot t'$ et de $s' \odot t$. Ces deux séries étant rationnelles, cela est décidable. □

Les automates circulaires sur \mathbb{C} .

Comme le comportement d'un automate circulaire sur \mathbb{C} est une série de Hadamard (cf. proposition 13), intéressons-nous maintenant à ces automates.

Proposition 35. Soit s une série \mathbb{C} -rationnelle (respectivement \mathbb{R} -rationnelle, respectivement \mathbb{Q} -rationnelle) à support plein. Il existe un automate circulaire valide sur \mathbb{C} (respectivement \mathbb{R} , respectivement \mathbb{Q}) dont le comportement est l'inverse de Hadamard de s.

Démonstration. D'après la proposition 27, il existe un nombre rationnel positif λ tel que la série $t = A^* - s \odot \overline{s} \odot \mathsf{Geom}(\lambda)$ est une série \mathbb{R}_+ -rationnelle, dont chaque coefficient est dans [0;1[. Cette série peut être réalisée par un automate unidirectionnel $\mathcal{A}=(Q,E,I,T)$ sur \mathbb{R}_+ , que l'on peut sans perte de généralité supposer standard.

Soit \mathcal{B} l'automate circulaire construit à partir de \mathcal{A} en ajoutant :

- une **r**-transition de chaque état final p vers l'unique état initial, de poids T(p),
- un nouvel état \perp , initial et final avec poids 1, telle que pour toute lettre de A, il y ait une boucle de poids 1 avec cette lettre sur \perp .

Cette construction est illustrée par les Figures 3.28 et 3.29.

Soit w un mot de A^* . Tout calcul étiquetté par w dans \mathcal{B} est soit une boucle sur \bot , de poids 1, soit la concaténation de calculs dans \mathcal{A} suivis d'une **r**-transition.

Comme le poids de chaque calcul dans \mathcal{A} est positif, et que pour tout mot w, la somme des poids des calculs d'étiquette w dans \mathcal{A} est strictement inférieure à 1, le poids d'un mot dans \mathcal{B} est défini.

Ce poids est l'étoile du poids de w dans \mathcal{B} . Ainsi, \mathcal{B} est valide, et son comportement est $(A^* - s \odot \overline{s} \odot \mathsf{Geom}(\lambda))^\circledast = \circ \frac{A^*}{s \odot \overline{s} \odot \mathsf{Geom}(\lambda)}$. On peut facilement construire un automate circulaire réalisant $(A^* - s \odot \overline{s} \odot \mathsf{Geom}(\lambda))^\circledast \odot s \odot \mathsf{Geom}(\lambda) = \circ \frac{A^*}{s}$.

Remarque 21. La construction naïve, consistant à ajouter des **r**-transitions partant des états finaux aux états initiaux d'un automate réalisant $A^* - s$ ne fonctionne pas. Soit par exemple \mathcal{A} l'automate de la Figure 3.27 réalisant la série $a^* - \left(\frac{a}{2}\right)^*$. Si l'on construit l'automate qui, après chaque état final, permet de retourner dans un état initial avec une **r**-transition, on obtient un automate non valide. En effet, pour tout mot il existe une infinité de chemins de poids -1.

Proposition 36. Toute série de Hadamard sur \mathbb{C} (respectivement \mathbb{R} , respectivement \mathbb{Q}) est réalisable par un automate circulaire sur \mathbb{C} (respectivement \mathbb{R} , respectivement \mathbb{Q}).

Démonstration. Soit s une série de Hadamard : elle est représentable par une paire de séries rationnelles (t,t'). On peut grâce à la proposition 35 construire un automate réalisant $\circ \frac{A^*}{t'}$, et de là, construire un automate réalisant $\circ \frac{A^*}{t'}$ $\odot t$, c'est-à-dire s.

Décidabilité

Comme il y a équivalence entre les \mathbb{Q} -automates circulaires et les expressions de Hadamard sur \mathbb{Q} (propositions 23 et 36), on peut déduire les deux résultats suivants :

Théorème 3.3. L'équivalence de deux \mathbb{C} -automates circulaires valides est décidable.

Cela découle de la décidabilité de l'équivalence entre deux expressions de Hadamard valides, en proposition 34.

Proposition 37. La validité d'un Q-automate circulaire est indécidable.

Cela découle de l'indécidabilité de la validité d'une \mathbb{Q} -expression de Hadamard, en proposition 31.

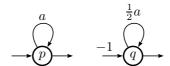


FIGURE 3.26 – Un automate A_1 réalisant $a^* - \left(\frac{a}{2}\right)^*$.

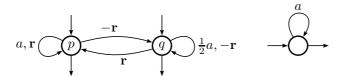


FIGURE 3.27 – Automate non valide, suivant la construction naïve.

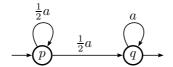


FIGURE 3.28 – Un automate standard et positif équivalent à A_1 .

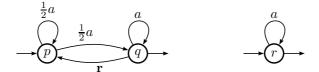


FIGURE 3.29 – Un automate A_2 réalisant l'inverse de Hadamard de $|A_1|$.

Chapitre 4

Automates boustrophédons sur les corps commutatifs

One ring to rule them all.

- The Lord of the Rings (1954),

J.R.R. Tolkien

Maintenant, et dans toute la suite du chapitre, on notera \mathbb{K} un corps commutatif, tel que \mathbb{Q} , \mathbb{R} et \mathbb{C} par exemple.

Nous nous sommes intéressés dans le chapitre précédent aux automates circulaires, et à ce que l'on pouvait dire d'eux dans les semi-anneaux rationnellement additifs et dans \mathbb{C} . Dans ce chapitre, nous allons nous interesser à l'expressivité des automates boustrophédons, en général plus puissants. En effet, même sur les semi-anneaux rationnellement additifs commutatifs, par exemple, il existe des séries réalisables par des automates boustrophédons, mais pas par des automates circulaires (cf. [Guillon, 2016]).

Le but de ce chapitre est de démontrer le théorème suivant :

Théorème 4.1. Soit \mathbb{K} un corps commutatif. Les séries sur \mathbb{K} réalisables par un automate boustrophédon valide sont des quotients de Hadamard de séries \mathbb{K} -rationnelles.

Ce résultat est effectif : à partir d'un automate boustrophédon, on peut construire deux automates unidirectionnels tels que le quotient de Hadamard des séries qu'ils réalisent est égal à la série réalisée par l'automate boustrophédon initial. Cette construction, inspirée de [Anselmo et Bertoni, 1992], a été publiée dans [Dando et Lombardy, 2018a]. Cet article présentait une faille dans la preuve : elle requiert que l'étoile d'une certaine matrice (que l'on décrira plus loin) soit définie. Or, il apparaît que la validité de l'automate n'est pas une condition suffisante pour garantir cette propriété. Il faut en fait que l'automate soit fortement valide (définition 2.13 page 27).

C'est pourquoi on montre d'abord que l'on peut, à partir d'un automate boustrophédon valide, construire un automate boustrophédon équivalent fortement valide.

Ensuite, nous prouverons le théorème pour les automates fortement valides. Afin d'établir ce théorème, nous allons tout d'abord montrer que l'on peut exprimer le poids d'un mot par un automate boustrophédon grâce à l'étoile d'une matrice.

On peut exprimer cette étoile en tant qu'inverse d'une matrice. De manière classique, chaque coefficient de l'inverse s'exprime comme le quotient d'un mineur et d'un déterminant; on montrera que chacune de ces valeurs peut être calculée par un automate unidirectionnel. On aura donc montré qu'il s'agit d'un quotient de Hadamard de deux séries K-rationnelles.

Par les propositions 28 et 36, on sait que sur \mathbb{C} , un tel quotient est réalisable par un automate circulaire. Toutes les étapes étant constructives, on peut donc construire, à partir d'un automate boustrophédon valide sur \mathbb{C} , un automate circulaire valide équivalent.

Construction de l'automate fortement valide

On souhaite donc construire un automate boustrophédon fortement valide équivalent à un automate boustrophédon valide donné. Pour ce faire, nous allons stocker de l'information supplémentaire dans les états de l'automate : les contextes.

Il s'agit de matrices représentant les chemins de l'automate et servant à décider si un mot w permet d'arriver à la position actuelle à partir d'un état initial pour les contextes gauches, et pour les contextes droits, d'atteindre un état final, à partir de la position actuelle. On montrera dans la suite que cette information est suffisante.

Dans la suite, on notera $\mathcal{A} = (Q_+, Q_-, E, I, T)$ l'automate boustrophédon valide dont on veut calculer un équivalent fortement valide.

Représentation booléenne et contextes

Pour tout mot u de A^* , on peut définir une matrice booléenne, $\overrightarrow{\mu(u)}$, représentant les chemins sur u de l'automate qui commencent à gauche et finissent à droite. Plus précisément, on pose $\overrightarrow{\mu(u)}_{p,q} = 1$, avec $p, q \in Q_+$, si et seulement s'il existe dans l'automate un chemin sur u allant de l'état p en position 0 à l'état q en position |u|. C'est donc une matrice de $M^{|Q_+|\times|Q_+|}(\mathbb{B})$.

On peut définir de la même manière trois autres matrices, $\mu(u)$ pour les chemins commençant à droite et finissant à gauche, dans $M^{|Q_-|\times|Q_-|}(\mathbb{B})$, $\mu(u)$ pour les chemins commençant à gauche et finissant à gauche, en lisant un

préfixe du mot, dans $M^{|Q_+|\times|Q_-|}(\mathbb{B})$, et $\mu(u)$ pour les chemins commençant à droite et finissant à droite, en lisant un suffixe du mot, dans $M^{|Q_-|\times|Q_+|}(\mathbb{B})$.

Pour tout mot $u \in A^*$, on notera:

$$\mu(u) = \begin{bmatrix} \overrightarrow{\mu(u)} & \overleftarrow{\mu(u)} \\ \overrightarrow{\mu(u)} & \overleftarrow{\mu(u)} \end{bmatrix}.$$

On peut remarquer que pour toute lettre a, la matrice $\mu(a)$ est la matrice d'adjacence du graphe sous-jacent de l'automate restreint à ses transitions étiquetées par a.

D'après [Birget, 1990], pour tous mots $u, v \in A^*$, on peut calculer $\mu(uv)$ à partir de $\mu(u)$ et de $\mu(v)$.

On a:

$$\overrightarrow{\mu(uv)} = \overrightarrow{\mu(u)} \left(\overrightarrow{\mu(v)} \overrightarrow{\mu(u)} \right)^* \overrightarrow{\mu(v)}, \tag{4.1}$$

$$\overrightarrow{\mu(uv)} = \overrightarrow{\mu(u)} + \overrightarrow{\mu(u)} \left(\overrightarrow{\mu(v)} \overrightarrow{\mu(u)} \right)^* \overrightarrow{\mu(v)} \overrightarrow{\mu(u)}, \tag{4.2}$$

$$\overrightarrow{\mu(uv)} = \overrightarrow{\mu(v)} + \overrightarrow{\mu(v)}\overrightarrow{\mu(u)} \left(\overrightarrow{\mu(v)}\overrightarrow{\mu(u)}\right)^* \overrightarrow{\mu(v)}, \text{ et}$$
(4.3)

$$\frac{\overleftarrow{\mu(uv)}}{\overleftarrow{\mu(u)}} = \frac{\overleftarrow{\mu(v)}}{\overleftarrow{\mu(u)}} \left(\frac{\overleftarrow{\mu(u)}}{\overleftarrow{\mu(v)}} \right)^* \frac{\overleftarrow{\mu(u)}}{\overleftarrow{\mu(u)}}.$$
(4.4)

Muni de cette opération (associative, cf. [Birget, 1990]), que l'on notera $\mu(u) \mathbf{z} \mu(v)$, l'ensemble des matrices obtenues par μ forme un monoïde fini, noté M. On a donc $\alpha \in M \Leftrightarrow \exists w \in A^*$, $\mu(w) = \alpha$. Le monoïde M est donc le monoïde de transition de l'automate boustrophédon. Si $Q_- = \emptyset$, on retrouve la notion de monoïde de transition pour un automate unidirectionnel. On peut étendre naturellement la définition de μ à $\mu(\vdash)$ et $\mu(\dashv)$.

On note I le vecteur ligne tel que $I_p=1$ si et seulement si p est initial, et T le vecteur colonne tel que $T_p=1$ si et seulement si p final. On définit également pour tout mot $u\in A^*$ un vecteur ligne \overrightarrow{L}_u , tel que $\overrightarrow{L}_{up}=1$, avec $p\in Q_+$ si et seulement s'il existe un chemin de $\mathcal A$ sur le mot $\vdash u$ allant d'un état initial en position 0 à l'état p en position |u|. Ainsi, $\overrightarrow{L}_\varepsilon=I$. De même, on définit pour tout mot $u\in A^*$ un vecteur colonne \overrightarrow{R}_u , pour les chemins sur u allant jusqu'à un état final, et l'on a $\overrightarrow{R}_\varepsilon=T$.

Exemple 12

Si l'on considère A_1 , l'automate de la Figure 4.1, on a

$$\mu(a) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 1 \end{bmatrix}, \quad \mu(b) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 1 & 0 \end{bmatrix}$$
(4.5)

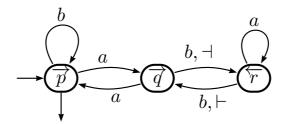


FIGURE 4.1 – L'automate A_1 .

$$\overrightarrow{\mu(\vdash)} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \ \overrightarrow{\mu(\dashv)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ I = \begin{bmatrix} 0 & 1 \end{bmatrix}, \ T = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{4.6}$$

On peut également définir les matrices $\overrightarrow{L_u}$ comme $\overrightarrow{\mu(\vdash u)}$, et $\overrightarrow{R_u} = \overrightarrow{\mu(u \dashv)}$. Il existe de la même manière une interprétation de ces matrices en terme de chemins : $(\overrightarrow{L_u})_{p,q}$, avec $p \in Q_-$ et $q \in Q_+$, vaut 1 si et seulement s'il existe un chemin sur le mot $\vdash u$ allant de l'état p en position |u| à l'état q en position |u|; et symétriquement $(\overrightarrow{R_u})_{p,q}$, avec $p \in Q_+$ et $q \in Q_-$, vaut 1 si et seulement s'il existe un chemin sur le mot $u \dashv$ allant de l'état p en position 0 à l'état q en position 0.

Ainsi, avec $u \in A^*$, on appellera contexte gauche une matrice de la forme $\begin{bmatrix} \overrightarrow{L_u} \\ \overrightarrow{L_u} \end{bmatrix}$, et contexte droit une matrice de la forme $\begin{bmatrix} \overrightarrow{R_u} | \overleftarrow{R_u} \end{bmatrix}$.

On pose
$$\mathcal{I} = \begin{bmatrix} \overrightarrow{L_{\varepsilon}} \\ \overrightarrow{L_{\varepsilon}} \end{bmatrix}$$
 et $\mathcal{T} = \begin{bmatrix} \overrightarrow{R_{\varepsilon}} | \overleftarrow{R_{\varepsilon}} \end{bmatrix}$.

On peut naturellement étendre la définition du produit z au produit entre un contexte et une matrice, qui nous donne un contexte de même latéralité et également entre un contexte gauche L_u et un contexte droit R_v , ce qui nous donne un booléen qui vaut vrai s'il existe un calcul de poids non nul sur uv dans l'automate. On peut vérifier qu'on a bien $L_{uv} = L_u z \mu(v)$, et également $R_{uv} = \mu(u) z R_v$.

On note $\mathcal{C}_{\mathscr{L}}$ l'ensemble des contextes gauches, que l'on peut voir comme $\{\mathcal{I} \mathsf{z} \gamma | \gamma \in M\}$, et $\mathcal{C}_{\mathscr{R}}$ l'ensemble des contextes droits, égal à $\{\gamma \mathsf{z} \mathcal{T} | \gamma \in M\}$. Dans la suite, on omettra la notation z lors du produit.

Définition de l'automate des contextes

Munis de ces objets, nous pouvons maintenant définir l'automate des contextes de \mathcal{A} , l'automate $\mathcal{B} = (\{i_0\} \cup S_+, S_-, E', \chi_{i_0}, T')$.

86 L.-M. Dando

^{1.} Les positions sont relatives à u : si w = vu, il faut leur ajouter |v| pour obtenir les positions absolues dans le mot.

L'idée de cet automate, inspirée de [Carton, 2012], est de mémoriser dans les états deux contextes : le contexte gauche ayant permis d'arriver dans cet état à partir d'un état initial, et le contexte droit qui permet d'accéder à un état final. Ainsi, on aura des états de la forme (p, L, R), avec $L \in \mathcal{C}_{\mathscr{L}}$ et $R \in \mathcal{C}_{\mathscr{R}}$, tels que tout calcul sur le mot uv à la position |u| est dans un état de la forme $(p, \mathcal{I}\mu(u), \mu(v)\mathcal{T})$.

Esquisse de la construction

Avant de définir formellement l'automate, on décrit la façon dont les deux contextes sont mis à jour. En fonction du type des états, différents éléments doivent être calculés.

Si les deux états forment un demi-tour, comme on ne change pas de position dans le mot, on ne change pas le contexte, il faut juste vérifier que le contexte permet de faire ce demi-tour.

Si les deux états (p, L, R) et (q, L', R') sont de même type, alors au moins une des deux composantes peut être immédiatement mise à jour. Supposons par exemple que les états p et q soient forward : la deuxième composante peut être mise à jour sans difficulté et vaut $L\mu(a)$. Ainsi, si l'on a $L = \mathcal{I}\mu(u)$, alors on a $L' = \mathcal{I}\mu(ua)$, ce qui est bien la propriété désirée. De plus, si $\mu(a)$ est inversible (pour le morphisme), alors on peut également mettre immédiatement à jour le troisième élément de l'état : $R' = \mu(a)^{-1}R$.

Si l'on n'a pas cette inversibilité, l'automate va avancer dans le mot en calculant une information sur la partie du mot restant à lire, qui permettra de mettre à jour correctement le contexte, ainsi que de revenir à la bonne position dans le mot. Une fois revenu à la bonne position, l'automate met à jour le deuxième élément, et passe à la transition suivante. On peut donc toujours calculer localement l'état suivant.

Il faut également pouvoir initialiser le calcul, car l'on souhaite commencer dans un état de la forme $(p, \mathcal{I}, \mu(w)\mathcal{T})$, avec w le mot d'entrée. Pour calculer $\mu(w)\mathcal{T}$, l'automate va se placer à la fin du mot, et revient en stockant l'image par μ du suffixe lu. Une fois à la gauche du mot, on peut commencer la simulation.

Les états de l'automate des contextes

L'automate des contextes est donc constitué de trois types d'états :

- les états d'initialisation, notés S^0 ,
- les états de simulation, notés S^1 , simulant directement les états de \mathcal{A} ,
- les états d'anticipation, notés S^2 , pour calculer un pseudo-inverse lorque la matrice de la lettre lue n'est pas inversible,

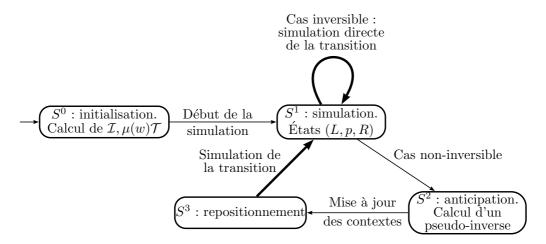


FIGURE 4.2 – Schéma du fonctionnement de l'automate des contextes.

• et les états de repositionnement, notés S^3 , permettant de replacer la tête de lecture à la bonne position après le calcul d'un pseudo-inverse.

On trouvera en Figure 4.2 une explication schématique de l'enchaînement des différents états lors d'un calcul.

Les états forward. L'automate des contextes dispose d'un unique état initial de poids $1:i_0$. Les autres états forward sont décomposables en trois catégories :

- S_+^1 , un ensemble d'états contextualisés, dans $Q_+ \times \mathcal{C}_{\mathscr{L}} \times \mathcal{C}_{\mathscr{R}}$, qui simulent les états de Q_+ .
 - On souhaite que, lorsque l'automate est en position |u| sur le mot uv, ces états soient de la forme $(p, \mathcal{I}\mu(u), \mu(v)\mathcal{T})$.
- S_+^2 , un ensemble d'états d'anticipation, dans $Q_+ \times \mathcal{C}_{\mathscr{L}} \times \mathcal{C}_{\mathscr{R}} \times M \times M$. Ces états sont de la forme $(p, \mathcal{I}\mu(ua), \mu(v)\mathcal{T}, \mu(a), \mu(v_0))$, avec v_0 un préfixe de v, lorsque l'automate est en position $|uav_0|$ sur le mot uav, et que $\mu(a)$ n'est pas inversible.
- et S_+^3 , un ensemble d'états de repositionnement, dans $Q_- \times \mathcal{C}_{\mathscr{L}} \times \mathcal{C}_{\mathscr{R}} \times M \times M$.

Ces états sont de la forme $(p, \mathcal{I}\mu(u), \mu(v)\mathcal{T}, \mu(u_1a), \mu(u_2))$, avec $u = u_0u_1a$ et u_2 un préfixe de u_1 , lorsque l'automate est en position $|u_0u_2|$ sur le mot uv, et que $\mu(a)$ n'est pas inversible.

Les seuls états finaux sont les états de S^1_+ de la forme (p, L, \mathcal{T}) : ils sont finaux de poids T(p).

Les états *backward*. Les états *backward* de l'automate des contextes sont décomposables en quatre catégories :

• S_{-}^{0} , un ensemble d'états d'initialisation permettant de calculer au départ l'image par μ de w, dans $\mathcal{C}_{\mathcal{R}}$.

Ces états sont de la forme $(p, \mu(v)\mathcal{T})$, lorsque l'automate est en position |u| sur le mot uv.

• S_{-}^1 , un ensemble d'états contextualisés, dans $Q_{-} \times \mathcal{C}_{\mathscr{L}} \times \mathcal{C}_{\mathscr{R}}$, qui simulent les états de Q_{-} .

On souhaite que ces états soient de la forme $(p, \mathcal{I}\mu(u), \mu(v)\mathcal{T})$, lorsque l'automate est en position |u| sur le mot uv.

- S_{-}^2 , un ensemble d'états d'anticipation, dans $Q_{-} \times \mathcal{C}_{\mathscr{L}} \times \mathcal{C}_{\mathscr{R}} \times M \times M$, Ces états sont de la forme $(p, \mathcal{I}\mu(u), \mu(av)\mathcal{T}, \mu(a), \mu(u_1))$, avec $u_0u_1 = u$, lorsque l'automate est en position $|u_0|$ sur le mot uav, et que $\mu(a)$ n'est pas inversible.
- et S_{-}^3 , un ensemble d'états de repositionnement, dans $Q_{-} \times \mathcal{C}_{\mathscr{L}} \times \mathcal{C}_{\mathscr{R}} \times M \times M$.

Ces états sont de la forme $(p, \mathcal{I}\mu(u), \mu(v)\mathcal{T}, \mu(u_1a), \mu(v_0))$, avec v_0 un préfixe de v, lorsque l'automate est en position $|uv_0|$ sur le mot uv, et que $\mu(a)$ n'est pas inversible.

Dans un souci de simplicité et de pédagogie, on notera \overrightarrow{p} si $p \in S_+$ et \overleftarrow{p} si $p \in S_-$.

Propriétés supplémentaires Les états de S^1_+ , de la forme (p,L,R), doivent vérifier les propriétés suivantes :

$$\begin{cases} (\mathcal{I}.\overrightarrow{L}(\overrightarrow{R}\overrightarrow{L})^*)_p &= 1\\ ((\overrightarrow{R}\overrightarrow{L})^*\overrightarrow{R}.\mathcal{T})_p &= 1 \end{cases}$$

Ceux de S_{-}^{1} , de la forme (p, L, R), doivent vérifier les propriétés suivantes :

$$\begin{cases} (\mathcal{I}.\overrightarrow{L}(\overrightarrow{R}\overrightarrow{L})^*\overrightarrow{R})_p &= 1\\ (\overrightarrow{L}(\overrightarrow{R}\overrightarrow{L})^*\overrightarrow{R}.\mathcal{T})_p &= 1 \end{cases}$$

Par ces conditions, on s'assure grâce aux contextes que l'état est bien accessible et co-accessible dans \mathcal{A} .

Les transitions de l'automate des contextes

Le but de cet automate est de simuler l'automate original \mathcal{A} . Pour simuler une de ses transitions (p, a, q) de poids k, avec p et q forward, en partant d'un état (p, L, R), plusieurs cas se présentent :

- soit $\mu(a)$ est inversible, auquel cas la transition est simulée par une seule transition de (p, L, R) à $(q, L\mu(a), \mu(a)^{-1}R)$, de poids k.
- Soit $\mu(a)$ n'est pas inversible, et l'on continue à lire l'entrée vers la droite jusqu'à trouver un mot v_0 pour lequel il existe un pseudo-inverse γ tel que $\gamma\mu(av_0) = \mu(v_0)$. Il faut donc arriver dans l'état $(q, L\mu(a), \gamma R)$ pour finir la simulation de la transition (p, a, q), mais il faut auparavant ramener la tête de lecture à la bonne position dans le mot. Cela se fait en lisant un suffixe v_1 de v_0 toujours plus grand jusqu'à obtenir que $\mu(v_1) = \mu(av_0)$. Ce chemin est illustré en Figure 4.3. La preuve de l'exactitude de cette méthode de repositionnement est en proposition 38.
- Si un tel mot v_0 n'existe pas, on lit tout le suffixe v du mot d'entrée restant à lire. L'état à rejoindre est alors $(q, L\mu(a), \mu(v)\mathcal{T})$. Le repositionnement se fait de manière identique au cas précédent.

Le raisonnement est similaire dans le cas où p et q sont des états back-ward. Si p et q ont des directions opposées, on passe directement de (p, L, R)à (q, L, R) avec poids k.

Voyons maintenant leur définition plus formelle. Certaines transitions ont dans leur définition des conditions du type "il existe un \mathcal{L} dans $\mathcal{C}_{\mathscr{L}}$ tel que ... ", ou bien "il existe un \mathcal{R} dans $\mathcal{C}_{\mathscr{R}}$ tel que ... ". Ces conditions assurent que la lettre lue respecte le contexte de l'état. On pose $a \in A$, $L \in \mathcal{C}_{\mathscr{L}}$ et $R \in \mathcal{C}_{\mathscr{R}}$.

Il existe six grands types de transitions:

- Les transitions d'initialisation, au début de l'automate :
 - $-E'(i_0, a, i_0) = 1$: on lit le mot de gauche à droite.
 - $-E'(i_0, \dashv, \mathcal{T}) = 1$: on fait demi-tour à la fin du mot, et on passe dans les états de $Q_{-,0}$, qui stockent l'image du mot par μ .
 - $-\ E'(R,a,\mu(a)R)=1$: en revenant, on met à jour l'image du suffixe.
 - $-E'(R, \vdash, \overrightarrow{(p, \mathcal{I}, R)}) = I(p)$, avec $p \in Q_+$: on initialise l'état avec le contexte calculé.
- les transitions de demi-tour sur les états de simulation (ceux de S^1), qui vérifient simplement que le demi-tour est compatible avec le contexte :
 - Si $p \in Q_+$ et $q \in Q_-$, alors $E'(\overline{(p, L, T)}, \dashv, \overline{(q, L, T)}) = E(p, \dashv, q)$, et $E'(\overline{(p, L, R)}, a, \overline{(q, L, R)}) = E(p, a, q)$ si et seulement si $\exists \mathcal{R} \in \mathcal{C}_{\mathscr{R}}$ tel que $\mu(a)\mathcal{R} = R$.

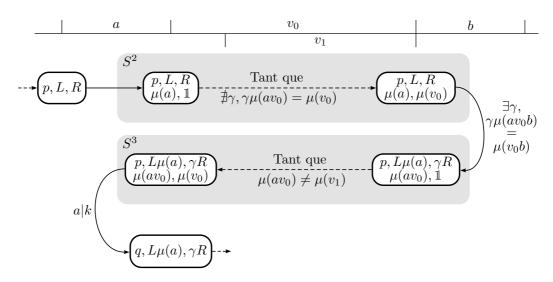


FIGURE 4.3 – Chemin calculant un pseudo-inverse.

- Si $p \in Q_-$ et $q \in Q_+$, alors $E'((p, \mathcal{I}, R), \vdash, (q, \mathcal{I}, R)) = E(p, \vdash, q)$, et E'((p, L, R), a, (q, L, R)) = E(p, a, q) si et seulement si $\exists \mathcal{L} \in \mathcal{C}_{\mathscr{L}}$ tel que $\mathcal{L}\mu(a) = L$.
- les transitions pour lesquelles $\mu(a)$ est inversible :

- Si
$$p, q \in Q_+$$
, alors $E'((\overline{p, L, R}), a, (\overline{q, L\mu(a), \mu(a)^{-1}R})) = E(p, a, q)$.

- Si $p, q \in Q_-$, alors $E'((p, L, R), a, (q, L\mu(a)^{-1}, \mu(a)R)) = E(p, a, q)$.
- les transitions pour lesquelles $\mu(a)$ n'est pas inversible, et avec lesquelles on passe donc dans les états d'anticipation (ceux de S^2):
 - Si $p \in Q_+$, $E'(\overrightarrow{(p, L, R)}, a, \overrightarrow{(p, L\mu(a), R, \mu(a), 1)}) = 1$ si et seulement si $\exists \mathcal{R} \in \mathcal{C}_{\mathscr{R}}$ tel que $\mu(a)\mathcal{R} = R$.
 - Si $p \in Q_-$, $E'(\overleftarrow{(p,L,R)}, a, \overleftarrow{(p,L,\mu(a)R,\mu(a),1)}) = 1$, si et seulement si $\exists \mathcal{L} \in \mathcal{C}_{\mathscr{L}}$ tel que $\mathcal{L}\mu(a) = L$.
- les transitions de calcul intermédiaire simulant une transition forward :
 - $-E'(\overrightarrow{(p,L,R,\alpha,\gamma)},a,\overrightarrow{(p,L,R,\alpha,\gamma\mu(a))}) = 1, \text{ avec } \alpha,\gamma \in M, \ p \in Q_+$ si $\gamma\mu(a)$ et $\alpha\gamma\mu(a)$ ne sont pas \mathcal{L} -équivalents, et si $\exists \mathcal{R} \in \mathcal{C}_{\mathscr{R}}$ tel que $\alpha\gamma\mu(a)\mathcal{R} = R$.
 - $-E'((\overline{(p,L,R,\alpha,\beta)},a,(\overline{p,L,\gamma R,\alpha\beta,\mathbb{1}}))=1,\,\text{avec}\,\,\alpha,\beta\in M\,\,\text{et}\,\,p\in Q_+,\\\text{s'il existe}\,\,\gamma\in M\,\,\text{tel que}\,\,\beta\mu(a)=\gamma\alpha\beta\mu(a)\,\,(i.e.\,\,\beta\mu(a)\sim_{\mathcal{L}}\alpha\beta\mu(a)),\\\text{et si}\,\,\exists\mathcal{R}\in\mathcal{C}_{\mathscr{R}}\,\,\text{tel que}\,\,\alpha\beta\mu(a)\mathcal{R}=R\,:\,\text{on fait demi-tour}.$

- $-E'((\overline{(p,L,R,\alpha,\beta)}, \dashv, (\overline{p,L,\beta\mathcal{T},\alpha\beta,1})) = 1$, avec $\alpha,\beta \in M$ et $p \in Q_+$: on est arrivé à la fin du mot, on fait demi-tour.
- $-E'((p, L, R, \beta, \zeta), a, (p, L, R, \beta, \mu(a)\zeta)) = 1$, avec $\beta, \zeta \in M$, $p \in Q_+$, si $\mu(a)\zeta \neq \beta$ et si $\exists \gamma \in M$ tel que $\gamma\mu(a)\zeta = \beta$: on retourne par des états backward vers la bonne position.
- $-E'(\overleftarrow{(p,L,R,\beta,\zeta)},a,\overrightarrow{(q,L,R)})=E(p,a,q)$, avec $\beta,\zeta\in M,p,q\in Q_+$, si $\mu(a)\zeta=\beta$: on fait demi-tour sur la lettre qui nous a fait rentrer dans le calcul intermédiaire.
- les transitions de calcul intermédiaire simulant une transition backward :
 - $-E'(\overleftarrow{(p,L,R,\alpha,\gamma)},a,\overleftarrow{(p,L,R,\alpha,\mu(a)\gamma)})=1$, avec $\alpha,\gamma\in M,\ p\in Q_-$ si $\mu(a)\gamma$ et $\mu(a)\gamma\alpha$ ne sont pas \mathcal{R} -équivalents, et si $\exists \mathcal{L}\in\mathcal{C}_{\mathscr{L}}$ tel que $\mathcal{L}\mu(a)\gamma\alpha=L$.
 - $-E'(\overleftarrow{(p,L,R,\alpha,\beta)},a,\overrightarrow{(p,L\gamma,R,\beta\alpha,\mathbb{1})})=1, \text{ avec } \alpha,\beta\in M \text{ et } p\in Q_{-}$ s'il existe $\gamma\in M$ tel que $\mu(a)\beta=\mu(a)\beta\alpha\gamma$ (i.e. $\mu(a)\beta\sim_{\mathcal{R}}\mu(a)\beta\alpha$), et si $\exists \mathcal{L}\in\mathcal{C}_{\mathscr{L}}$ tel que $\mathcal{L}\mu(a)\beta\alpha=L$: on fait demi-tour.
 - $-E'((p, L, R, \alpha, \beta), \vdash, (p, \mathcal{I}\beta, R, \beta\alpha, \mathbb{1})) = 1$, avec $\alpha, \beta \in M$, $p \in Q_-$: on est arrivé au début du mot, on fait demi-tour.
 - $-E'(\overline{(p,L,R,\beta,\zeta)},a,\overline{(p,L,R,\beta,\zeta\mu(a))})=1$, avec $\beta,\zeta\in M,\ p\in Q_-$, si $\zeta\mu(a)\neq\beta$ et si $\exists\gamma\in M$ tel que $\zeta\mu(a)\gamma=\beta$: on retourne par des états forward vers la bonne position.
 - $-E'(\overline{(p,L,R,\beta,\zeta)},a,\overline{(q,L,R)})=E(p,a,q),$ avec $\beta,\zeta\in M,$ $p,q\in Q_-,$ si $\zeta\mu(a)=\beta$: on fait demi-tour sur la lettre qui nous a fait rentrer dans le calcul intermédiaire.

Exemple 13

L'automate des contextes obtenu à partir de l'automate de la Figure 4.1 se trouve en Figure 4.4.

Propriétes de l'automate des contextes

Proposition 38. Soit w un mot de A^* , et ρ un calcul de \mathcal{B} sur w.

Pour tous les états (p, L, R) de S^1 à la position |u| dans ρ , on a $L = \mathcal{I}\mu(u)$ et $R = \mu(v)\mathcal{T}$.

Démonstration. La preuve se fait par récurrence sur les états (p, L, R) consécutifs dans ρ .

Tout calcul sur \mathcal{B} commence par un chemin déterministe, de l'état initial i_0 jusqu'à l'état \mathcal{T} , puis de cet état à l'état $\mu(w)\mathcal{T}$. On va ensuite dans un

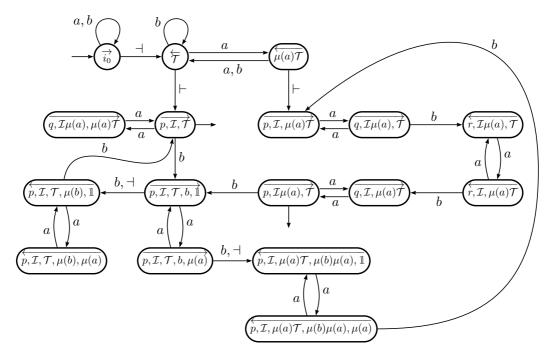


FIGURE 4.4 – Automate des contextes de A_1 .

état $(p, \mathcal{I}, \mu(w)\mathcal{T})$, qui est le premier état de S^1 rencontré. On est bien à la position 0, donc la propriété est vérifiée dans ce cas.

Supposons maintenant qu'elle soit vraie dans un état donné : soit l'on fait demi-tour, soit on passe à la position suivante, soit on passe à la position précédente. Le raisonnement pour passer à la position précédente étant similaire à celui pour la position suivante, il ne sera pas détaillé.

Si l'on fait demi-tour, on passe dans un état dont les deuxièmes et troisièmes composantes sont identiques à celles de l'état précédent, et on ne change pas de position : la propriété est donc toujours vérifiée.

Supposons que l'on se trouve dans un état forward (p, L, R) à la position |u| dans w, avec $a \in A \cup \{\exists\}$, $u, v \in A^*$ tels que uav = w, $L = \mathcal{I}\mu(u)$ et $R = \mu(av)\mathcal{T}$. Pour passer à la position suivante, le chemin suivi par le calcul est déterminé par l'inversibilité de $\mu(a)$.

Si $\mu(a)$ est inversible, alors on passe dans un état $(p, L\mu(a), \mu(a)^{-1}R)$. Comme $\mu(a)^{-1}R = \mu(v)\mathcal{T}$, on a donc la propriété encore vérifiée pour cet état. Si $\mu(a)$ n'est pas inversible, alors on passe dans l'état $(p, L\mu(a), R, \mu(a), 1)$.

On lit ensuite la lettre suivante, b, et si $\mu(b) \nsim_{\mathcal{L}} \mu(ab)$ on passe dans l'état $(p, L\mu(a), R, \mu(a), \underline{\mu(b)})$, jusqu'à ce que, après avoir lu un suffixe v_0 de v, on arrive dans un état $(p, L\mu(a), R, \mu(a), \mu(v_0))$, avec $\mu(v_0) \nsim_{\mathcal{L}} \mu(av_0)$. Pour chaque nouvelle lettre b lue, soit l'on met à jour v_0 , soit on a $\mu(v_0b) \sim_{\mathcal{L}} \mu(av_0b)$, soit $b = \dashv$.

Si $\mu(v_0b) \sim_{\mathcal{L}} \mu(av_0b)$, alors on passe de l'état $\overline{(p, L\mu(a), R, \mu(a), \mu(v_0))}$ à

l'état $(p, L\mu(a), \gamma R, \mu(av_0), \mathbb{1})$, avec γ tel que $\mu(v_0b) = \gamma \mu(av_0b)$.

De l'état $(p, L\mu(a), \gamma R, \mu(av_0), 1)$, on lit v_0 à l'envers en passant par des états $(p, L\mu(a), \gamma R, \mu(av_0), \mu(v_1))$ après avoir lu v_1 , suffixe de v_0 . On continue tant que $\mu(bv_1) \neq \mu(av_0)$, avec b la lettre à lire. Lorsque $\mu(bv_1) = \mu(av_0)$, on passe de l'état $(p, L\mu(a), \gamma R, \mu(av_0), \mu(v_1))$ à l'état $(q, L\mu(a), \gamma R)$.

Montrons maintenant que $\gamma R = \mu(v)\mathcal{T}$ et que nous sommes à la position |u|+1 dans w. On sait que $\mu(v_0b)=\gamma\mu(av_0b)$. Comme v_0b est un préfixe de v, il existe un mot v' tel que $v=v_0bv'$, donc $\mu(v)=mu(v_0b)\mu(v')=\gamma\mu(av_0b)\mu(v')=\gamma\mu(av)$. Ainsi, puisque $R=\mu(av)\mathcal{T}$, on a $\mu(v)\mathcal{T}=\gamma R$.

Il est immédiat que si $v_1 = v_0$, alors $\mu(bv_1) = \mu(av_0)$. Montrons par l'absurde qu'il n'existe pas de suffixe t de v_0 tel que $\mu(bt) = \mu(av_0)$, en posant $v_0 = t'bt$. On sait que $\mu(v_0) \nsim_{\mathcal{L}} \mu(av_0)$. Or $\mu(v_0) = \mu(t')\mu(av_0)$: contradiction. La seule possibilité est donc que l'on soit dans la position |u| + 1. Ainsi, la propriété est vraie dans l'état $(q, L\mu(a), \gamma R)$.

Il est possible qu'il n'existe pas un tel v_0 . Les états d'anticipation nous emmènent à la fin du mot, et nous avons donc la dernière coordonnée de l'état avant le demi-tour qui contient $\mu(v)$. Le raisonnement pour le repositionnement se déroule comme précédemment.

Ainsi, si le long d'un calcul de w on est dans un état (p, L, R) à la position |u|, avec w = uav, tel que $L = \mathcal{I}\mu(u)$ et $R = \mu(av)\mathcal{T}$, alors l'état suivant de S_3 est $(q, L\mu(a), \mu(v)\mathcal{T})$, avec q un état de Q.

Naturellement, pour la simulation d'une transition entre deux états backward, les conditions portent sur la \mathcal{R} -équivalence, mais le raisonnement est le même.

Montrons à présent que cet automate est fortement valide et équivalent à \mathcal{A} .

Lemme 5. Il existe une bijection entre les calculs de A et ceux de B. Cette bijection respecte les poids.

Démonstration. On construit l'application φ suivante entre les calculs de \mathcal{B} et ceux de \mathcal{A} : l'image d'un calcul de \mathcal{B} est la restriction de ce calcul aux états de S^1 , de forme (p, L, R), projetés ensuite sur leur première variable.

C'est bien une surjection : à partir d'un calcul ρ sur \mathcal{A} de w, un mot de A, on peut construire un calcul de \mathcal{B} d'image ρ . En effet, il existe un chemin de i_0 à $(p_0, \mathcal{I}, \mu(w)\mathcal{T})$, avec p_0 l'état initial de ρ .

Soit maintenant (p, L, R) un état de S_3 , avec $L = \mathcal{I}\mu(u)$ et $R = \mu(av)\mathcal{T}$, en posant w = uav, tel que (p, L, R) soit à la position |u| dans le calcul. Si dans ρ la transition suivante est (p, a, q), alors soit les deux états sont de types différents, dans ce cas il s'agit d'un demi-tour et on peut passer à (q, L, R) dans \mathcal{B} , soit les deux états sont de même type, supposons forward (dans l'autre sens, il s'agit du même raisonnement).

Dans ce cas, deux cas se présentent en fonction de l'inversibilité de $\mu(a)$. Soit $\mu(a)$ est inversible, et l'on peut aller dans $(q, L\mu(a), \mu(a)^{-1}R)$, soit $\mu(a)$ n'est pas inversible, et il existe un chemin d'états de S^2 et de S^3 menant à l'état $(p, L\mu(a), \mu(v)\mathcal{T}, \mu(av_0), \mu(v_0))$ en position |u| + 1, avec v_0 un préfixe de v tel que $\mu(av_0) \sim_{\mathcal{L}} \mu(v_0)$, à partir duquel on peut aller dans l'état $(q, L\mu(a), \mu(v)\mathcal{T})$.

Dans tous les cas, le chemin simulant E(p, a, q) est de même poids. De plus, si ρ finit sur l'état p, alors l'état $(p, \mathcal{I}\mu(w), \mathcal{T})$ est atteint par le calcul sur \mathcal{B} , de poids T(p). En fait, toute transition d'un calcul de \mathcal{A} est simulée par un chemin de \mathcal{B} entre deux états de S^1 , soit de longueur 1 si $\mu(a)$ est inversible, avec a la lettre lue, soit des chemins ne contenant que des états de S^2 puis des états de S^3 , en plus des deux extrémités du chemin.

Montrons qu'il s'agit également d'une injection. Soit ρ_1 et ρ_2 deux calculs de \mathcal{B} sur un mot w. Supposons qu'ils aient la même image par l'application définie ci-dessus, et notons p_i les états rencontrés successivement par cette image, avec p_0 initial. Comme ρ_1 et ρ_2 sont deux calculs sur le mot w, ils ont le même début, puisque \mathcal{B} n'est non-déterministe que pour les transitions menant vers des états de S^1 .

Comme ils ont la même image, ce chemin les emmène tous deux dans l'état $(p_0, \mathcal{I}, \mu(w)\mathcal{T})$. Ensuite, pour chaque état de S^1 , comme ils doivent tous deux lire la même lettre, ils ont même comportement : soit ils vont dans un autre calcul de S^1 , soit ils commencent un chemin composé d'abord d'états de S^2 puis d'états de S^3 . Dans ce cas, il est immédiat que ces deux chemins sont identiques, car déterministes et commençant du même état. De plus, l'état pour sortir de ce chemin est imposé par l'égalité de leur image.

Ainsi, on peut montrer que
$$\rho_1 = \rho_2$$
.

Corollaire 5. A et B sont équivalents.

$$D\acute{e}monstration$$
. Immédiat par le lemme 5.

Lemme 6. Soit (p, L, R) un état de S^1 .

S'il existe $u, v \in A^*$ tels que $\mathcal{I}\mu(u) = L$ et $\mu(v)\mathcal{T} = R$, alors il existe un calcul de \mathcal{B} d'étiquette uv qui passe par (p, L, R) en position |u|.

 $D\'{e}monstration.$ Par définition de l'état (p,L,R), on a les deux équations suivantes :

$$\left(I.\overrightarrow{\mu(u)}\left(\overrightarrow{\mu(v\dashv)}\overrightarrow{\mu(\vdash u)}\right)^*\right)_n = 1,$$
(4.7)

$$\left(\left(\overrightarrow{\mu(v \dashv)} \overrightarrow{\mu(\vdash u)} \right)^* \overrightarrow{\mu(v)} T \right)_p = 1. \tag{4.8}$$

Ainsi, $I\overline{\mu(uv)}T = 1$: il existe un calcul de \mathcal{A} d'étiquette uv, et il passe par p en position |u|. Comme il y a bijection des calculs entre \mathcal{A} et \mathcal{B} , il existe un calcul de \mathcal{B} d'étiquette uv passant par (p, L, R) à la position |u|.

Lemme 7. Soit ρ un circuit stationnaire de \mathcal{B} d'étiquette t.

De deux choses l'une :

- Soit $t = t' \dashv$, avec $t' \in A^* \cup \{\vdash\}$. Alors pour tout u préfixe propre de t, pour tout état (p, L, R) de S^1 de ce chemin en position |u|, $R = \mu(v)\mathcal{T}$, en posant $v = u^{-1}t$.
- Soit $t \in A^* \cup \{\vdash\}$. Alors il existe $w \in A^*$, tel que pour tout u préfixe propre de t, pour tout état (p, L, R) de S^1 de ce chemin en position |u|, $R = \mu(vw)\mathcal{T}$, en posant $v = u^{-1}t$.

Démonstration. La preuve est similaire dans les deux cas, même si le résultat s'écrit de façon différente. On montre le deuxième cas, $t \in A^*$.

Considérons une transition étique tée par la dernière lettre de t. Il s'agit nécessairement d'un de mi-tour. Soit $\overline{(p_0,L_0,R_0)}$ l'état de S^1 précédant cette transition dans ρ .

On note (p_i, L_i, R_i) le $i^{\grave{e}me}$ état de S^1 de ρ après (p_0, L_0, R_0) , et u_i le préfixe de t tel que (p_i, L_i, R_i) est en position $|u_i|$ dans ρ . On note également $v_i = u_i^{-1}t$. On montre maintenant le résultat par récurrence sur i.

Par définition de l'automate \mathcal{B} , les conditions d'existence des transitions de tous les demi-tours imposent : $\exists \mathcal{R} \in \mathcal{C}_{\mathscr{R}}$ tel que $R_0 = \mu(v_0)\mathcal{R}$. Par définition de $\mathcal{C}_{\mathscr{R}}$, $\exists w \in A^*$ tel que $R_0 = \mu(v_0w)\mathcal{T}$.

Supposons maintenant que $R_i = \mu(v_i)\mu(w)\mathcal{T}$, et montrons que $R_{i+1} = \mu(v_{i+1})\mu(w)\mathcal{T}$, quel que soit le type de p_i et de p_{i+1} , forward ou backward.

- S'ils sont de type différent, alors $R_{i+1} = R_i$ et $\mu(v_{i+1}) = \mu(v_i)$: la propriété est vérifiée.
- Si p_i et p_{i+1} sont tous deux des états de $Q_{-,3}$, alors $R_{i+1} = \mu(a)R_i$ et $v_{i+1} = av_i$, avec a la lettre lue. On a donc bien $R_{i+1} = \mu(v_{i+1})\mu(w)\mathcal{T}$. Cela ne dépend pas de l'inversibilité de $\mu(a)$.
- Si p_i et p_{i+1} sont tous deux des états de Q_+ , alors posons $v_{i+1} = av_i$. Si $\mu(a)$ est inversible, alors $R_{i+1} = \mu(a)^{-1}R_i = \mu(a)^{-1}\mu(av_{i+1})\mu(w)\mathcal{T}$. Sinon, le chemin entre les états (p_i, L_i, R_i) et $(p_{i+1}, L_{i+1}, R_{i+1})$ est un chemin d'états de S^2_+ , puis un chemin d'états de S^3_- , d'étiquette av'b, avec $a, b \in A$ et $v' \in A^*$ un préfixe de v_{i+1} .

Par définition des demi-tours de l'automate, $\exists \gamma \in M$ tel que $\gamma \mu(av'b) = \mu(v'b)$ et $R_{i+1} = \gamma R_i$.

On a donc $R_{i+1} = \gamma R_i = \gamma \mu(v_i) \mu(w) \mathcal{T} = \mu(v_{i+1}) \mu(w) \mathcal{T}$.

Le résultat se montre de la même manière du coté gauche :

96 L.-M. Dando

Lemme 8. Soit ρ un circuit stationnaire de \mathcal{B} d'étiquette t.

De deux choses l'une :

- Soit $t = \vdash t'$, avec $t' \in A^* \cup \{\exists\}$. Alors pour tout u préfixe de t', pour tout état (p, L, R) de S^1 de ce chemin en position $|u|, L = \mathcal{I}\mu(u)$.
- Soit $t \in A^* \cup \{\exists\}$. Alors il existe $w \in A^*$, tel que pour tout u préfixe de t, pour tout état (p, L, R) de S^1 de ce chemin en position |u|, $L = \mathcal{I}\mu(wu)$.

On peut donc maintenant montrer le résultat suivant :

Proposition 39. \mathcal{B} est fortement valide (dans un corps normé).

 $D\acute{e}monstration$. On montre ce résultat par l'absurde. Supposons que \mathcal{B} ne soit pas fortement valide. Il admet donc un circuit stationnaire ρ de poids x, correctement étiquetté par w, dont l'étoile n'est pas définie.

Ce circuit contient forcément des états de S_3 et peut-être des états de S_1 et S_2 . Soit (p, L, R) un état de S_+^1 de ce circuit 1 . Il existe u et v de A^* tel que uv est l'étiquette de ce circuit, avec (p, L, R) à la position |u| dans ce circuit.

Par les lemmes 7 et 8, il existe w_0 et w_1 de A^* tels que u est un suffixe de $\vdash w_0$ et v est un préfixe de $w_1 \dashv$, et que $L = \mathcal{I}\mu(w_0)$ et $R = \mu(w_1)\mathcal{T}$. Par le lemme 6, il existe donc un calcul d'étiquette w_0w_1 qui passe par (p, L, R) en position $|w_0|$.

Soit ρ_1 et ρ_2 deux chemins tels que ce calcul soit égal à la concaténation de ces deux chemins en (p, L, R), de poids respectifs x_1 et x_2 . Pour tout i entier, $\rho_1.(\rho)^i.\rho_2$ est donc un calcul de \mathcal{B} , d'étiquette w_0w_1 , de poids $x_1x^ix_2$. Tous ces calculs sont en bijection avec des calculs de \mathcal{A} de même poids sur le même mot. Ainsi, dans \mathcal{A} , il existe une famille de calculs sur le mot w_0w_1 dont la somme des poids est $\sum_{i\in\mathbb{N}} x_1x^ix_2$. Si cette somme était définie, alors serait aussi

définie $\sum_{i\in\mathbb{N}} x^i$, car nous sommes dans un corps. Or l'étoile de x n'est pas définie, et de plus, on sait que comme \mathbb{K} est normé, toute sous-famille d'une famille sommable est sommable.

Ainsi, la famille des poids des mots de w_0w_1 n'est pas sommable, et donc \mathcal{A} n'est pas un automate valide : contradiction.

Cette construction est la preuve du théorème suivant :

Théorème 4.2. Soit \mathbb{K} un corps muni d'une norme. Alors pour tout automate boustrophédon valide, il existe un automate boustrophédon fortement valide équivalent.

^{1.} un tel circuit contient nécessairement des états de S^1_+ et de S^1_- .

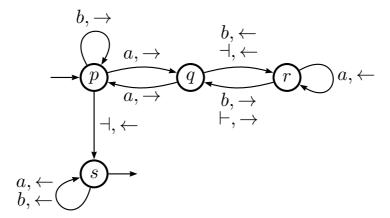


FIGURE 4.5 – L'automate A_1 qui servira d'exemple dans cette partie.

D'un automate boustrophédon fortement valide à un automate circulaire

Nous pouvons donc maintenant considérer que l'on travaille uniquement avec des automates boustrophédons fortement valides. On souhaite construire à partir d'un tel automate un automate circulaire qui lui est équivalent.

Description du modèle pour l'automate boustrophédon

Afin de simplifier les calculs, nous allons dans cette section utiliser un modèle d'automate boustrophédon qui n'est pas celui que nous avons utilisé jusqu'ici : dans ce modèle, les calculs commencent et finissent au début du mot. Ce modèle est cependant, ainsi que nous l'avons vu dans la sous-section 2.3.6 page 34, équivalent aux différents modèles usuels.

Plus formellement:

Définition 4.1. Un \mathbb{K} -automate boustrophédon est un quadruplet (Q, E, I, T), avec :

- Q l'ensemble fini des états,
- E la fonction de transition, de $Q \times (A \cup \{\vdash, \dashv\}) \times \{+1, -1\} \times Q$ dans \mathbb{K} .
- I et T les fonctions respectivement initiale et finale, de Q dans \mathbb{K} .

Exemple 14

L'automate A_1 de la Figure 4.5, qui est dans ce formalisme, servira d'exemple tout au long de cette partie.

Dans les transitions, la valeur +1 ou -1 indique la direction de la tête de lecture après avoir lu la lettre : +1, la tête de lecture se déplace vers la

droite, -1, elle se déplace vers la gauche. Il ne peut donc y avoir de transition étiquetée par $(\vdash, -1)$ ou par $(\dashv, +1)$.

Un chemin pour le mot $w = w_1 \dots w_k$ de A^* est une suite de transitions $\rho = (q_{i-1}, x_i, m_i, q_i)_{i \in [1,\ell]}$, avec (q_i) des états de Q, pour laquelle il existe une fonction pos de $[0,\ell]$ dans [0,k+1], indiquant la position de la tête de lecture.

Cette fonction vérifie les conditions suivantes :

- Pour tout i strictement positif, $pos(i) = pos(i-1) + m_i$.
- Pour tout i de $[1, \ell]$, si pos(i-1) = 0, alors $x_i = \vdash$, si pos(i-1) = k+1, alors $x_i = \dashv$, et sinon, $x_i = w_{pos(i-1)}$.

De plus, on appelle calcul sur w tout chemin sur w tel que q_0 est initial, q_l est final, pos(0) = 1 et $pos(\ell) = 0$. Le poids d'un calcul est :

$$I(p_0). \left(\prod_{i=1}^{\ell} E(p_{i-1}, x_i, m_i, p_i) \right) . T(p_{\ell}).$$

Comme d'habitude, le poids d'un mot w est égal à la somme des poids des calculs sur w. Dans la partie suivante, nous allons montrer que ce poids peut s'exprimer à l'aide de l'étoile d'une matrice.

Calcul de poids par une matrice

Une façon de représenter les automates pondérés est d'utiliser la matrice d'adjacence du graphe sous-jacent. Nous allons utiliser une extension classique de cette représentation : pour chaque lettre a de $A \cup \{\vdash, \dashv\}$, on définit deux matrices de taille $n \times n$, F(a) et B(a), représentants les chemins de taille 1, vers la droite pour F, et vers la gauche pour B. Plus formellement, avec p et q deux états, on pose $F(a)_{p,q} = E(p, a, +1, q)$ et $B(a)_{p,q} = E(p, a, -1, q)$. Notons que $F(\dashv) = B(\vdash) = 0$: aux extrémités du mot, on ne peut se déplacer que dans une seule direction.

On définira ensuite une matrice M_w , et deux vecteurs, I initial et T final, tels que IM_wT nous donne le poids des calculs de longueur 1, IM_w^2T le poids des calculs de longueur 2, et plus généralement pour tout entier i, IM_w^iT le poids des calculs de longueur i.

Dans un automate unidirectionnel, la longueur des calculs acceptant un mot donné est borné par la taille du mot. Dans un automate boustrophédon, il peut exister des calculs de toute longueur, à cause des circuits stationnaires. C'est pourquoi on s'intéressera à M_w^* .

Exemple 15

Les matrices de l'automate A_1 de la Figure 4.5 sont :

$$I = \left[egin{array}{cccc} 1 & 0 & 0 & 0 \end{array}
ight] \quad T \left[egin{array}{c} 0 \ 0 \ 0 \ 1 \end{array}
ight]$$

On rappelle la définition de l'étoile d'une matrice :

Définition 4.2. Soit M une matrice carrée, à coefficients dans \mathbb{K} . L'étoile de la matrice M est la somme infinie de toutes les puissances de M. Si elle est définie, elle vérifie l'équation $M^* = \operatorname{Id} + M.M^*$.

Si les quatres sous-matrices de l'équation suivante sont définies, alors l'étoile de M est définie, et (cf. [Conway, 1971]):

$$M^* = \left[\begin{array}{cc} A & B \\ C & D \end{array} \right]^* = \left[\begin{array}{cc} (A^*BD^*C)^*A^* & A^*(BD^*CA^*)^*BD^* \\ D^*C(A^*BD^*C)^*A^* & D^* + D^*CA^*(BD^*CA^*)^*BD^* \end{array} \right],$$

avec A et D deux matrices carrées.

Soit maintenant $w = w_1 \dots w_k$ un mot de A^* de longueur k. On définit, pour étudier les calculs de w, des matrices par blocs de dimension $(k+2) \times (k+2)$, dont chacune des sous-matrices est de taille $n \times n$, et l'on indicera les blocs avec des entiers dans [0, k+1].

Soit donc maintenant $M_w,$ la matrice représentant les chemins de longueur 1 sur w :

$$M_{w} = \begin{bmatrix} 0 & F(\vdash) & 0 & \dots & 0 \\ \hline B(w_{1}) & 0 & F(w_{1}) & \dots & \vdots \\ \hline 0 & B(w_{2}) & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 & F(w_{k}) \\ \hline 0 & \dots & 0 & B(\dashv) & 0 \end{bmatrix}.$$
(4.9)

Ainsi, en notant $M_{i,j}$ la matrice de taille $n \times n$ à la i-ème ligne des blocs et à la j-ème colonne, on a pour tout i, $(M_w)_{i,i+1} = F(w_i)$ et $(M_w)_{i,i-1} = B(w_i)$. En particulier, $((M_w)_{i,i+1})_{p,q}$ représente une transition forward de p à q qui peut être empruntée à la position i.

On veut montrer maintenant que la matrice $(M_w^*)_{i,j}$ représente tous les sous-chemins des calculs sur w commençant à la position i et finissant à la position j, en fonction de leurs états de départ et d'arrivée.

Pour cela, on note $C^{(r)}$, avec r dans [0, k+1], la matrice par blocs, de dimension $(r+1) \times (r+1)$, dont les coefficients $(C^{(r)})_{i,j}$ sont des matrices $n \times n$ telles que pour tout $i, j \in [0, k+1]$, $((C^{(r)})_{i,j})_{(p,q)}$ est la somme des poids des chemins compatibles avec w qui vont de la position i et de l'état p à la position j et l'état q, sans passer par une position plus grande (plus à droite) que r. La forme générale de ces chemins est illustrée en Figure 4.6.

Afin de simplifier l'écriture des matrices par blocs, on notera L_i la matrice ligne de r+1 blocs de taille $n\times n$, tous nuls sauf le ième, égal à la matrice identité.

On peut énoncer le lemme suivant :

Lemme 9. $M_w^* = C^{(k+1)}$.

Démonstration. La preuve se fait par induction, avec la restriction de l'étoile de M_w à ses r+1 premières lignes et colonnes. Notons $M^{(r)}$ cette restriction.

On a donc $M^{(k+1)} = M_w$, et l'on va prouver par induction sur r que pour tout r de [0, k+1], $(M^{(r)})^* = C^{(r)}$.

On a tout d'abord $(M^{(0)})^* = 0^* = \mathsf{Id} = C^{(0)}$.

Soit maintenant r dans [0, k+1] tel que $(M^{(r)})^* = C^{(r)}$.

On a donc:

$$M^{(r+1)} = \begin{bmatrix} M^{(r)} & {}^{t}L_{r+1}.F(w_r) \\ B(w_{r+1}).L_{r+1} & 0 \end{bmatrix}.$$
 (4.10)

Si l'on note $R = M^{(r)}, L = L_{r+1}, F = F(w_r), \text{ et } B = B(w_{r+1}), \text{ on obtient :}$

$$(M^{(r+1)})^* = \begin{bmatrix} (R^{*t}LFBL)^*R^* & R^*(^tLFBLR^*)^{*t}LF \\ BL(R^{*t}LFBL)^*R^* & (BLR^{*t}LF)^* \end{bmatrix}$$
(4.11)

Calculons à présent $C^{(r+1)}$. Ce calcul se fera en quatre parties : considérons tout d'abord $C^{(r+1)}_{i,j}$, avec i et j dans $[\![0,r]\!]$.

Il s'agit des chemins de la position i à la position j, tels qu'aucune position intermédiaire ne soit supérieure à r+1. Ces chemins peuvent être de deux formes. Soit ils n'ont aucune position intermédiaire supérieure à r, soit ils ont atteint la position r+1. Le détail de la décomposition est expliqué dans la Figure 4.6.

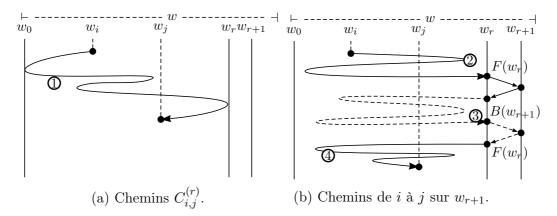


FIGURE 4.6 – Décomposition des chemins de $C_{i,j}^{(r+1)}$, avec i et j dans [0,r].

On a donc, pour tout i, j de $\llbracket 0, r \rrbracket$:

$$C_{i,j}^{(r+1)} = C_{i,j}^{(r)} + C_{i,r}^{(r)} F(w_r) B(w_{r+1}) \left(C_{r,r}^{(r)} F(w_r) B(w_{r+1}) \right)^* C_{r,j}^{(r)}$$
(4.12)

On peut raisonner de la même manière sur chacune des trois autres possibilités pour i, j. Cela nous donne les équations suivantes :

$$\forall i \in [0, r] \ C_{i,r+1}^{(r+1)} = C_{i,r}^{(r)} \left(F(w_r) B(w_{r+1}) C_{r,r}^{(r)} \right)^* F(w_r)$$

$$(4.13)$$

$$\forall j \in [0, r] \ C_{r+1,j}^{(r+1)} = B(w_r) \left(C_{r,r}^{(r)} F(w_r) B(w_{r+1}) \right)^* C_{r,j}^{(r)}$$

$$(4.14)$$

$$C_{r+1,r+1}^{(r+1)} = \left(B(w_{r+1})C_{r,r}^{(r)}F(w_r)\right)^* \tag{4.15}$$

On remarque de plus que pour tout i, j:

- $\bullet \ C_{i,r}^{(r)} = \left(C^{(r)\ t}L\right)_i,$
- $C_{r,j}^{(r)} = (LC^{(r)})_i$, et
- $C_{r,r}^{(r)} = LC^{(r)} {}^{t}L.$

Avec l'hypothèse d'induction, on obtient :

$$\forall i, j \in [0, r] \left(C^{(r+1)} \right)_{i,j} = \left(R^* + R^{*t} LFB \left(LR^{*t} LFB \right)^* LR^* \right)_{i,j} \qquad (4.16)$$
$$= \left(\left(R^{*t} LFBL \right)^* R^* \right)_{i,j}, \qquad (4.17)$$

ce qui correspond bien à la partie voulue dans $(M^{(r+1)})^*$.

Les trois autres équations donnent de même les résultats souhaités. Par conséquent, $(M^{(r+1)})^* = C^{(r+1)}$.

Ainsi, on a une matrice qui nous permet de savoir le poids des chemins compatibles avec w, qui vont de la position i dans l'état p, à la position j dans l'état q, sans dépasser la position k+1.

On peut donc, en appliquant directement le lemme 9, obtenir la proposition suivante :

Proposition 40. Le poids de w dans A est égal à $I.L_2.M^*$. $^tL_1.T$.

On souhaite désormais calculer ce poids. Remarquons que si \mathcal{A} est fortement valide, alors l'étoile de M_w est définie pour tout mot w. On a donc que $\operatorname{Id} - M_w$ est inversible. En effet, si ce n'était pas le cas, il existerait un vecteur non nul v tel que $M_w.v = v$. On aurait donc $M_w^*.v = (\operatorname{Id} + M_w^*.M_w).v = v + M_w^*.v$, et donc v = 0 ce qui est contradictoire.

Ainsi, pour tout mot w, on peut écrire :

$$\langle \mathcal{A}, w \rangle = I.L_2.(\mathsf{Id} - M_w)^{-1}.{}^tL_1.T$$
 (4.18)

$$= \frac{1}{\det(\mathsf{Id} - M_w)} \sum_{p,q \in Q} I_p.(\operatorname{adj}(\mathsf{Id} - M_w))_{n+p,q}.T_q, \tag{4.19}$$

avec adj (X) la transposée de la matrice des cofacteurs de X, et det (X), noté aussi |X|, le déterminant de X.

On peut donc définir les séries suivantes :

- pour tout p, q dans Q, la série $\alpha_{p,q}$ telle que $\langle \alpha_{p,q}, w \rangle = (\operatorname{adj} (\operatorname{Id} M_w))_{n+p,q}$,
- la série δ telle que $\langle \delta, w \rangle = \det(\mathsf{Id} M_w)$.

On a donc

$$\langle \mathcal{A}, w \rangle = \frac{1}{\langle \delta, w \rangle} \sum_{p,q \in Q} I_p \langle \alpha_{p,q}, w \rangle T_q,$$
 (4.20)

Ainsi, si nous arrivons à montrer que les séries $\alpha_{p,q}$ et δ sont rationnelles, nous obtiendrons que toute série réalisable par un automate boustrophédon peut s'écrire comme le quotient de Hadamard de deux séries rationnelles.

Le but de la partie suivante est de construire un automate unidirectionnel réalisant chacune de ces séries, ce qui nous donnera le résultat.

Du calcul d'un déterminant au quotient de Hadamard de deux séries rationnelles

Les séries que nous souhaitons montrer rationnelles sont toutes des déterminants de matrices par blocs de forme particulière : la diagonale est l'identité, et à l'exception des blocs sur la sur-diagonale et la sous-diagonale, tous les blocs sont vides : c'est une matrice bloc-tridiagonale. Nous allons dans cette partie montrer que l'on peut calculer récursivement le déterminant d'une matrice tridiagonale par blocs. Le calcul inductif du déterminant de telles matrices a déja été étudié, voir [Molinari, 2008] par exemple, mais nous construisons ici un automate unidirectionnel le calculant.

Déterminant d'une matrice tridiagonale par blocs

Soit n et k deux entiers positifs. On considère deux familles de matrices carrées à coefficients dans $\mathbb{K}: (A_i)_{i \in [\![1,k]\!]}$ et $(A_i')_{i \in [\![0,k]\!]}$.

Pour tout r de [0, k], on considère la matrice $N^{(r)}$ de $\mathbb{K}^{rn \times (r+1)n}$ suivante :

$$N^{(r)} = \begin{bmatrix} A'_r & \mathsf{Id}_n & A_r & & 0\\ 0 & A'_{r-1} & \mathsf{Id}_n & A_{r-1} & & \\ & & \ddots & \ddots & \ddots & \\ & & & A'_1 & \mathsf{Id}_n & A_1\\ 0 & & & & A'_0 & \mathsf{Id}_n \end{bmatrix}, \tag{4.21}$$

avec Id_n la matrice identité de taille n.

On cherchera dans la suite à calculer le déterminant de cette matrice privée de n colonnes. Pour cela, nous introduisons quelques notations :

- Pour tout ensemble x, pour tout entier i, on note \mathcal{P}_iX l'ensemble des parties de X à i élements.
- On note \overline{X} le complémentaire d'un ensemble d'indices X, *i.e.* l'ensemble des indices n'appartenant pas à X, et ΣX la somme des élements de X.
- On note $M_{X\times Y}$ la restriction de M, une matrice, aux lignes de X et aux colonnes de Y, avec X et Y deux ensembles d'indices.
- Pour tout ensemble C de $\mathcal{P}_n[[1,2n]]$, on note $G^{(r)}(C)$ la matrice carrée $N_{[1,(r+1)n]]\times\overline{C}}^{(r)}$: c'est $N^{(r)}$ privée de n colonnes, celles de C.
- Pour tout C, D de $\mathcal{P}_n[[1, 2n]]$, pour tout a, b de A, on définit

$$K^{(r)}(C,D) = \begin{bmatrix} A'_r & \mathsf{Id}_n & A_r & 0\\ 0 & A'_{r-1} & \mathsf{Id}_n & A_{r-1} \end{bmatrix}_{\llbracket 1,2n \rrbracket \times (\overline{C} \cap \llbracket 1,2n \rrbracket) \cup (D+2n)}. \tag{4.22}$$

Il s'agit d'une matrice carrée de taille 2n: dans les deux premières colonnes de taille n, on enlève les colonnes de C, et dans les deux dernières, on garde celles de D.

Le lemme suivant nous dit comment calculer récursivement le déterminant de $N^{(r)}$ privée de n colonnes.

Lemme 10. Pour tout C de $\mathcal{P}_n[[1,2n]]$, pour tout r de [[2,k]], on a:

$$\left| G^{(r)}(C) \right| = \sum_{D \in \mathcal{P}_n[[1,2n]]} (-1)^{\operatorname{sig}(D)} \cdot \left| K^{(r)}(C,D) \right| \cdot \left| G^{(r-2)}(D) \right| \tag{4.23}$$

$$avec \ \mathsf{sig}(D) = \ \frac{n(n+1)}{2} + \Sigma D \ pour \ tout \ D \ de \ \mathcal{P}_n[\![1,2n]\!].$$

 $D\acute{e}monstration$. On pose d un entier positif, N une matrice de $\mathbb{K}^{d\times d}$, et X un sous-ensemble de [1,d]. L'idée est d'utiliser la formule de Laplace suivante, sur le calcul inductif du déterminant :

$$|N| = \sum_{Y \in \mathcal{P}_{|X|}[1,d]} (-1)^{\Sigma X + \Sigma Y} |N_{X \times Y}| \cdot |N_{\overline{X} \times \overline{Y}}|.$$
 (4.24)

On applique cette formule à $G^{(k)}(C)$, en prenant X = [1, 2n]. Si Y ne contient pas [1, n], alors il existe une colonne nulle dans $N_{\overline{X} \times \overline{Y}}$: son déterminant est nul. De plus, si Y contient un élément plus grand que 3n, alors il existe une colonne nulle dans $N_{X \times Y}$: son déterminant est nul.

Ainsi, avec $D = Y \setminus [1, n]$, on peut réécrire :

$$\left| G^{(r)}(C) \right| = \sum_{D \in \mathcal{P}_n \llbracket n+1, 3n \rrbracket} (-1)^{\mathsf{sig_0}(D)} \cdot \left| K^{(r)}(C, D-n) \right| \cdot \left| H^{(r)}(D) \right|, \qquad (4.25)$$

avec

$$\operatorname{sig}_{0}(D) = n(2n+1) + \frac{n(n+1)}{2} + \Sigma D \text{ et}$$
 (4.26)

$$H^{(r)}(D) = G^{(r)}(C)_{[2n+1,kn] \times ([n+1,kn] \setminus D)}.$$
(4.27)

Posons maintenant D', dans $\mathcal{P}_n[[1,2n]]$ tel que D'+n=D. On obtient alors :

$$sig_0(D) = sig(D') = \frac{n(n+1)}{2} + \Sigma D',$$
 (4.28)

$$K^{(r)}(C, D-n) = K^{(r)}(C, D'), \tag{4.29}$$

$$H^{(r)}(D) = G^{(r-2)}(D').$$
 (4.30)

Cela nous donne bien le résultat du lemme. On peut noter que sig(D) ne dépend pas de r.

Ce lemme nous permet de calculer le déterminant de $(\operatorname{Id} - M_w)$ ainsi que sa matrice des cofacteurs, en choisissant judicieusement les blocs de la matrice $N^{(r)}$. C'est le but de la partie suivante : décrire un automate réalisant la série qui à tout mot w associe $\det(\operatorname{Id} - M_w)$.

L'automate unidirectionnel calculant le déterminant

Avant de pouvoir décrire l'automate, on introduit quelques notations qui nous seront utiles pour faire le lien entre l'automate et la matrice $N^{(r)}$ décrite ci-dessus.

On note ${}^{\circ}M$ la rotation de la matrice M par un demi-tour. Pour tout $i, j \in [1, n]$, $({}^{\circ}M)_{i,j} = M_{n-i+1, n-j+1}$. Notez que $|X| = |{}^{\circ}X|$.

On modifie également les notations ci-dessus :

• On pose, avec u un mot de A^* de longueur k,

$$N^{(u)} = \begin{bmatrix} -^{\circ}F(u_k) & \operatorname{Id}_n & -^{\circ}B(u_k) \\ & \ddots & \ddots & \ddots \\ & & -^{\circ}F(u_1) & \operatorname{Id}_n & -^{\circ}B(u_1) \\ 0 & & -^{\circ}F(\vdash) & \operatorname{Id}_n \end{bmatrix};$$

- pour tout ensemble C de $\mathcal{P}_n[1, 2n]$, on pose $G^{(u)}(C)$ la matrice carrée $N^{(u)}_{[1,(k+1)n]\times\overline{C}}$: c'est $N^{(u)}$ privée de n colonnes, celles de C;
- pour tout C, D de $\mathcal{P}_n[1, 2n]$, pour tout a, b de A, on définit

$$K^{(a,b)}(C,D) = \left[\begin{array}{ccc} -{}^{\circlearrowright}F(a) & \operatorname{Id}_n & -{}^{\circlearrowright}B(a) & 0 \\ 0 & -{}^{\circlearrowright}F(b) & \operatorname{Id}_n & -{}^{\circlearrowright}B(b) \end{array} \right]_{\mathbb{I}1,2n\mathbb{I}\times\overline{C}\cup(D+2n)}.$$

On rapelle également que $F(\dashv) = B(\vdash) = 0$. Ainsi, le lemme 10 s'écrit :

$$\left| G^{(uab)}(C) \right| = \sum_{D \in \mathcal{P}_n [\![1, 2n]\!]} (-1)^{\operatorname{sig}(D)} \cdot \left| K^{(b,a)}(C, D) \right| \cdot \left| G^{(u)}(D) \right|. \tag{4.31}$$

De plus, on a que pour tout mot w = u.a,

$$|\operatorname{Id} - M_w| = |^{\circ}(\operatorname{Id} - M_w)| = \left[egin{array}{cccc} \operatorname{Id}_n & -^{\circ}B(\dashv) & 0 & 0 & \cdots & 0 \\ -^{\circ}F(a) & \operatorname{Id}_n & -^{\circ}B(a) & 0 & \cdots & 0 \\ 0 & & & & & \\ \vdots & & & N^{(u)} & & & \\ 0 & & & & & \end{array} \right]$$

En appliquant la formule de Laplace (équation (4.24)), on obtient, en posant X = [1, 2n], et $Y = [1, n] \cup D$,

$$|\mathsf{Id} - M_w| = \sum_{D \in \mathcal{P}_n[[1,2n]]} (-1)^{sig(D)} |K^{(\dashv,a)}([[1,n]],D)| \cdot |G^{(u)}(D)|.$$
 (4.32)

On construit dorénavant un automate qui va calculer ce déterminant. L'idée est de montrer que l'on peut calculer $G^{(u)}(D)$ dans les états de l'automate après avoir lu u. Le poids final de l'automate sera obtenu grâce à l'équation ci-dessus. Comme la formule de récurrence pour calculer $G^{(u)}(D)$ est d'ordre 2, on aura deux types d'états en fonction de la parité du mot.

L'idée de la construction est en Figure 4.7, et l'on note f(a, b, C, D) pour $(-1)^{\operatorname{sig}(D)} |K^{(b,a)}(C,D)|$. Remarquons que dans un état (a,D), on vient de lire une lettre a, de même parité que la dernière lettre.

Plus précisément, l'automate $\mathcal{A} = (Q, E, I, T)$, dont on montrera plus tard qu'il calcule bien le déterminant, est défini comme suit :

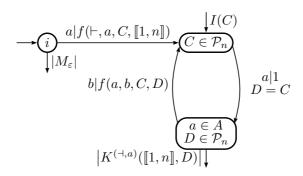


FIGURE 4.7 – Schéma de l'automate calculant δ .

- L'ensemble des états est $\{i\} \cup \mathcal{P}_n[1, 2n] \cup A \times \mathcal{P}_n[1, 2n]$.
- L'état i est initial de poids 1, et final de poids $|M_{\varepsilon}|$.
- Pour tout C dans $\mathcal{P}_n[[1,2n]]$, C est initial de poids $\Big| [-{}^{\circ}F(\vdash) \ \mathsf{Id}_n]_{[[1,n]] \times \overline{C}} \Big|$.
- Tous les états de la forme (a, D) sont finaux, de poids $|K^{(\dashv,a)}([1, n], D)|$.
- Pour tout C de $\mathcal{P}_n[[1,2n]]$, pour tout $a \in A$, il y a une transition de i à C étiquetée par a, de poids $|K^{(a,\vdash)}(C,[1,n])|$.
- Pour tout C de $\mathcal{P}_n[[1, 2n]]$, pour tout $a \in A$, il y a une transition de C à (a, C) étiquetée par a, de poids 1.
- Pour tout C, D de $\mathcal{P}_n[[1, 2n]]$, pour tout $a, b \in A$, il y a une transition de (a, D) à C étiquetée par b, de poids $(-1)^{\operatorname{sig}(D)} |K^{(b,a)}(C, D)|$.

Montrons tout d'abord la propriété suivante :

Lemme 11. Pour tout mot $u \in A^*$, pour tout $C \in \mathcal{P}_n[[1, 2n]]$, la somme des poids des chemins de \mathcal{A} sur u commençant en un état initial (dont on compte le poids) et finissant en C est de $|G^{(u)}(C)|$.

Démonstration. Cette preuve se fait par récurrence sur la taille de u, avec deux cas, un cas pair, et un cas impair, car la récurrence du lemme 10 est d'ordre 2. Cependant la seule différence est à l'initialisation. En effet, pour les mots de longueur paire, les chemins concernés commencent en i et vont ensuite dans un état C, après avoir lu une première lettre. Les chemins sur les mots de longueur impaire, eux, commencent directement en un état C. Mais dans les deux cas, la propagation de la propriété à un mot u.a.b si elle est vraie pour le mot u se fait de la même manière. Ainsi, il nous faut donc prouver trois choses :

- le poids initial d'un état C est bien de poids $|G^{(\varepsilon)}(C)|$,
- les chemins de i à C d'étiquette a sont bien de poids $|G^{(a)}(C)|$, et

• si la propriété est vraie pour un mot u, alors elle est vraie pour u.a.b, pour toutes lettres a, b de A.

Montrons tout d'abord que le poids initial d'un état C est bien de poids $|G^{(\varepsilon)}(C)|$.

On a le poids initial de C qui vaut

$$\left| \left[-{}^{\circ}F(\vdash) \quad \mathsf{Id}_{n} \right]_{\llbracket 1,n \rrbracket \times \overline{C}} \right| \tag{4.33}$$

Or $G^{(\varepsilon)}(C)$ est la matrice

$$\left[-{}^{\circ}F(\vdash) \quad \mathsf{Id}_n \right]_{\mathbb{I}_{1,n}\mathbb{I}\times\overline{C}} \tag{4.34}$$

Son déterminant est bien égal au poids initial de l'état C.

Montrons maintenant que les chemins de i à C d'étiquette a sont bien de poids $|G^{(a)}(C)|$. Ils n'ont qu'une seule transition, et leur poids initial est de 1. Pour chaque état C, le chemin de i à C d'étiquette a est de poids $|K^{(a,\vdash)}(C, \llbracket 1, n \rrbracket)|$ c'est-à-dire :

$$\begin{bmatrix} -{}^{\circ}F(a) & \mathsf{Id}_n & -{}^{\circ}B(a) \\ 0 & -{}^{\circ}F(\vdash) & \mathsf{Id}_n \end{bmatrix}_{\llbracket 1,2n \rrbracket \times \overline{C}} . \tag{4.35}$$

Or $G^{(a)}(C)$ est la matrice

$$\begin{bmatrix} -^{\circ}F(a) & \mathsf{Id}_{n} & -^{\circ}B(a) \\ 0 & -^{\circ}F(\vdash) & \mathsf{Id}_{n} \end{bmatrix}_{\llbracket 1,2n \rrbracket \times \overline{C}}.$$
 (4.36)

Son déterminant est donc bien égal au poids des chemins de i à C d'étiquette a.

Supposons maintenant que la somme des poids des chemins de \mathcal{A} sur u commençant en un état initial (dont on compte le poids) et finissant en C est de $|G^{(u)}(C)|$. On cherche donc la valeur de la somme des poids des chemins de \mathcal{A} sur u.a.b commençant en un état initial et finissant en C. Ces chemins sont des chemins sur u commençant en un état initial et finissant en un état D, suivis de deux transitions, une étiquetée par a et allant de D à (a, D), et une de (a, D) à C étiquetée par b. Comme par l'hypothèse de récurrence on connait le poids de ces chemins en D, on peut écrire que le poids des chemins en C est de :

$$\sum_{D \in \mathcal{P}_n[[1,2n]]} |G^{(u)}(D)| .E(D,a,(a,D)).E((a,D),b,C).$$
 (4.37)

Or
$$E(D, a, (a, D)) = 1$$
 et $E((a, C), b, D) = (-1)^{\mathsf{sig}(D)} |K^{(b, a)}(D, C)|$.

Ainsi, le poids en C après avoir lu u.a.b est de :

$$\sum_{D \in \mathcal{P}_n[\![1,2n]\!]} (-1)^{\mathsf{sig}(D)} \cdot \left| K^{(b,a)}(C,D) \right| \cdot \left| G^{(u)}(D) \right|, \tag{4.38}$$

qui est bien égal à $|G^{(uab)}(D)|$.

Ainsi, on a bien prouvé par récurrence notre propriété.

On peut maintenant poser:

Lemme 12. Pour tout mot w de A^+ , $\langle A, w \rangle = |\mathsf{Id} - M_w|$.

Démonstration. Le poids d'un mot w.a dans \mathcal{A} est la somme des poids des calculs, i.e. la somme des poids des chemins allant d'un état initial à un état final. Les seuls chemins acceptants sont les chemins sur w commençant en un état initial et finissant en C, suivis d'une transition de C à (a,C) étiquetée par a, avec C dans $\mathcal{P}_n[\![1,2n]\!]$. Comme la somme des poids des chemins sur w finissant en C est de $G^{(w)}(C)$, on a que $\langle \mathcal{A}, w.a \rangle = \sum_{C \in \mathcal{P}_n[\![1,2n]\!]} G^{(w)}(C).E(C,a,(a,C).T(a,C))$.

On peut le réécrire comme

$$\langle \mathcal{A}, w.a \rangle = \sum_{C \in \mathcal{P}_n [\![1, 2n]\!]} G^{(u)}(C).(-1)^{sig(C)}. \left| K^{(\dashv, a)}([\![1, n]\!], D) \right|).$$
 (4.39)

Or l'équation (4.32) nous dit que cette somme est égale à $|Id - M_w|$.

Ainsi, par le lemme 12, la série δ telle que $\langle \delta, w \rangle = |\mathsf{Id} - M_w|$ est une série rationnelle.

On cherche maintenant à montrer que pour tout p, q dans Q, la série $\alpha_{p,q}$ telle que $\langle \alpha_{p,q}, w \rangle = (\text{adj}(\text{Id} - M_w))_{n+p,q}$ est rationnelle.

Or, on peut poser $M_w^{(q,n+p)}$ une matrice telle que $(\operatorname{adj}(\operatorname{Id}-M_w))_{n+p,q} = |M_w^{(q,n+p)}|$.

C'est la matrice $\operatorname{Id} - M_w$ dans laquelle certains blocs ont été changés. On remplace la ligne n + p et la colonne q par une ligne et une colonne de 0, sauf à leur intersection, que l'on pose égale à 1.

On pose, avec w = u un mot de longueur k:

- $\mathsf{Id}_n^{(p)}$ la matrice identité de taille n dans laquelle le 1 à la p-ième ligne et p-ième colonne est remplacé par un 0,
- $B^{(p,q)}(a)$ la matrice B(a) dans laquelle la p-ième ligne et p-ième colonne son nulles sauf pour $B^{(p,q)}(a)_{p,q}$ qui vaut 1,
- $F^{(p)}(a)$ la matrice F(a) dans laquelle la p-ième ligne est remplacée par une ligne nulle, et

$$\bullet \ N'^{(u)} = \left[\begin{array}{cccc} B(u_2) & \operatorname{Id}_n & F(u_k) \\ & \ddots & \ddots & \ddots \\ & & B(u_k) & \operatorname{Id}_n & F(u_k) \\ 0 & & B(\dashv) & \operatorname{Id}_n \end{array} \right].$$

On a donc

$$M_w^{(q,n+p)} = \begin{bmatrix} \operatorname{Id}_n^{(q)} & F(\vdash) & 0 & 0 & \cdots & 0 \\ B^{(p,q)}(w_1) & \operatorname{Id}_n^{(p)} & F^{(p)}(w_1) & 0 & \cdots & 0 \\ 0 & & & & & \\ \vdots & & & N'^{(u)} & & & \\ 0 & & & & & \\ \end{bmatrix}. \tag{4.40}$$

On peut donc appliquer la même procédure qu'au cas précédent : on définit un automate dont les états calculent récursivement le déterminant de $M_w^{(q,n+p)}$. Cependant, la première lettre n'a pas les mêmes matrices F et B que toutes ses autres occurrences dans le mot. En effet, on leur a supprimé une ligne et/ou une colonne.

On va donc ajouter à l'automate deux ensembles d'états : des états de $\mathcal{P}_n[1, 2n]$, et des états de $A \times \mathcal{P}_n[1, 2n]$, annotés pour ne pas les confondre avec les états originaux. L'idée de cette construction est illustrée dans la Figure 4.8. On note dans ce schéma f(a, b, C, D) comme auparavant, $f_{p,q}(a, b, C, D)$ pour

On note dans ce schéma
$$f(a,b,C,D)$$
 comme auparavant, $f_{p,q}(a,b,C,D)$ pour $(-1)^{\operatorname{sig}(D)} \begin{bmatrix} -{}^{\circ}F(b) & \operatorname{Id}_n & -{}^{\circ}B(b) & 0 \\ 0 & -{}^{\circ}F^{(p)}(a) & {}^{\circ}\operatorname{Id}_n^{(p)} & -{}^{\circ}B^{(p,q)}(a) \end{bmatrix}_{\llbracket 1,2n \rrbracket \times (\overline{C} \cap \llbracket 1,2n \rrbracket) \cup (D+2n)}$, et $I'(C) = \begin{bmatrix} -{}^{\circ}F(\vdash) & \operatorname{Id}_n^{(q)} \end{bmatrix}_{\llbracket 1,n \rrbracket \times \overline{C}}$.

Avec les poids et transitions identiques à l'automate précédent, sauf en ce qui concerne les deux premières lignes de la matrice, et en traitant à part les mots de longueur 1, on peut montrer que cet automate calcule bien le déterminant de $M_w^{(q,n+p)}$. Pour tout p,q dans Q, la série $\alpha_{p,q}$ est donc rationnelle.

On peut donc énoncer notre théorème :

Théorème (Rappel du théorème 4.1). Soit \mathbb{K} un corps commutatif. Les séries sur \mathbb{K} réalisables par un automate boustrophédon fortement valide sont des quotients de Hadamard de séries rationnelles.

Or on a vu dans le chapitre précédent (proposition 28) que sur \mathbb{Q} , \mathbb{R} et \mathbb{C} , les séries de Hadamard et les quotients de Hadamard de séries rationnelles coïncidaient. Comme sur ces semi-anneaux, les séries de Hadamard sont réalisables par des automates circulaires (proposition 36), on peut énoncer également :

Théorème 4.3. Soit \mathbb{K} égal à \mathbb{Q} , \mathbb{R} ou \mathbb{C} . Les séries sur \mathbb{K} réalisées par les \mathbb{K} -automates boustrophédons fortement valides sont réalisables par des automates circulaires sur \mathbb{K} .

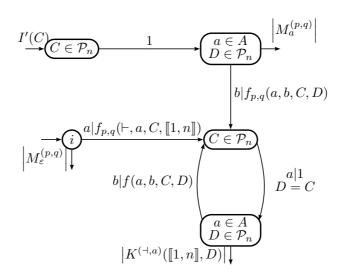


FIGURE 4.8 – Schéma de l'automate réalisant $\alpha_{p,q}$.

z-représentation sur \mathbb{Q} , \mathbb{R} , ou \mathbb{C}

On souhaite ici montrer que sur $\mathbb{K} = \mathbb{Q}$, \mathbb{R} ou \mathbb{C} , toute série s réalisable par un automate boustrophédon est z-reconnaissable, c'est-à-dire qu'il existe un produit z associatif sur des \mathbb{K} -matrices, et on peut définir deux fonctions de projection \mathcal{I} et \mathcal{T} ainsi qu'un morphisme de monoïdes μ de A^* dans ces matrices, tels que le poids de tout mot w dans s est égal à \mathcal{I} z $\mu(w)$ z \mathcal{T} . Pour toute cette partie, on pose $\mathcal{A} = (Q_+, Q_-, E, I, T)$ un automate boustrophédon valide.

Pour tout mot u de A^* , on peut définir une matrice, $\overline{\mu(u)}$, représentant les chemins sur u de l'automate qui commencent à gauche et finissent à droite. Plus précisément, on pose $\overline{\mu(u)}_{p,q} = k$, avec $p,q \in Q_+$, si et seulement si k est la somme des poids des chemins sur u de l'automate, allant de l'état p en position 0 à l'état q en position |u|. C'est donc une matrice de $M^{|Q_+|\times|Q_+|}(\mathbb{K})$. On peut définir de la même manière trois autres matrices, $\mu(u)$ pour les chemins commençant à droite et finissant à gauche, dans $M^{|Q_-|\times|Q_-|}(\mathbb{K})$, $\mu(u)$ pour les chemins commençant à gauche et finissant à gauche, en lisant un préfixe du mot, dans $M^{|Q_+|\times|Q_-|}(\mathbb{K})$, et $\mu(u)$ pour les chemins commençant à droite et finissant à droite, en lisant un suffixe du mot, dans $M^{|Q_-|\times|Q_+|}(\mathbb{K})$.

Pour tout mot $u \in A^*$, on notera

$$\mu(u) = \left[\begin{array}{cc} \overrightarrow{\mu(u)} & \overleftarrow{\mu(u)} \\ \overrightarrow{\mu(u)} & \overleftarrow{\mu(u)} \end{array} \right].$$

^{1.} On retourne dans le modèle d'automates boustrophédons de la définition 2.9

On peut remarquer que pour toute lettre a, on a :

- $\overrightarrow{\mu(a)}_{p,q} = E(p, a, q) \text{ avec } p, q \in Q_+,$
- $\mu(a)_{p,q} = E(p, a, q)$ avec $p \in Q_+, q \in Q_-,$
- $\overrightarrow{\mu(a)}_{p,q} = E(p, a, q)$ avec $p \in Q_-, q \in Q_+,$
- $\mu(a)_{p,q} = E(p, a, q)$ avec $p, q \in Q_-$.

La matrice $\mu(a)$ est la matrice d'adjacence du graphe sous-jacent à l'automate restreint à ses transitions étiquetées par a.

On pose
$$\mathcal{I} = \begin{bmatrix} I \\ \overrightarrow{\mu}(\vdash) \end{bmatrix}$$
 et $\mathcal{T} = \begin{bmatrix} T \middle \overrightarrow{\mu}(\dashv) \end{bmatrix}$.

On considère sur ces matrices le produit définit par les équations (4.1) à (4.4). Attention, comme ce produit est défini à partir d'étoiles, il n'est pas toujours défini sur \mathbb{C} .

Lemme 13. Soit \mathcal{A} un automate boustrophédon, et μ l'application définie cidessus. Si \mathcal{A} est fortement valide, alors pour tout mots u et v, les matrices $\mu(u)$ et $\mu(v)$ sont définies, et le produit $\mu(u) \ge \mu(v)$ est défini et vaut $\mu(uv)$.

Démonstration. La preuve est par induction sur la longueur des mots u et v. Elle se fait indépendamment pour chacune des quatre matrices $\mu(u) \ge \mu(v)$, $\mu(u) \ge \mu(v)$ et $\mu(u) \ge \mu(v)$, mais dans un souci de concision, seule la preuve pour $\mu(u) \ge \mu(v)$ est faite ici, les preuves supplémentaires suivant le même raisonnement.

On a
$$\overline{\mu(u)} \circ \mu(v) = \overline{\mu(u)} \left(\overrightarrow{\mu(v)} \overrightarrow{\mu(u)} \right)^* \overline{\mu(v)}$$
. Par définition des matrices $\overrightarrow{\mu(v)}$

et $\mu(u)$, $\mu(v)\mu(u)$ est la somme des poids des chemins sur le mot uv commençant et finissant en position |u| dans des états forward, et ne passant qu'une seule autre fois par la position |u|. De manière plus générale, $\left(\stackrel{\longleftarrow}{\mu(v)} \stackrel{\longleftarrow}{\mu(u)} \right)^k$ est la somme des poids des chemins sur le mot uv commençant et finissant en position |u| dans des états forward, et passant par la position |u| exactement k fois.

Ainsi, la somme des poids des chemins sur le mot uv commençant et finissant en position |u| dans des états forward est la somme de toutes les puissances de $\mu(v)\mu(u)$, ce qui s'obtient en en prenant l'étoile.

Comme l'automate est fortement valide, cette étoile est bien définie. Or un chemin sur le mot uv commençant en position 0 et finissant en position |uv| est obtenu en concaténant un chemin sur u de 0 à |u|, puis un chemin sur uv

allant de |u| à |u|, puis un chemin sur v allant de la position (relative) 0 à la position |v|.

Ainsi,
$$\overrightarrow{\mu(uv)} = \overrightarrow{\mu(u)} \left(\overrightarrow{\mu(v)} \overrightarrow{\mu(u)} \right)^* \overrightarrow{\mu(v)} = \overrightarrow{\mu(u)} \mathbf{z} \, \mu(v).$$

Définition 4.3. Soit $m, n \in \mathbb{N}$. On appelle z-représentation de la série s un triplet $(\mathcal{I}, \mu, \mathcal{T})$ avec :

- \mathcal{I} , composé d'un vecteur de $\mathbb{K}^{1\times m}$ et d'une matrice de $\mathbb{K}^{n\times m}$,
- μ un morphisme de $(A^*, .)$ dans $(\mathbb{K}^{(m+n)\times(m+n)}, \mathbf{z})$,
- \mathcal{T} , composé d'un vecteur de $\mathbb{K}^{m\times 1}$ et d'une matrice de $\mathbb{K}^{m\times n}$,

qui vérifie la propriété suivante :

$$\forall w \in A^*, \langle s, w \rangle = \mathcal{I} \operatorname{r} \mu(w) \operatorname{r} \mathcal{T}.$$

Si une série admet une z-représentation, on dit qu'elle est z-reconnaissable.

Avant de s'intéresser au cas des corps, on rappelle ce qui ce passe dans le cas des semi-anneaux rationnellement additifs.

Théorème 4.4 ([Lombardy, 2016]). Si \mathbb{K} est rationellement additif, une série est réalisable par un automate boustrophédon si et seulement si elle est \mathfrak{z} -reconnaissable.

D'une manière plus générale, si le semi-anneau vérifie les axiomes **Ax.1** à **Ax.6**, alors on peut énoncer la proposition suivante :

Proposition 41. Soit s une série réalisée par un automate fortement valide. La série s est z-reconnaissable.

Démonstration. Soit $\mathcal{A} = (Q_+, Q_-, E, I, T)$ l'automate fortement valide réalisant s. On identifie I (resp. T) au vecteur ligne (resp. colonne) qui le caractérise : $I_p = I(p)$. L'application μ de la définition 4.3 est bien un morphisme.

Vérifions que $\langle s, w \rangle = \mathcal{I}\mu(w)\mathcal{T}$.

Soit S_w la somme des poids des chemins sur $\vdash w \dashv$ commençant en position 1 et finissant en position |w|+1. En séparant selon s'ils effectuent des demitours sur \vdash , et en notant, dans un souci de concision, $S_1 = \overrightarrow{\mu(w)} \left(\overrightarrow{\mu(\dashv)} \overrightarrow{\mu(w)} \right)^*$, la contribution des chemins sans effectuer de tels demi-tours, on a :

$$S_w = S_1 + \left(\left(\overrightarrow{\mu(w)} + S_1 \overrightarrow{\mu(\dashv)} \overrightarrow{\mu(w)} \right) \overrightarrow{\mu(\vdash)} \right)^+ S_1. \tag{4.41}$$

En notant
$$S_2 = \left(\overrightarrow{\mu(w)} + S_1 \overrightarrow{\mu(\dashv)} \overrightarrow{\mu(w)} \right) \overrightarrow{\mu(\vdash)}$$
, on a $S_w = S_1 + S_2 S_2^* S_1$.

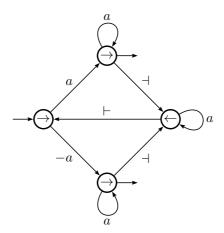


FIGURE 4.9 – Automate A_2 obtenu à partir du triplet $(\mathcal{I}_2, \mu_2, \mathcal{T}_2)$.

Comme $\langle s, w \rangle$ est la somme des poids des chemins sur $\vdash w \dashv$ commençant en position 1 dans un état initial et finissant en position |w| + 1 dans un état final, on a $\langle s, w \rangle = I(S_2)^* S_1 T$, qui est bien égal à $\mathcal{I} \mathsf{z} S_w \mathsf{z} \mathcal{T}$.

Ainsi, $(\mathcal{I}, \mu, \mathcal{T})$ est bien une z-représentation de s.

En particulier, grâce au théorème 4.2, on a :

Théorème 4.5. Si \mathbb{K} est un corps muni d'une norme, toute série sur \mathbb{K} réalisable par un automate boustrophédon valide est z-reconnaissable.

Démonstration. Corollaire immédiat du théorème 4.2 et de la proposition 41.

Remarque 22. La réciproque de ce théorème n'est pas aussi immédiate. En effet, l'automate qui correspond naturellement à une z-représentation n'est pas nécessairement valide. Considérons la z-représentation $(\mathcal{I}_2, \mu_2, \mathcal{T}_2)$ suivante :

$$\mathcal{I}_{2} = \begin{bmatrix} 1 & 0 & 0 \\ \hline 1 & 0 & 0 \end{bmatrix}, \quad \mathcal{T}_{2} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mu_{2}(a) = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.42)$$

On peut vérifier que les produits suivants sont définis :

- $\mathcal{I} \mathbf{z} \mathcal{T} = 0$,
- $\mu_2(a) \mathbf{z} \mu_2(a) = \mu_2(a)$, et
- $\mathcal{I}_2 \, \mathsf{z} \, \mu_2(a) \, \mathsf{z} \, \mathcal{T}_2 = 0.$

Ainsi, $(\mathcal{I}_2, \mu_2, \mathcal{T}_2)$ est une z-représentation de la série nulle. Cependant si l'on considère l'automate \mathcal{A}_2 de la Figure 4.9, on constate que pour tout mot non vide, il existe à la fois une infinité de calculs de poids 1, et une infinité de calculs de poids -1. Cet automate n'est donc pas valide.

Ainsi, grâce aux résultats de ce chapitre et du chapitre précédent, nous avons un certains nombre d'objets équivalents pour représenter certaines séries sur \mathbb{Q} , \mathbb{R} , et \mathbb{C} . Plus précisément, on a montré que :

- les séries réalisées par des automates boustrophédons fortement valides étaient z-reconnaissables,
- dans les corps normés, les automates boustrophédons valides étaient équivalents aux automates boustrophédons fortement valides,
- si le corps est commutatif, les automates boustrophédons fortement valides réalisent des quotients de Hadamard de séries rationnelles, et que
- sur \mathbb{Q} , \mathbb{R} , et \mathbb{C} , de tels quotients sont réalisables par des automates circulaires (chapitre précédent).

Chapitre 5

Automates boustrophédons sur les semi-anneaux et les bi-monoïdes localement finis

"Capri, c'est fini"

– Hervé Vilard

Nous nous sommes intéressés dans le chapitre précédent aux automates boustrophédons dans les corps commutatifs, et nous avons vu que ces derniers étaient équivalents aux automates circulaires. Dans ce chapitre, nous allons nous interesser à l'expressivité des automates boustrophédons lorsque la structure des poids est finie.

Le résultat principal de ce chapitre, publié dans [Dando et Lombardy, 2018b], est le théorème 5.1, résultat classique sur les automates booléens ([Rabin et Scott, 1959] et [Shepherdson, 1959]), dans le cas localement fini : les automates boustrophédons et les automates unidirectionnels réalisent les mêmes séries.

Dans la première section, nous présentons les objets qui nous seront utiles pour représenter les calculs des automates boustrophédons : les coupes. Ces objets ont été introduits par [Shepherdson, 1959] pour montrer l'équivalence des automates unidirectionnels et boustrophédons dans le cas booléen. On présente un certain nombre d'extensions et d'utilisations de ces objets dont nous allons avoir besoin dans la suite.

La deuxième section présente l'automate des coupes d'un automate boustrophédon, qui lui est équivalent, ainsi que les façons dont on s'en servira.

La troisième section s'attache à montrer l'équivalence des automates unidirectionnels et boustrophédons lorsque les poids forment un bi-monoïde localement fini, en construisant un automate basé sur les coupes, telles qu'étendues dans la section précédente.

Les coupes

Dans [Shepherdson, 1959], Shepherdson a montré l'équivalence entre automates boustrophédons et unidirectionnels dans le cas booléen grâce aux coupes, en anglais *crossing sequences*.

L'idée est d'analyser les calculs de l'automate boustrophédon, et de considérer l'ensemble des états rencontrés à une position donnée dans le mot. Cet ensemble, nommé coupe, peut comporter plusieurs fois le même état. Cependant, si c'est le cas, cela veut dire que le calcul considéré passe deux fois à la même position dans le mot, et dans le même état. Le chemin entre ces deux positions sera appelé circuit stationnaire. Le calcul privé de ce circuit est encore un calcul sur le même mot : au moins dans le cas booléen, qui ne s'intéresse qu'à l'existence d'au moins un calcul, on peut donc ne considérer que les coupes sans répétition. À l'opposé, un circuit stationnaire peut être répété un nombre arbitraire de fois dans le calcul et engendrer ainsi un nombre infini de calculs. Ceci nous conduit à considérer deux ensembles particuliers de coupes, celles sans répétitions, et celles avec une répétition.

On montre que si un automate n'admet que des coupes sans répétition, ses calculs peuvent être représentés par un automate fini unidirectionnel. Si le semi-anneau des poids est commutatif, on montre en outre que l'on peut calculer le poids d'une transition entre deux coupes afin d'obtenir un automate unidirectionnel équivalent à l'automate boustrophédon.

Si l'automate admet des coupes avec une répétition, on montre que l'on peut décider si l'automate boustrophédon est valide, et le cas échéant, construire un automate unidirectionnel équivalent. Certains cas simples, les bi-monoïdes pour lesquelles toutes les sommes infinies sont définies, ou ceux pour lesquelles aucune n'est définie, sont étudiés. Le cas général nécessite, pour des raisons que l'on verra plus tard, une déterminisation supplémentaire.

Exemple 16

Afin d'illustrer les définitions, on introduit ici l'automate \mathcal{A}_1 qui servira d'exemple tout au long de ce chapitre. Il est composé de deux états forward, p et r, et d'un état backward q. Il est décrit complètement en Figure 5.1a.

Définitions

Définition 5.1. Une coupe d'un automate boustrophédon $\mathcal{A} = (F, B, E, I, T)$ est un mot de $(FB)^*F$. La coupe d'un calcul ρ à la position i est la restriction de ce calcul aux états p_j tels que pos(j) = i. Une coupe est dite simple si aucun état n'y apparaît deux fois.

5. Automates boustrophédons sur les semi-anneaux et les bi-monoïdes localement finis

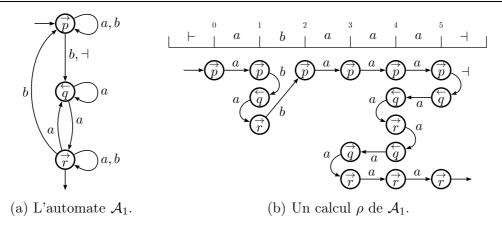


FIGURE 5.1 – L'automate qui servira d'exemple dans ce chapitre.

Exemple 17

Les coupes du calcul ρ de la Figure 5.1b sont : p aux positions 0 et 2, pqr en position 1, 3, et 5, et pqrqr à la position 4. Parmi celles-ci, seules les coupes p et pqr sont simples. Il existe d'autres coupes, certaines n'appartenant pas à ce calcul (r, par exemple), voire n'appartenant à aucun calcul (telle rqp).

Remarque 23. Une coupe simple étant un mot de $(FB)^*F$ sans répétition, tout automate a donc un nombre fini de coupes simples. Plus précisément, en notant f = |F| et b = |B|, les coupes simples sont au nombre de

$$\sum_{i=0}^{\min\{b,f-1\}} \frac{b!}{(b-i)!} \frac{f!}{(f-(i+1))!}.$$
 (5.1)

On peut montrer, en notant u_n cette valeur lorsque f = b + 1 = n, que u_n et n!(n+1)! sont équivalents.

Ainsi, on a:

$$u_n \sim (n+1)(n!)^2$$
 (5.2)

Automates simples

Définition 5.2. Sur un automate boustrophédon, un calcul ρ sur un mot w est dit *simple* si pour chaque position, la coupe de ρ à cette position est simple. Cela signifie que dans ce chemin, aucun état n'apparait deux fois à la même position dans le mot.

Si un calcul ρ sur w n'est pas simple, il passe donc par un état p au moins deux fois, et à la même place dans w. Ainsi, on peut écrire ce calcul $\rho_1\rho_2\rho_3$ avec p l'état séparant ρ_1 de ρ_2 et ρ_2 de ρ_3 . Le chemin ρ_2 est appelé *circuit stationnaire* de ρ . Le chemin $\rho_1\rho_3$ est également un calcul sur le mot w.

П

Exemple 18

Le calcul ρ de la Figure 5.1b n'est pas simple : en effet, il admet en position 4 la coupe pqrqr, dans laquelle apparaît deux fois q. Le circuit (q, a, r)(r, a, q) est un circuit stationnaire.

Remarque 24. S'il existe un calcul sur un mot w qui admet au moins un circuit stationnaire, il existe un autre calcul sur w sans circuit stationnaire. De plus, si l'automate est caractéristique, alors ces deux calculs ont même poids, ce qui n'est pas nécessairement vrai dans le cas général.

Définition 5.3. Un automate est dit *simple* si tout mot de A^* n'est l'étiquette que d'un nombre fini de calculs.

Remarque 25. Tout automate unidirectionnel est simple.

Proposition 42. Tous les calculs d'un automate simple sont des calculs simples.

Démonstration. Immédiat par contraposée.

Dans le cas des automates boustrophédons pondérés sur des semi-anneaux dont l'étoile n'est définie que pour le 0, Anselmo à montré dans [Anselmo, 1990] que seuls les automates simples sont valides et qu'ils sont dans ce cas équivalents aux automates unidirectionnel si le semi-anneau est commutatif.

Extension des coupes

L'idée dans le cas booléen était d'utiliser les coupes simples pour simuler le comportement de l'automate boustrophédon initial : chaque coupe avait un représentant de taille bornée, une coupe simple, et on en avait donc un nombre fini. De plus, le nombre de calculs acceptant un mot donné n'a pas d'importance dès qu'il est non nul.

Dans le cas pondéré, il faut tenir compte de ce nombre, potentiellement infini. Et dans le cas d'un calcul possédant une coupe avec un état apparaissant deux fois, chacun des calculs obtenus par itération du circuit stationnaire a potentiellement un poids différent. Se pose également le problème de la validité.

Pour résoudre ces problèmes, nous travaillerons avec un automate caractéristique, de telle sorte que le poids d'un calcul ne dépendra que de son état final. Ainsi, on pourra représenter une famille infinie de calculs de même poids grâce à une coupe avec répétition d'un état.

On ajoute aux définitions précédentes la définition formelle des coupes dans lesquelles un état peut apparaître deux fois.

Définition 5.4. On appelle *coupes avec répétition* les coupes dans lesquelles aucun état n'apparaît plus de deux fois.

Remarque 26. Les mots de B^* dans lesquels aucune lettre n'apparait plus de deux fois sont, avec b = |B|, au nombre de

$$\sum_{u+d \le b} \binom{b}{u} \binom{b-u}{d} \frac{(u+2d)!}{2^d}.$$
 (5.3)

On peut remarquer, que pour les mots de $(FB)^*F$, en posant f-1=b, la longueur maximale d'une coupe avec répétition est de 4f+1, et donc leur quantité est bornée par f^{4f+1} .

L'automate des coupes

Nous allons maintenant construire un automate à partir de ces coupes. L'idée est de reconstruire les calculs de \mathcal{A} à partir des coupes. Voyons tout d'abord comment l'on peut joindre deux coupes.

Chaînage de coupes

On souhaite reconstituer un calcul à partir d'une suite de coupes. Pour cela, on explicite quelles propriétés doivent vérifier les coupes aux extrémités du mot, ainsi que les façons de "recoller" deux coupes, qu'elles soient simples ou non.

Aux extrémités du mot

Une coupe $q_1
ldots q_{2n+1}$ est dite *initiale* avec poids k^1 si q_1 est initial $(I(q_1) \neq 0_{\mathbb{K}})$, si pour tout i compris entre 1 et n, on a $E(q_{2i}, \vdash, q_{2i+1}) \neq 0_{\mathbb{K}}$, et si $k = I(q_1)\prod_{i=1}^n E(q_{2i}, \vdash, q_{2i+1})$.

Une coupe $q_1 ldots q_{2n+1}$ est dite *finale* avec poids k si q_{2n+1} est final $(T(q_{2n+1}) \neq 0_{\mathbb{K}})$, si pour tout i comprise ntre 0 et n, on a $E(q_{2i+1}, \dashv, q_{2i}) \neq 0_{\mathbb{K}}$, et si $k = (\prod_{i=0}^n E(q_{2i+1}, \dashv, q_{2i})) T(q_{2n+1})$.

Exemple 19

Comme il n'existe pas de transition de demi-tour sur \vdash dans \mathcal{A}_1 , sa seule coupe initiale est p. De plus, l'ensemble des coupes finales est la famille $(pq)^*r$.

^{1.} Le poids d'une coupe a peu de sens dans le monde non-commutatif : considérer les poids entre deux coupes d'un calcul, c'est réorganiser l'ordre des poids de ce calcul. On pourrait toutefois définir les coupes sans considérer les poids.

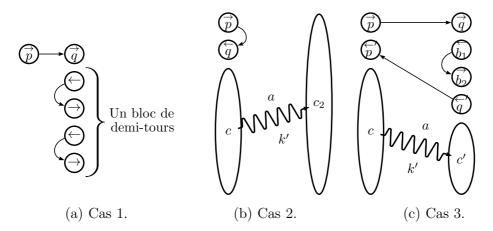


FIGURE 5.2 – Existence d'un maillon entre deux coupes.

Jonction de deux coupes

On s'intéresse également aux propriétés pour joindre deux coupes. Afin de simplifier la description de la relation, on définit les blocs de demi-tours. Il s'agit des suites de transitions de demi-tour consécutives dans une même coupe. Plus précisément, un bloc de demi-tours pour a est un mot w de $(BF)^*$ tel que pour tout i dans [0, |w|/2 - 1], $E(w_{2i}, a, w_{2i+1})$ est non nul. Son poids est $k = \prod_{i=0}^{|w|/2-1} E(q_{2i}, \vdash, q_{2i+1})$.

On ne considère que les blocs de demi-tours constitués de transitions de demi-tours gauches, cela nous suffit pour décrire récursivement deux coupes que l'on souhaite "recoller".

Il existe un maillon de poids k pour la lettre a entre deux coupes c_1 et c_2 , ce que l'on note $c_1 \overset{a}{\underset{k}{\leadsto}} c_2$, si l'une des conditions suivantes est remplie :

Cas 1 : $c_1 = p$, avec p dans F, et c_2 est de la forme q.u, avec u un bloc de demitours pour a de poids k', et q dans F, avec E(p, a, q) non nul tel que k = k'E(p, a, q).

Cas 2 : $c_1 = p.q.c$, avec E(p, a, q) non nul, et c et c_2 chaînables par a avec poids k' tel que k = k'E(p, a, q).

Cas 3 : $c_1 = p.p'.c$, $c_2 = q.b.q'.c'$, avec E(p, a, q) non nul, b un bloc de demi-tours pour a de poids k', E(q', a, p') non nul, et c et c' chaînables par a avec poids k'' tel que k = E(p, a, q)k'E(q', a, p')k''.

On dira qu'il existe un *chaînage* de poids k pour la lettre a entre deux coupes c et c' s'il existe au moins un maillon pour a entre c et c' et que k est la somme des poids de ces maillons.

On peut remarquer que la relation entre deux états de l'automate des

5. Automates boustrophédons sur les semi-anneaux et les bi-monoïdes localement finis

$$(pq, \epsilon)|E(p, a, q) \qquad (\epsilon, qp)|E(q, a, p)$$

$$(p, p')|E(p, a, p') \qquad r$$

$$(q, q')|E(q, a, q')$$

FIGURE 5.3 – Le transducteur \mathcal{C}_a réalisant la relation de chaînage.

coupes est calculable par un transducteur ¹ unidirectionnel. En effet, le transducteur suivant, représenté en Figure 5.3, réalise la relation de chaînage pour la lettre a : les états sont l et r, avec l initial et r final, on note E' la fonction de transition, et pour tout p, p' états de F, et q, q' états de B, on pose :

- $E'(l, l) = (pq, \varepsilon)$ si $E(p, a, q) \neq 0$,
- $E'(l,r) = (p, p') \text{ si } E(p, a, p') \neq 0$,
- E'(r, l) = (q, q') si $E(q, a, q') \neq 0$, et
- $E'(r,r) = (\varepsilon, qp)$ si $E(q, a, p) \neq 0$.

On peut définir ainsi un transducteur \mathcal{C}_a pour chaque lettre a. Deux coupes c et c' sont chaînables par a si et seulement si (c,c') est accepté par \mathcal{C}_a .

Les transitions de r à r reconnaissent les blocs de demi-tours, donc les transitions ramenant à la coupe de départ, celles de l à l capturent les demi-tours de la coupe de gauche, et celles de l à r ou de r à l capturent les transitions qui permettent de changer de coupe.

De plus, si l'automate initial est δ -normalisé, alors ces transducteurs sont déterministes (sur le couple entrée-sortie). En effet, supposons que l'on se trouve dans l'état l, et considérons la première lettre de la coupe de gauche qu'il nous reste à lire. C'est donc un état de l'automate boustrophédon qui va soit uniquement vers des états forward, soit uniquement vers des états backward. S'il ne va que vers des états backward, alors nécessairement, il faut prendre une transition qui fait rester dans l, puisque la coupe commence donc par un demi-tour. Sinon, il faut aller dans l'état r. L'automate est donc non ambigu : entre n'importe quelle paire de coupes, il n'existe qu'un maillon possible pour relier ces deux coupes.

Proposition 43. Soit A un automate boustrophédon. Si A est δ -normalisé, il y a bijection des chemins entre A et son automate des coupes.

^{1.} Automate dont les étiquettes sont des paires de mots : cf. [Eilenberg, 1974] pour la définition originale.

Démonstration. Si \mathcal{A} est δ-normalisé, alors pour toute paire d'états de l'automate des coupes, il n'existe qu'un seul maillon possible pour chaque lettre entre ces deux coupes. Ainsi, tout calcul de l'automate des coupes permet de reconstituer un et un seul calcul de \mathcal{A} .

L'automate des coupes d'un automate boustrophédon simple

Les coupes d'un automate encodent une information suffisante sur les calculs pour pouvoir le simuler avec un automate unidirectionnel dont la taille est le nombre des coupes. Cet automate peut donc être infini. Cependant, si l'automate est simple, alors on verra que cet automate est fini.

Définition 5.5. L'automate des coupes d'un automate boustrophédon $\mathcal{A} = (F, B, E, I, T)$ est défini comme suit : ses états sont les coupes de \mathcal{A} , les poids initiaux sont les poids des coupes initiales, les poids finaux sont les poids des coupes finales, et il existe une transition entre c et c' de poids k étiquetée par a si k est le poids du chaînage pour a entre c' et c'.

On ne considère bien sûr que la partie accessible de cet automate. Cet automate est potentiellement infini.

Proposition 44. Tout automate boustrophédon est équivalent à son automate des coupes.

 $D\'{e}monstration$. Considérons un calcul de \mathcal{A} : il s'agit en fait d'une suite de coupes, et cette suite est un calcul de l'automate des coupes. S'il y a plusieurs maillons possibles pour un chaînage, alors les poids de ces deux calculs peuvent être différents. Mais l'existence de plusieurs maillons est preuve de l'existence d'autre calculs de \mathcal{A} sur le même mot. La somme des poids de tous ces calculs est égale au poids du calcul de l'automate des coupes.

Ainsi, tout calcul de \mathcal{A} est simulé par son automate des coupes. La réciproque est immédiate. \square

Lemme 14. L'automate des coupes d'un automate boustrophédon \mathcal{A} est de taille finie (i.e. il a un nombre fini d'états) si et seulement si toute coupe de \mathcal{A} est simple.

Démonstration. Si toute coupe est simple, par la remarque 23, il y a un nombre fini de coupes, et donc l'automate des coupes est de taille finie.

Si l'automate des coupes est de taille finie, supposons qu'il possède une coupe c qui ne soit pas simple.

Cette coupe est de la forme u.p.v.p.w avec p un état, et u, v, w des suites finies d'états. Comme c est accessible, il existe un calcul de l'automate des coupes passant par cette coupe. On peut donc reconstruire au moins un calcul de \mathcal{A} admettant c à une position i, avec i un entier naturel. Ce calcul admet

donc un circuit stationnaire. On peut écrire ce calcul $\rho_1\rho_2\rho_3$ avec ρ_2 le circuit stationnaire.

Tous les chemins $\rho_1(\rho_2)^i\rho_3$ sont des calculs de \mathcal{A} : toutes les coupes de la forme $u.(p.v)^i.p.w$ appartiennent donc à l'automate des coupes. Ce dernier doit donc être de taille infinie, contradiction.

Proposition 45. Si \mathbb{K} est commutatif, tout automate boustrophédon simple est équivalent à un automate unidirectionnel.

Démonstration. Si l'automate boustrophédon est simple, alors toutes ses coupes sont simples (proposition 42). Ainsi, par le lemme 14, l'automate des coupes est fini. L'équivalence vient par la proposition 44. □

Remarque 27. La condition de commutativité est nécessaire : La série s qui au mot a^n associe le mot b^nc^n ($\langle s,a^n\rangle=b^nc^n$) est réalisable par un automate boustrophédon simple (et même par un automate circulaire simple), mais pas par un automate unidirectionnel.

Compter les calculs

On considère désormais un automate boustrophédon A.

En s'inspirant du raisonnement sur les booléens, on peut construire un automate unidirectionnel \mathcal{B} basé sur les coupes, qui n'est pas équivalent à \mathcal{A} , mais qui servira à compter les calculs dans celui-ci. Comme ils peuvent être en nombre infini, les poids de \mathcal{B} seront dans $(\mathbb{N} \cup \infty, +, .)$, noté \mathcal{N}_{∞} .

Définition 5.6. L'automate des coupes avec répétition $\mathcal{B} = (R, F, J, U)$ de \mathcal{A} est un automate unidirectionnel pondéré sur \mathcal{N}_{∞} , tel que :

- R est l'ensemble des coupes avec répétition de A. On notera C_1 l'ensemble des coupes simples, et $C_2 = R \setminus C_1$.
- $J(c) = \begin{cases} \infty \text{ si } c \text{ est une coupe initiale de } C_2, \\ 1 \text{ si } c \text{ est une coupe initiale de } C_1, \\ 0 \text{ sinon.} \end{cases}$
- F(c, a, c') est nul si c et c' ne sont pas chaînables par a. Si elles le sont, alors $F(c, a, c') = \begin{cases} \infty \text{ si } c \in C_1 \text{ et } c' \in C_2, \\ 1 \text{ sinon.} \end{cases}$
- U(c) = 1 si et seulement si c est une coupe finale.

L'idée de la construction se trouve en Figure 5.4.

Cet automate \mathcal{B} sert à compter les calculs dans \mathcal{A} qui finissent dans un état donné. En effet, tant qu'un chemin n'est pas passé dans C_2 , il est compté sur \mathbb{N} ,

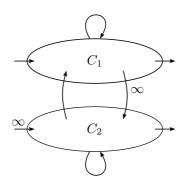


FIGURE 5.4 – Schéma de l'automate des coupes à répétition.

mais dès qu'il passe par un état de C_2 , cela veut dire qu'il existe un calcul dans \mathcal{A} admettant un circuit stationnaire. Cela implique l'existence d'une infinité de calculs finissant dans le même état et reconnaissant le même mot. D'où le poids ∞ pour tout calcul passant par un état de C_2 .

Proposition 46. Soit \mathcal{A} un automate boustrophédon δ -normalisé, \mathcal{B} son automate des coupes, et p un état final de \mathcal{A} . On pose \mathcal{B}_p l'automate obtenu à partir de \mathcal{B} en restreignant les coupes finales à celles finissant par p.

Alors pour tout mot w de A^* , pour tout état p, le nombre de calculs de A sur w finissant dans l'état p est égal à $\langle \mathcal{B}_p, w \rangle$.

Démonstration. Soit $w \in A^*$. Il existe une bijection immédiate entre les calculs simples de \mathcal{A} sur w et les calculs de \mathcal{B} passant uniquement par C_1 : tout calcul de \mathcal{A} est une suite d'états de C_1 , et réciproquement tout calcul de \mathcal{B} passant uniquement par C_1 permet bien de construire un unique calcul de \mathcal{A} . Ainsi, comme tout calcul de \mathcal{B} passant uniquement par C_1 à poids 1, le poids total de leur contribution au poids de w est égal au nombre de calculs simples de \mathcal{A} sur w.

Soit ρ un calcul de \mathcal{B} sur w passant par au moins un état de C_2 et finissant en p: on a donc $\langle \mathcal{B}_p, w \rangle = \infty$. Or ce calcul permet de reconstruire un calcul de \mathcal{A} , et possédant une coupe non simple, et donc un circuit stationnaire. On peut donc itérer ce circuit, et le nombre de calculs de \mathcal{A} se finissant en p est donc ∞ .

L'équivalence boustrophédon-unidirectionnel

Munis de ces outils, on cherche à déterminer quelles sont les séries réalisés par les automates boustrophédons pondérés lorsque le semi-anneau des poids est fini. Il s'agit d'une extension du résultat de [Rabin et Scott, 1959] et de [Shepherdson, 1959], que grâce à une remarque de Manfred Droste, nous avons étendu aux bi-monoïdes :

Théorème 5.1. Sur un bi-monoïde fort localement fini, la validité des automates boustrophédons est décidable. De plus, tout automate boustrophédon valide est équivalent à un automate unidirectionnel.

Notez que le résultat est encore vrai si le bi-monoïde n'est pas commutatif, contrairement à ce à quoi on pourrait s'attendre au vu des chaînages des coupes.

Il est immédiat que toute propriété montrée sur les automates pondérés sur un bi-monoïde fort fini s'étend au cas localement fini, puisque l'automate de départ est fini, et donc n'utilise qu'un nombre fini d'éléments du bi-monoïde. C'est pourquoi nous allons nous intéresser au cas fini.

Dans toute la suite, on considèrera $\mathcal{A} = (Q_+, Q_-, E, I, T)$ un automate boustrophédon pondéré sur \mathbb{K} , un bi-monoïde fort fini, et $\mathcal{B} = (R, F, J, U)$ son automate des coupes avec répétition.

On considère de plus que \mathcal{A} est caractéristique et δ -normalisé, ce qui est toujours faisable dans un bi-monoïde fini, cf. propositions 7 et 10 pages 31 et 34.

Les cas simples

Tout d'abord, on s'intéresse à la construction dans quelques cas où celle-ci est rendue plus simple de par la définition de la somme infinie. Il s'agit des bimonoïdes localement finis fortement idempotents, des bi-monoïdes localement finis dans lesquels la somme infinie n'est définie nulle part (sauf pour le 0), et des bi-monoïdes localement finis dans lesquels la somme infinie est définie partout.

Les bi-monoïdes localement finis fortement idempotents

On considère tout d'abord une classe de bi-monoïdes localement finis pour laquelle l'extension du résultat (et de la preuve) est immédiate.

Définition 5.7. Un bi-monoïde K est dit fortement idempotent si pour tout

$$x \in \mathbb{K}, x + x = x \text{ et } \sum_{i=0}^{\infty} x = x.$$

Exemple 20

Considérons le semi-anneau idempotent $(0,1,\omega)$, dans lequel $1+\omega=\omega$ et $\omega\omega=\omega$. Si la somme infinie de 1 vaut ω , il s'agit d'un bi-monoïde fini idempotent, mais pas fortement idempotent, mais si cette somme infinie est définie comme égale à 1, il est fortement idempotent.

On a vu lors de notre étude des coupes que la présence d'un circuit stationnaire d'un calcul était signe de l'existence d'une infinité de calculs. Mais comme \mathcal{A} est caractéristique, tous ces calculs ont même poids. L'addition étant idempotente, il suffit de compter un seul de ces calculs pour obtenir le même résultat.

On construit donc l'automate suivant :

Définition 5.8. L'automate des coupes simples d'un automate boustrophédon $\mathcal{A} = (Q_+, Q_-, E, I, T)$ caractéristique et δ -normalisé est un automate unidirectionnel $\mathcal{C} = (Q, E', I', T')$ défini comme suit :

- Q est l'ensemble des coupes simples de A
- $E'(c_1, a, c_2) = 1$ si c_1 et c_2 sont chaînables par a.
- I'(c) = 1 si et seulement si c est une coupe initiale.
- T'(c) = T(p), avec p le dernier état de c, si et seulement si c est une coupe finale.

Cet automate est équivalent à \mathcal{A} . En effet, si l'on simule tous les calculs simples, on simule tous les calculs, par l'idempotence de la somme (et de la somme infinie). Il suffit donc de montrer que cet automate simule bien tous les calculs simples de l'automate \mathcal{A} . Tout calcul de \mathcal{C} est une suite de coupes, et comme \mathcal{A} est δ -normalisé, entre chaque paire de coupes, il n'existe qu'un seul maillon. Ainsi, chaque calcul de \mathcal{C} simule bien une et une seule fois chaque calcul simple de \mathcal{A} .

Cette preuve du théorème 5.1 est exactement celle de [Shepherdson, 1959]. D'ailleurs, $\mathbb B$ est lui-même un bi-monoïde fort localement fini fortement idempotent.

Lorque le bi-monoïde n'est pas fortement idempotent, il se peut qu'il possède des propriétés qui simplifient la construction, notamment les deux cas suivants.

Aucune somme infinie n'est définie

Dans le cas où aucune somme infinie n'est définie, alors le poids d'un mot dont un calcul admet un circuit stationnaire n'est pas défini. En effet, si un calcul pour un mot w admet un circuit stationnaire, alors en répétant ce circuit, on génère une infinité de calculs pour w qui contribuent tous à son poids. Il faut donc calculer la somme infinie de ces calculs, or cette somme n'est pas définie.

Ainsi, si une coupe avec répétition est accessible dans \mathcal{B} , alors il existe un mot dont un calcul de \mathcal{A} admet un circuit stationnaire : \mathcal{A} n'est donc pas valide. Cette propriété est décidable.

Remarque 28. Si l'automate n'est pas valide, le langage des mots qui ne sont l'étiquette que de calculs simples forme un langage rationnel, et l'on peut construire l'automate qui le reconnaît à partir de \mathcal{A} .

Si en revanche aucune coupe avec répétition n'est accessible dans \mathcal{B} , alors \mathcal{A} est valide. On peut construire à partir de \mathcal{B} un automate unidirectionnel \mathcal{B}' qui est équivalent à \mathcal{A} . Il suffit de modifier la fonction finale, de sorte que le poids final d'une coupe simple $c = p_1 \dots p_n$ soit le poids final de p_n dans \mathcal{A} .

Toutes les sommes infinies sont définies

En revanche, dans le cas où toutes les sommes infinies sont définies, alors l'automate \mathcal{A} est immédiatement valide. De plus, on peut construire un automate qui lui est équivalent. Pour cela, on utilise une variante de \mathcal{B} , dans laquelle on annote les états avec un booléen indiquant si l'on a déjà croisé une coupe avec répétition dans le calcul. Il suffit ensuite d'adapter les poids.

Lemme 15. Soit A un K-automate boustrophédon caractéristique et δ -normalisé. Si toutes les sommes infinies de K sont définies, alors A est valide et il existe un automate unidirectionnel équivalent.

Démonstration. Soit $\mathcal{C}=(R',F',J',U')$ l'automate unidirectionnel sur \mathbb{K} suivant :

- $R' = C_1 \times \{1\} \cup (C_1 \cup C_2) \times \{\infty\}$ avec C_1 et C_2 définis comme à la définition 5.6.
- J'(c,n) vaut 1 si n=1 et que c est une coupe initiale de C_1 , ou bien si $n=\infty$ et que c est une coupe initiale de C_2 , et vaut 0 sinon.
- U'(c,1) = T(p), et $U'(c,\infty) = \infty.T(p)$, avec p le dernier état de c, si et seulement si c est une coupe finale.
- $F'((c, m), a, (c', n)) = \infty$ si $m = 1, n = \infty, c \in C_1$ et $c' \in C_2$.
- F'((c, m), a, (c', n)) = 1 si $c, c' \in C_1$ et que n = m, ou si $c' \in C_2$ et que $m = \infty$.

Le schéma représentant C se trouve en Figure 5.5, en posant x le poids final dans A d'une coupe de C_1 , et y celui d'une coupe de C_2 .

Montrons que cet automate est équivalent à \mathcal{A} : soit w un mot de A^* . Si l'on ignore les poids, il y a surjection immédiate entre les calculs de \mathcal{A} et les calculs de \mathcal{C} . Il y a même une bijection entre les calculs simples de \mathcal{A} et les calculs de \mathcal{C} finissant dans un état de $C_1 \times 1$.

De plus, tous les calculs ayant même image par la surjection peuvent être groupés en classes d'équivalence en fonction de leur état final, et l'on peut leur choisir un représentant qui soit un calcul avec un seul circuit stationnaire. La somme infinie des poids des représentants de ces calculs est équivalente à la somme de tous ces calculs.

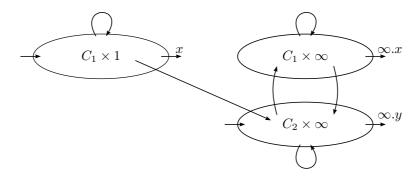


FIGURE 5.5 – Schéma de l'automate \mathcal{C} équivalent à \mathcal{A} lorsque toutes les coupes sont définies.

Or il existe une bijection entre ces représentants de ces calculs et les calculs de C. En effet, A est δ -normalisé.

Ainsi
$$\langle \mathcal{A}, w \rangle = \langle \mathcal{C}, w \rangle$$
.

Le cas général

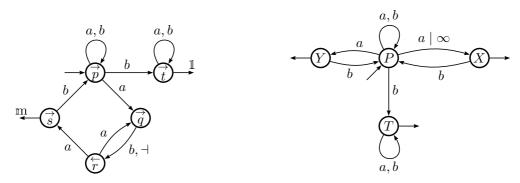
Il existe des bi-monoïdes pour lesquels certaines sommes infinies ne sont pas définies, alors que d'autres le sont. Par exemple le bi-monoïde \mathbb{K}_1 dont les opérations sont décrites ci-dessous est un tel bi-monoïde : une somme infinie de \mathbb{I} n'est pas définie, mais une telle somme plus un m vaut m.

Exemple 21

Le bi-monoïde K_1 servira d'exemple dans cette partie. Il pondère l'automate \mathcal{A}_3 de la Figure 5.6a. La Figure 5.6b représente son automate des coupes \mathcal{B}_3 . Pour plus de concision, nous avons noté P la coupe p, X la coupe qrqrs, Y la coupe qrs, et T la coupe t.

Pour savoir si le poids d'un mot est défini, il est dans ce cas nécessaire de connaître l'ensemble des poids de ses calculs et de déterminer si cette famille potentiellement infinie est sommable. Rappelons que cette famille sera représentée par une description, (cf. définition 1.14 page 12).

Pour pouvoir prendre cette décision localement, nous allons donc déterminiser l'automate, afin de grouper les différents calculs sur un même mot. En général, on ne connait pas d'algorithme pour déterminiser un automate pon-



- (a) Un \mathbb{K}_1 -automate boustrophédon \mathcal{A}_3 .
- (b) L'automate des coupes de A_3 .

FIGURE 5.6 – L'automate A_3 qui servira d'exemple dans cette partie, et son automate des coupes.

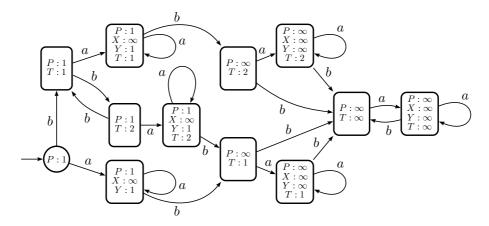


FIGURE 5.7 – L'automate \mathcal{B}_3 déterminisé modulo (1,2).

déré. Pour que cette déterminisation termine, nous allons l'effectuer dans un ensemble fini ce qui est une condition suffisante.

Plus précisément, nous construirons d'abord un automate unidirectionnel qui compte les calculs, vu en sous-section 5.2.3. Mais au lieu de compter les calculs sur $\mathbb{N} \cup \infty$, on les compte modulo l'ordre du bi-monoïde fini, c'est-à-dire sur $\mathcal{N}_{m,r}$, le semi-anneau $[0; m+r-1] \cup \{\infty\}$ où les opérations entre entiers finis sont modulo (m,r) (cf. section 1.3). C'est sur ce semi-anneau fini que l'on effectue la déterminisation.

Exemple 22

On peut voir la déterminisation de \mathcal{B}_3 en Figure 5.7. L'ordre de \mathbb{K}_1 est (1,2). Chaque état est une description, c'est-à-dire un vecteur de paires "coupe : n", avec $n \in \mathcal{N}_{m,r}$. Notez que l'on a pas encore représenté les poids finaux.

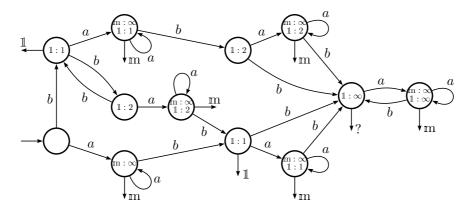


FIGURE 5.8 – L'automate \mathcal{D}_3 , équivalent à \mathcal{A}_3 .

Ainsi, chaque état stocke le nombre de calculs finissant dans la coupe que cet état encode, ainsi que le poids de ces calculs. On peut donc associer à chaque état c de ce nouvel automate \mathcal{D} la description de la famille, possiblement infinie, des poids des calculs qui finissent par c.

À partir de cette description, on peut décider si la famille décrite est sommable, et le cas échéant calculer son poids. On montre que s'il existe une description d'une famille non sommable, alors l'automate de départ n'était pas valide, et que si toutes les familles décrites sont sommables, alors en posant cette somme comme poids final de l'état, ce nouvel automate est équivalent à l'automate boustrophédon original.

La suite de cette section prouve le théorème 5.1, en construisant l'automate \mathcal{D} unidirectionnel équivalent à \mathcal{A} .

Soit donc $\mathcal{D} = (R', F', J', U')$ l'automate unidirectionnel sur \mathbb{K} tel que :

- $R' = R^{\mathcal{N}_{m,r}}$, les états sont des descriptions (cf. définition 1.14 page 12) de familles (possiblement infinies) de coupes;
- le seul état initial est le vecteur V_0 , avec $V_0(c) = J(c)$, *i.e.* le vecteur dans lequel le poids de chaque coupe est son poids initial dans \mathcal{B} ;
- $F'(v_1, a, v_2) = 1$ si et seulement si pour toute coupe c' de R, $v_2(c') = \sum_{c \in R} v_1(c).F(c, a, c')$;
- le poids final d'un état est la somme de la famille décrite par cet état si cette somme existe.

Exemple 23

L'automate \mathcal{D}_3 unidirectionnel équivalent à \mathcal{A}_3 obtenu par la construction cidessus se situe en Figure 5.8. On constate l'existence d'un état dont la somme de la description n'est pas définie : l'automate \mathcal{A}_3 n'est donc pas valide.

S'il existe des états dont le poids final n'est pas défini, *i.e.* qui décrivent une famille non sommable, alors l'automate boustrophédon original \mathcal{A} n'est pas valide. En effet, si l'on considère un mot w arrivant dans un tel état c, le calcul unique (l'automate est déterministe) arrivant en c sur ce mot simule l'ensemble des calculs de \mathcal{A} sur w, puisque c'est ce que fait \mathcal{B} et qu'il s'agit juste d'une déterminisation modulo l'ordre, déterminisation qui conserve donc les poids. Ainsi, comme tous les calculs pour w sont bien représentés par la description de l'état c, la non sommabilité de c implique la non validité de \mathcal{A} .

On a donc également que si le poids final de c est défini, il est égal au poids de w dans \mathcal{A} . Ainsi si tous les états sont des descriptions sommables, l'automate \mathcal{D} et l'automate \mathcal{A} sont équivalents.

Remarque 29. L'ensemble des mots qui rendent non valide un automate boustrophédon sur \mathbb{K} , un bi-monoïde fort localement fini, est rationnel.

Conclusion Ainsi, on a montré que, comme dans le cas booléen, les séries réalisées par les automates boustrophédons sur les bi-monoïdes forts finis étaient exactement celles réalisées par les automates unidirectionnels.

Cependant, on peut remarquer que dans le cas général, avec un automate boustrophédon avec n états forward et n états backward, pondéré sur un bimonoïde localement fini d'ordre (m, r), la taille de l'automate unidirectionnel équivalent que nous obtenons grâce à cette construction est de $(m+r)^{O(n^{4n})}$.

Index

z-représentation, 113 z-reconnaissable, 113 automate, 18 équivalence, 19 à navette, 21 δ-normalisé, 33 boustrophédon, 25 validité, 27 caractéristique, 30 circulaire, 20 des follow, 59 des contextes, 86 des coupes, 124 avec répétition, 125 simples, 128 des dérivées, 53 des positions, 59 positif, 32 r-local, 69 simple, 120 standard, 32 Thompson-like, 66 unidirectionnel, 18 bi-monoïde, 6 fortement idempotent, 127 localement fini, 11 bloc de demi-tours, 122 calcul, 19, 25 simple, 119 chaînage, 122 chemin, 19, 25	longueur, 25 circuit stationnaire, 119 circulation sur un automate, 29 comportement d'un automate à navette, 23 unidirectionnel, 19 boustrophédon, 26 circulaire, 21 contexte, 86 coupe, 118 automate, 124 avec répétition, 120 simple, 118 dérivée, 54 description d'une famille, 12 équivalence £-équivalence, 6 R-équivalence, 6 d'automates, 19 état backward, 22, 25 forward, 22, 25 étiquette d'un chemin, 19, 25 d'une transition, 19 étoile d'une matrice, 100 de Hadamard, 13 de Kleene, 7, 13 expression dérivée, 52
chemin, 19, 25 étiquette, 25	dérivée, 52 de Hadamard, 14

interprétation, 14	de Conway, 10
position, 58	fini, 7
pré-Hadamard, 50	rationnellement additif, 10
rationnelle, 14	série, 12
c 111 =	de Hadamard, 14
famille, 7	formelle, 12
sommable, 7	propre, 12
Hadamard	rationnelle, 14
étoile, 13	réalisée par un automate, 19
expression, 14	somme, 12
-	terme constant, 12
inverse, 72	simple
produit, 13	automate, 120
quotient, 72	calcul, 119
identités de Conway, 10	coupe, 118
inverse de Hadamard, 72	support
mverse de Hadamard, 72	d'une fonction, 19
langage, 5	plein, 12
reconnu, 19	,
lettre, 5	transition
,	entrante, 19
maillon, 122	sortante, 19
monoïde, 5	1. 1
commutatif, 6	validité
de transition, 85	d'un automate
libre, 6	à navette, 23
morphisme de, 6	boustrophédon, 27
	circulaire, 21
opérations	d'une expression, 15
rationnelles, 14	
terme a terme, 14	
ordre	
d'un élément, 11	
d'un monoïde, 12	
.1 . 2 14	
polynôme, 14	
produit	
de Cauchy, 13	
de Hadamard, 13	
quotient de Hadamard, 72	
revêtement d'automates, 28	
semi-anneau, 6	

Bibliographie

- Anselmo, Marcella, 1990. Two-way automata with multiplicity. Dans *ICALP'90*, tome 443 de *Lect. Notes in Comput. Sci.*, pages 88–102. Springer. doi:10.1007/BFb0032024.
- Anselmo, Marcella et Bertoni, Alberto, 1992. Two-way probabilistic automata and rational power series. Dans *Proc. IV Conv. It. Inform. Teor.*, page 9–23. World Scientific.
- Antimirov, Valentin, 1996. Partial derivatives of regular expressions and finite automaton constructions. *Theor. Comp. Sci.*, 155(2):291 319. doi: http://dx.doi.org/10.1016/0304-3975(95)00182-4.
- BERSTEL, Jean et REUTENAUER, Christophe, 2010. Noncommutative Rational Series with Applications, tome 137 de Encyclopedia of Mathematics and its Applications. Cambridge University Press.
- BIRGET, Jean-Camille, 1989. Concatenation of inputs in a two-way automaton. *Theor. Comput. Sci.*, 63(2):141–156. doi:10.1016/0304-3975(89)90075-3.
- BIRGET, Jean-Camille, 1990. Two-way automaton computations. *RAIRO Theor. Inf. and Appl.*, 24:47–66.
- CARON, Pascal et Flouret, Marianne, 2003. Glushkov construction for series: The non commutative case. *Int. J. Comput. Math.*, 80(4):457–472. doi:10.1080/0020716021000038992.
 - URL http://dx.doi.org/10.1080/0020716021000038992
- CARTON, Olivier, 2012. Two-way transducers with a two-way output tape. Dans *DLT'12*, tome 7410 de *Lect. Notes in Comput. Sci.*, pages 263–272. Springer.
- Champarnaud, Jean-Marc, Laugerotte, Éric, Ouardi, Faissal et Ziadi, Djelloul, 2003. From regular weighted expressions to finite automata. Dans *Proc. of CIAA 2003*, tome 2759 de *LNCS*, pages 49–60. doi: 10.1007/3-540-45089-0 6.
 - URL https://doi.org/10.1007/3-540-45089-0_6

- CONWAY, John H., 1971. Regular Algebra and Finite Machines. Chapman and Hall, London.
- DANDO, Louis-Marie et LOMBARDY, Sylvain, 2017. From hadamard expressions to weighted rotating automata and back. Dans *International Conference on Implementation and Application of Automata*, pages 163–174. Springer.
- DANDO, Louis-Marie et LOMBARDY, Sylvain, 2018a. On Hadamard Series and Rotating Q-Automata. Dans Mathematical Foundations of Computer Science, MFCS 2018, tome 117 de 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018), pages 6:1–6:14. Igor Potapov and Paul Spirakis and James Worrell, Schloss Dagstuhl, Liverpool, United Kingdom. doi:10.4230/LIPIcs.MFCS.2018.6.
- DANDO, Louis-Marie et LOMBARDY, Sylvain, 2018b. Two-way automata over locally finite semirings. Dans *International Conference on Descriptional Complexity of Formal Systems*, pages 62–74. Springer.
- EILENBERG, Samuel, 1974. Automata, languages, and machines. Academic press.
- ÉSIK, Zoltán et KUICH, Werner, 2002. Rationally additive semirings. J. UCS, 8(2):173-183. doi:10.3217/jucs-008-02-0173.
- FLIESS, Michel, 1974. Matrices de hankel. J. Math. Pures Appl, 53(9):197–222.
- Guillon, Bruno, 2016. Two-wayness: Automata and Transducers. Thèse de doctorat, Université Paris Diderot Università degli studi di Milano, Paris.
- ILIE, Lucian et Yu, Sheng, 2003. Follow automata. *Inf. Comput.*, 186(1):140–162. doi:10.1016/S0890-5401(03)00090-7.
- KLEENE, S. C., 1956. Representation of events in nerve nets and finite automata. *Automata Studies*, pages 3–41.
- LOMBARDY, Sylvain, 2016. Two-way representations and weighted automata. RAIRO Theor. Inf. and Appl., 50(4):331–350. doi:10.1051/ita/2016026. URL http://dx.doi.org/10.1051/ita/2016026
- LOMBARDY, Sylvain et SAKAROVITCH, Jacques, 2005. Derivatives of rational expressions with multiplicity. *Theor. Comp. Sci.*, 332(1–3):141 177. doi: http://dx.doi.org/10.1016/j.tcs.2004.10.016.
- MOLINARI, Luca Guido, 2008. Determinants of block tridiagonal matrices. Linear algebra and its applications, 429(8-9):2221–2226.

- PAZ, Azaria, 1971. Introduction to Probabilistic Automata (Computer Science and Applied Mathematics). Academic Press, Inc., Orlando, FL, USA.
- PÉCUCHET, Jean-Pierre, 1985. Automates boustrophedon, semi-groupe de birget et monoide inversif libre. RAIRO Theor. Inf. and Appl., 19(1):71–100.
- PERRIN, Dominique, 1995. Les débuts de la théorie des automates. *Technique* et science informatiques, 14(4):409–433.
- RABIN, Michael O. et Scott, Dana, 1959. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125. doi:10.1147/rd.32.0114.
- SAKAROVITCH, Jacques, 2009. *Elements of Automata Theory*. Cambridge University Press.
- SCHÜTZENBERGER, Marcel-Paul, 1961. On the definition of a family of automata. *Inform. and Control*, 4:245–270.
- SCHÜTZENBERGER, Marcel-Paul, 1962. On a theorem of R. Jungen. *Proc. Amer. Math. Soc.*, 13(6):885–890.
- SHEPHERDSON, John C., 1959. The reduction of two-way automata to one-way automata. *IBM J. Res. Dev.*, 3(2):198–200. doi:10.1147/rd.32.0198.
- STALLINGS, John R, 1983. Topology of finite graphs. *Inventiones mathematicae*, 71(3):551–565.
- THOMPSON, Ken, 1968. Programming techniques: Regular expression search algorithm. Communications of the ACM, 11(6):419–422.
- ZIADI, Djelloul, PONTY, Jean-Luc et CHAMPARNAUD, Jean-Marc, 1997. Passage d'une expression rationnelle à un automate fini non-déterministe. Bulletin of the Belgian Mathematical Society-Simon Stevin, 4(1):177–203.