Ines Klimann

Autour de divers problèmes de décision sur les automates

Mémoire d'habilitation à diriger des recherches

soutenue le 17 janvier 2014

devant le jury composé de

Laurent Bartholdi,	rapporteur
Frédérique Bassino,	rapporteuse
Christian Choffrut,	rapporteur
Jean Mairesse,	examinateur
Nicolas Ollinger,	examinateur
Géraud Sénizergues,	examinateur
Andrzej Zuk,	examinateur

Université Paris Diderot, Sorbonne Paris Cité 2014

Remerciements

Je voudrais commencer par remercier Laurent Bartholdi, Frédérique Bassino et Christian Choffrut qui ont accepté de rapporter ce mémoire. Plus particulièrement Christian pour le temps qu'il a pris à mes côtés à pointer les imperfections et lacunes dans ce que j'avais écrit : la clarté et la fluidité du texte en ont été bien améliorées.

Je voudrais également remercier tous les membres de mon jury – les rapporteurs sus-cités, ainsi que Jean Mairesse, Nicolas Ollinger, Géraud Sénizergues et Andrzej Zuk – d'avoir accepté de participer à mon jury. Jean est bien plus qu'un membre de jury en fait, puisque sans lui ce mémoire n'existerait sûrement pas ou, en tout cas, aurait clairement un autre contenu. En effet c'est lui qui m'a poussée à continuer la recherche, noyée que j'étais (avec bonheur certes) au milieu de mes cours juste après mon recrutement, puis à la reprendre après mes retours de congés maternité, et c'est enfin lui qui m'a suggéré de travailler sur ce sujet passionnant que sont les groupes et semi-groupes d'automate. Qu'il en soit ici chaleureusement remercié.

Mon travail et les résultats qui y sont attachés n'auraient évidemment pas été les mêmes sans mes co-auteurs, mais aussi sans ceux avec qui nous avons essayé, sans toujours y parvenir, ni sans ceux avec qui nous avons scientifiquement échangé au détour d'un couloir ou devant leurs cafés (parfois même devant un tableau) : merci à Ali, Guillaume, Sylvain, Jean (encore), Matthieu, Christophe, Pierre-Cyrille, Cyril, Olivier, Charles, Jean, Laure, Luc, Luc, Olivier et Thomas.

La vie quotidienne serait en noir et blanc sans la couleur qu'y apportent les cours et ceux avec qui j'ai pris (et prends encore) plaisir à enseigner : Anne, Arnaud, Christophe, Dominique, Jean-Marie, François, Hugues, Matthieu, Mihaela, Vincent, Yann. Et Jean-Baptiste avec qui je ne crois pas avoir enseigné, mais que j'ai embêté assez souvent avec des questions de cours pour le rajouter ici.

Merci aux autres qui peuplent les couloirs et s'arrêtent pour un mot ou pour une visite à la cantine : Carole, Enrica, Isabelle, Matthieu, Michel, Peter, Vlady, Wolfgang.

Merci à l'équipe administrative et technique du liafa, de l'ufr et du département, qui évite que les petits problèmes ne deviennent plus gros : Houy, Laïfa, Nathalie et Noëlle au 4ème, Jean-Michel, Laurent, Marie-Claude, Mickael, Patricia, Raja et Sonia au 3ème ; Yannick et Christine loin là-bas.

Je ne peux pas terminer sans remercier ma famille pour sa patience et son soutien.

À Pierrot, Margot et Eugène

Introduction

Un automate est un objet fini qui permet de représenter des ensembles a priori infinis. L'existence d'une telle représentation a plusieurs buts : décrire l'ensemble bien entendu, mais également faire des calculs ou prendre des décisions sur cet ensemble. C'est dans ce cadre-là que se situe ce mémoire.

Les types d'ensembles représentés peuvent être de natures diverses. Je me penche plus particulièrement sur les séries rationnelles à variables non commutatives (chapitres 1 à 3) et sur les (semi-)groupes (chapitres 4 à 9).

Pourquoi ces ensembles-là en particulier ? Dans la droite ligne de ma thèse, j'ai continué à m'intéresser aux automates à multiplicités et aux séries. Puis, fin 2005, Jean Mairesse nous a proposé un groupe de lecture sur l'article *Automata, dynamical systems, and groups* de R. Grigorchuk, V. Nekrashevich et V. Sushchanskiĭ. Je pense que nous sommes loin d'avoir extrait et assimilé tout ce qui peut l'être dans cet article, mais il a clairement représenté pour moi un début de changement de thème, les techniques que je maîtrisais pour les séries venant enrichir ma façon d'envisager les (semi-)groupes d'automate. Ce mémoire est donc naturellement séparé en deux parties.

* * *

La première partie concerne les séries à variables non commutatives. Le premier problème sur les séries auquel je m'intéresse est le problème de séquentiabilité : étant donné un automate à multiplicités dans $\mathbb{R}_{\max}$, existe-t-il un automate séquentiel équivalent ? Avec Sylvain Lombardy, Jean Mairesse et Christophe Prieur nous avons montré que si l'automate de départ est finiment ambigu, ce problème est décidable et qu'on peut effectivement obtenir l'automate séquentiel équivalent. Si l'automate n'est pas séquentialisable mais uniquement désambiguisable, la même construction s'applique pour obtenir un équivalent non ambigu [KLMP04]. Ces résultats font l'objet du chapitre 2.

D. Kirsten et S. Lombardy ont étendu par la suite ce résultat aux automates polynomialement ambigus, avec des méthodes complètement différentes [KL09]. La question dans le cadre général n'est pas résolue ; Sylvain m'a dit un jour penser que c'était décidable, mais sans avoir de démonstration aboutie.

Le deuxième problème sur lequel je me penche porte sur les liens entre une série et son support. D. Kirsten a donné une caractéristique algébrique des semi-anneaux de coefficients tels que les supports des séries reconnaissables sont toujours reconnaissables (semi-anneaux SR). Cette propriété ne dépend pas du cardinal de l'alphabet à condition qu'il contienne au moins deux lettres. Avec Guillaume Chapuy, nous avons montré qu'on ne peut pas se restreindre à une seule lettre : il existe des semi-anneaux non-SR tels que les supports des séries reconnaissables sur une lettre sont tous reconnaissables. Nous avons également donné un premier exemple de semi-anneau non-SR ne contenant pas $\mathbb{Z}$ [CK11]. Le chapitre 3 détaille ce résultat.

* * *

La deuxième partie de ce mémoire porte sur les (semi-)groupes d'automate. Des automates même très simples permettent de représenter des groupes et semi-groupes extrêmement compliqués, comme des groupes à croissance intermédiaire. L'avantage de cette représentation est que l'automate est un levier combinatoire puissant pour l'étude du groupe. De fait, ces groupes ont fourni de nombreux exemples et contre-exemples en théorie des groupes, depuis leur introduction dans les années soixante.

Le problème principal auquel je m'intéresse est la décision de la finitude de (semi-)groupes d'automates. Avec Ali Akhavi, Sylvain Lombardy, Jean Mairesse et Matthieu Picantin, nous avons donné une série de conditions nécessaires ou suffisantes pour tester la finitude [AKL+12], qui améliorent les résultats obtenus avec les conditions antérieurement connues. Puis avec ces deux derniers, nous avons proposé d'utiliser la minimisation pour accélérer les calculs sur les groupes d'automate [KMP12]. Enfin j'ai montré que la finitude est décidable pour les groupes engendrés par des automates inversibles-réversibles à deux états (ou deux lettres), et que, de façon surprenante, les semi-groupes engendrés par des automates réversibles à deux états sont soit finis, soit libres [Kli13]. Tous ces résultats sont exposés dans les chapitres 6 à 8.

* * *

La section 3 du chapitre 2 et le chapitre 9 me permettent d'exposer mes perspectives de recherche. Ce sont parfois des problèmes sur lesquels j'ai déjà travaillé, sans obtenir de résultats concluants, mais qui me trottent encore dans la tête ; parfois des problèmes sur lesquels je travaille actuellement ; et parfois des questions que je me pose sans avoir vraiment eu le temps de regarder. Mais tous ont pour but de mieux comprendre la nature de l'objet représenté par l'automate, que ce soit dans le cas des séries ou dans celui des (semi-)groupes.

* * *

Le parti pris de ce document est de ne pas détailler les démonstrations quand c'est possible, mais d'en donner une intuition. Cela permet je pense de comprendre plus clairement les origines d'un résultat.

 [réf.]

Les résultats que j'ai obtenus (avec mes co-auteurs) sont précédés d'une ligne verticale, comme ce paragraphe, avec le logo adéquat.

 **Conjecture 0**

Mes conjectures se présentent de la même manière, avec un autre logo.


automate
ensemble
langage
série
peut
si
alphabet
état
produit
séries
tout
non
mot
langages
états
deux
support
chemin
reconnaissable
caractéristique
multiplicités
coefficient
semi-anneau
existe
transition
lettre
équivalent
nombre
structure
coefficients
transitions
exemple
section
déterministe
reconnaissables
finie
forme
mots
cette
chapitre
donc
plus
rationnel
finiment
dont
comme
initial
max
vide
alors
suite
ambigu
unique
matrice
problème
être
somme
reconnaît
propriété
fini
corps
cet
nombre
union
commutatif
éléments
opérations
poids
figure
non-ambigu
expression
coordonnées
semi-groupe
rationnelles
Schützenberger
rationnels
rationnelle
ordre
théorème
famille
neutre
KLMP04
final
façon
muni
lettres
max-plus
appelle
monoïde
entre
reconnu
exemples
question
équivalence
colonne
élément
CHAPITRE
suivant
résultat
nulle
chaque
fait
multiplication
donnée
montre
est-à-dire
tous
Figure
SÉRIES
LANGAGES
premier
étoile
séquentiel
toute
partir
reconnue
tel
également
resp
Rmax
A1
A2
Xi

Toute chaîne
A deux poids,
Toute peine
En a trois.

La belle Hélène, livret de Henri Meilhac et Ludovic Halévy

Cette partie contient uniquement des notions classiques ; aucun résultat original n'y est présenté. Les démonstrations et constructions non détaillées pourront être trouvées dans [Eil74, HU79, Aut94, Sako3, Sako9, BR84, Ber79, DKV09]. Je donnerai en particulier des références précises de [Sako3] pour les points importants dans la suite de ce mémoire.

1

Automates, langages et séries

Dans ce chapitre je présente l'objet d'où tout part : les automates. Les automates sont des machines finies qui permettent de décrire des ensembles a priori infinis. Je définis également des structures algébriques utiles à la fois pour généraliser l'objet automate lui-même et pour généraliser et structurer les ensembles que l'on peut décrire grâce à cet objet.

Une première généralisation des automates est présentée en fin de chapitre.

1 Langages et automates

Un **alphabet** est un ensemble fini non vide dont les éléments sont des **lettres**. Une suite finie de lettres est un **mot** et un ensemble de mots un **langage**.

La **longueur** d'un mot $\mathbf{w}$, notée $|\mathbf{w}|$, est son nombre de lettres. L'unique mot de longueur nulle est appelé **mot vide** et est noté 1. Le nombre d'occurrences d'une lettre a dans un mot $\mathbf{w}$ est noté $|\mathbf{w}|_a$. On peut concaténer deux mots en écrivant les lettres du second à la suite des lettres du premier : cette opération est associative et admet le mot vide comme élément neutre.

Le **monoïde libre** engendré par un alphabet Σ est noté Σ^*, c'est l'ensemble des mots sur Σ muni de la concaténation comme produit.

Soient deux langages L et M. On considère l'union de ces ensembles, appelée **somme** ou **union** des deux langages et notée indifféremment $L + M$ ou $L \cup M$; et le **produit** de ces deux langages donné par

$$LM = \{\mathbf{w} \in \Sigma^* \mid \exists \mathbf{u} \in L, \exists \mathbf{v} \in M \text{ tels que } \mathbf{w} = \mathbf{uv}\}.$$

Le produit des langages est une opération associative, d'élément neutre le singleton contenant le mot vide $\{1\}$. On peut ainsi définir la n^{e} puissance d'un langage puis son **étoile** :

$$L^* = \sum_{n \geq 0} L^n.$$

Somme et produit munissent l'ensemble des langages sur un alphabet donné d'une structure de semi-anneau.

La somme finie, le produit et l'étoile constituent les **opérations rationnelles** sur les langages.

1.1 Expressions rationnelles et langages rationnels

Soit un alphabet Σ. Les **expressions rationnelles** sur Σ sont les formules bien formées à partir des lettres de Σ, de deux constantes 0 et 1, de l'opérateur unaire * et des opérateurs binaires $+$ et $\cdot$.

A chaque expression rationnelle sur Σ, on associe un langage sur Σ de manière inductive :

– $L(0) = \emptyset$, $L(1) = \{1\}$ et $L(a) = a$, pour toute lettre a,
– $L(E^*) = L(E)^*$, $L(E + F) = L(E) + L(F)$ et $L(E \cdot F) = L(E)L(F)$.

Un langage est **rationnel** s'il peut être associé à une expression rationnelle. Bien entendu, chaque langage rationnel peut être décrit par une infinité d'expressions rationnelles.

1.2 Automates et langages reconnaissables

Un automate est un graphe orienté et étiqueté par des lettres appartenant à un alphabet (fini). Deux sous-ensembles de ses sommets – qu'on appelle **états** pour un automate – jouent un rôle particulier, les **états initiaux** et les **états finaux**. On s'intéresse à l'ensemble des étiquettes des **chemins réussis** (partant d'un état initial et allant vers un état final), qu'on appelle **langage reconnu** par l'automate.

Deux automates sont **équivalents** s'ils reconnaissent le même langage. Un langage est **reconnaissable** s'il existe un automate qui le reconnaît.

En général, on donne un automate sous la forme d'un quintuplet $\mathcal{A} = (Q, \Sigma, I, \Delta, F)$ où :
– Q est un **ensemble d'états** fini non vide,
– Σ est un alphabet (fini non vide),
– $I \subseteq Q$ est l'ensemble des **états initiaux**,
– $\Delta \subseteq Q \times \Sigma \times Q$ est l'ensemble des **transitions**,
– $F \subseteq Q$ est l'ensemble des **états finaux**.

Souvent on omet de mentionner l'alphabet et l'automate est alors donné sous forme d'un quadruplet (Q, I, Δ, F). La plupart du temps on préfère représenter un automate graphiquement, comme à la figure 1.1.

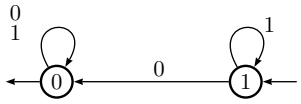


FIGURE 1.1 – Exemple d'automate.

Certains automates ont des propriétés structurelles remarquables. Nous en listons quelques-unes ci-dessous.

Un automate est **déterministe** s'il possède un unique état initial et si de tout état il part au plus une transition étiquetée par une lettre donnée. L'ensemble des transitions d'un automate déterministe est en général noté de façon fonctionnelle, $\mathcal{A} = (Q, i, \cdot, F)$: s'il existe une transition étiquetée par $a \in \Sigma$ et partant de $p \in Q$, l'état d'arrivée de cette transition est noté $p \cdot i$. Tout automate est équivalent à un automate déterministe et l'obtention d'un automate déterministe équivalent est effective. Je donne ici la construction dite *par sous-ensembles*, mais il existe d'autres méthodes de déterminisation.

Soit un automate fini $\mathcal{A} = (Q, I, \Delta, F)$ sur l'alphabet Σ. On considère l'automate déterministe sur Σ

$$\mathcal{D} = (\mathcal{P}(Q), \{I\}, \cdot, \{R \subseteq Q \mid R \cap F \neq \emptyset\})$$

(où $\mathcal{P}(Q)$ désigne l'ensemble des parties de Q), dont la fonction de transition est donnée par :

$$P \cdot a = \{q \in Q \mid \exists p \in P, (p, a, q) \in \Delta\}.$$

On montre de façon simple que $\mathcal{A}$ et $\mathcal{D}$ sont équivalents.

Un automate est **complet** si pour toute lettre, il part de tout état au moins une transition étiquetée par cette lettre. On peut compléter un automate en ajoutant un état puit (sans modifier le langage reconnu).

Un automate est **émondé** si tout état est accessible (c'est-à-dire qu'il existe un chemin allant d'un état initial vers cet état) et co-accessible (c'est-à-dire qu'il existe un chemin allant de cet état vers un état final). On peut émonder un automate sans modifier le langage reconnu.

Un automate est **non-ambigu** si tout mot reconnu étiquette exactement un chemin réussi. Un automate déterministe est non-ambigu.

Tout automate possède un **automate minimal** équivalent : c'est l'automate déterministe équivalent le plus petit en le nombre d'états. Il est unique à numérotation des états près. Nous ne justifierons pas ici cette affirmation, le lecteur pourra se reporter à [Sako3, § I.3.3]. Nous allons nous contenter de décrire un algorithme pour obtenir cet automate.

Soit un automate déterministe $\mathcal{A} = (Q, \Sigma, i, \cdot, F)$. L'**équivalence de Nerode** sur Q est la limite de la suite d'équivalences $(\equiv_k)_{k\in\mathbb{N}}$ définie récursivement par :

– l'équivalence $\equiv_0$ possède deux classes d'équivalence : F et son complémentaire dans Q,
– pour $k \geq 0$ et deux états $p, q \in Q$: $p \equiv_{k+1} q \Leftrightarrow \big(p \equiv_k q$ et $\forall i \in \Sigma,\ p\cdot i \equiv_k q\cdot i\big)$.

L'ensemble des états étant fini, cette suite est ultimement constante et de plus dès que deux éléments consécutifs sont égaux, on a atteint sa limite. On note $[i]$ la classe d'équivalence de l'état initial par l'équivalence de Nerode.

L'automate quotient $\mathcal{A}/\equiv = (Q/\equiv, \Sigma, [i], F/\equiv)$ est le **minimisé** de $\mathcal{A}$.

La complexité temporelle dans le pire des cas de la minimisation, en utilisant l'algorithme d'Hopcroft, est $\mathcal{O}(\#\Sigma \#Q \log \#Q)$ [AHU74].

1.3 Théorème fondamental

Le théorème fondamental de la théorie des langages formels et des automates a été énoncé par S. Kleene en 1956 :

Un langage est rationnel si et seulement s'il est reconnaissable.

Le passage d'un langage donné sous forme d'expression rationnelle à un automate et vice-versa est effectif. Je ne donne pas la construction, ultra classique, mais je rappelle quand même comment on construit le produit de deux automates (produit qui sert en particulier à montrer que l'intersection de deux langages reconnaissables est reconnaissable).

Soient deux automates $\mathcal{A}_i = (Q_i, \Sigma, I_i, \Delta_i, F_i)$, $i \in \{1, 2\}$, sur le même alphabet. Le **produit des deux automates** $\mathcal{A}_1$ et $\mathcal{A}_2$ est l'automate

$$\mathcal{A} = (Q_1 \times Q_2, \Sigma, I_1 \times I_2, \Delta, F_1 \times F_2).$$

où

$$\Delta = \{((p_1, p_2), a, (q_1, q_2)) \mid (p_1, a, q_1) \in \Delta_1 \text{ et } (p_2, a, q_2) \in \Delta_2\}.$$

On étend de façon naturelle la notion de produit à une famille finie quelconque d'automates.

Le théorème fondamental ci-dessus, couplé avec la déterminisation des automates, permet de caractériser de manière très simple les langages rationnels sur un alphabet à une lettre. En effet, un automate déterministe sur une lettre a une forme de "poêle à frire" comme montré sur la figure 1.2. On en déduit qu'un langage rationnel sur un alphabet à une seule lettre $\{a\}$ est de la forme a^N où $N \subseteq \mathbb{N}$ est ultimement périodique.

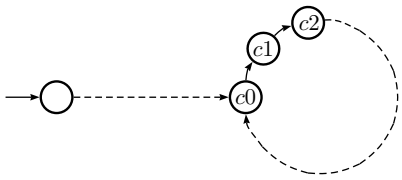


FIGURE 1.2 – Forme générique en "poêle à frire" d'un automate déterministe sur un alphabet à une lettre.

Le théorème de Kleene permet également de donner une condition nécessaire pour qu'un langage soit rationnel, appelée "lemme de pompage". L'idée de base est la suivante : si on lit un mot suffisamment long dans un automate, on finit par retomber sur le même état et on peut alors boucler indéfiniment en restant dans le langage. Cette propriété permet de montrer que le langage des mots ayant le même nombre d'occurrences de a et de b n'est pas rationnel. De même le langage des puissances d'un nombre donné $\{a^{(k^n)} \mid n \in \mathbb{N}\}$ $(k > 1)$ n'est pas rationnel.

La plupart des questions qu'on peut se poser sur les automates amènent des réponses assez faciles, du fait que les langages reconnaissables sont rationnels et qu'on peut minimiser un automate. Par exemple pour savoir si deux automates sont équivalents, il suffit de les réduire et de comparer leurs automates minimaux. De même pour savoir si un automate reconnaît tous les mots.

Il existe néanmoins des problèmes sur les automates qui apportent des réponses plus compliquées. Par exemple le problème dit de *hauteur d'étoile*. La hauteur d'étoile d'un langage est le minimum des hauteurs d'étoiles des expressions rationnelles qui le dénotent. Le problème de la décidabilité de la hauteur d'étoile bornée a fait coulé beaucoup d'encre [Egg63, Has88, Kir05, LS03].

2 Structures algébriques

Tout au long de ce mémoire, nous aurons l'occasion de croiser diverses structures algébriques, aussi bien permettant d'enrichir les automates (chapitre 1) qu'étant elles-mêmes fabriquées par des automates (chapitre 4). Nous les décrivons sommairement ici.

Une structure algébrique est souvent donnée sous la forme d'un ensemble et d'un certain nombre d'opérations. Pour un type de structure **T**, on peut considérer la structure de type **T** *engendrée* par certains éléments : il faut que ces éléments appartiennent à une structure de type **T** qui fournit les opérations qui permettent de les manipuler et on prend la plus petite structure de type **T** qui contient ces éléments, c'est-à-dire le plus petit ensemble contenant ces éléments et stable par les opérations sus-citées.

2.1 Avec une seule opération interne

Un **semi-groupe** est une structure algébrique composée d'un ensemble et d'une opération binaire interne et associative. La présence d'un élément neutre pour cette opération en fait un **monoïde**.

Quelques exemples de semi-groupes : $(\mathbb{N} - \{0\}, +)$, l'ensemble des mots non vides sur un alphabet muni de la concaténation.

Quelques exemples de monoïdes : $(\mathbb{N}, +)$, $(\mathbb{Z}, *)$, $(\mathbb{Z}/2\mathbb{Z}, +)$, l'ensemble des mots sur un alphabet muni de la concaténation, l'ensemble des langages (rationnels) sur un alphabet muni de la somme, ce même ensemble muni du produit, l'ensemble des applications d'un ensemble dans lui-même muni de la composition, l'ensemble des matrices carrées de taille n muni de la multiplication matricielle.

Un semi-groupe ou un monoïde est **commutatif** si son opération est commutative.

Les liens entre semi-groupes (*resp.* monoïdes) et automates et langages sont forts. Ainsi à tout automate on peut associer un semi-groupe particulier dit **semi-groupe des transitions** qui est le semi-groupe des matrices de transitions de l'automate et à tout langage on peut associer un semi-groupe particulier dit **semi-groupe syntaxique** qui est le quotient de l'ensemble des mots non vides par la congruence la plus grossière qui sature ce langage. Ces semi-groupes, ainsi que les monoïdes du même nom, ont fait et font encore l'objet d'études poussées (voir [Pin97] et ses références).

Si tous les éléments d'un monoïde admettent un inverse, on parle alors de **groupe**.

Quelques groupes classiques : $(\mathbb{Z}/n\mathbb{Z}, +)$, $((\mathbb{Z}/p\mathbb{Z})^*, *)$ avec p premier, $(\mathbb{Q} - \{0\}, *)$, $\mathbb{R}$, l'ensemble des permutations d'un ensemble muni de la composition, l'ensemble des matrice carrées inversibles (*resp.* unitaires) de taille n.

L'**ordre d'un (semi-)groupe** est son nombre d'éléments. L'**ordre d'un élément** est l'ordre du sous-(semi-)groupe engendré par cet élément. Il est bien connu que, dans un groupe, l'ordre d'un élément divise l'ordre du groupe (quand tout est fini, bien entendu). Dans le chapitre 8, nous nous intéresserons à l'ordre des (semi-)groupes d'automate.

2.2 Avec deux opérations internes

Un **semi-anneau** est une structure algébrique composée d'un ensemble et de deux opérations binaires internes et associatives, qu'on appelle souvent addition et multiplication et telles que l'addition confère à l'ensemble une structure de monoïde commutatif et la multiplication une structure de monoïde, que la multiplication est distributive sur l'addition et le neutre de l'addition absorbant.

Quelques exemples de semi-anneaux classiques : $\mathbb{B}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$. Si Σ est un alphabet, l'ensemble des langages et l'ensemble des langages rationnels $\mathrm{Rat}\Sigma^*$ sur Σ sont également des semi-anneaux.

Il existe une famille de semi-anneaux très utilisée en informatique et en automatique, les *semi-anneaux exotiques*. Le **semi-anneau tropical** $\mathbb{N}_{\min} = (\mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0)$ en est un exemple : la somme de deux éléments est le minimum usuel et leur produit la somme usuelle. Le neutre de l'addition est alors l'élément $+\infty$ et le neutre de la multiplication l'élément 0. De la même façon, on construit les semi-anneaux $\mathbb{Z}_{\max}$, $\mathbb{R}_{\max}$, etc.

Un **anneau** est un semi-anneau dont l'addition induit une structure de groupe ; $\mathbb{Z}$ et $\mathbb{Z}/n\mathbb{Z}$ constituent des exemples classiques d'anneaux. La **caractéristique** d'un anneau est l'ordre pour sa loi additive de l'élément neutre de la multiplication, si cet ordre est fini, sinon il est de caractéristique nulle : $\mathbb{Z}/n\mathbb{Z}$ est de caractéristique n, tandis que $\mathbb{Z}$ est de caractéristique nulle.

Un **corps** est un anneau dont la multiplication induit une structure de groupe sur l'ensemble privé du neutre de l'addition. Les corps sont soit de caractéristique nulle (exemples : $\mathbb{Q}$, $\mathbb{R}$), soit de caractéristique un nombre premier (exemples : $\mathbb{Z}/p\mathbb{Z}$, p premier).

3 Séries formelles et automates

Les automates tels que vus en section 1 de ce chapitre permettent de représenter des applications allant de l'ensemble des mots sur un alphabet donné vers les booléens. Or il est beaucoup de circonstances où l'on est en présence d'applications vers un autre semi-anneau. Pour représenter de telles applications, on ajoute alors un coefficient sur chaque transition de l'automate, qu'on appelle **production**.

Quand les productions sont des langages rationnels sur un alphabet donné, on appelle **transducteur** l'automate et **transduction** l'objet reconnu. Si elles ont une valeur numérique ($\mathbb{R}$, $\mathbb{Z}_{\min}$, ...), on parle d'**automate à multiplicités** et de **série** (à variables non commutatives).

Ces automates avec productions possèdent de nombreux champs d'applications : systèmes à événements discrets [Gau95], reconnaissance de langages [MPR05], compression d'images [CIvR99]. Ils servent également à montrer des résultats sur les automates sans multiplicités, par exemple liés au problème de hauteur d'étoile [Egg63, Has88, Kir05, LS03].

L'ajout de ce coefficient fait disparaître l'existence systématique d'un automate déterministe équivalent (le terme consacré dans le cas d'automates avec productions est **séquentiel**) et par là-même complique tout.

Le pouvoir d'expression des automates à multiplicités est suffisamment grand pour que bon nombre de questions soient indécidables. Ainsi, certains problèmes extrêmement simples sans coefficient deviennent indécidables avec, comme l'équivalence entre deux automates max-plus [Kro94].

Pouvoir décider de l'existence d'un automate séquentiel équivalent n'est pas évident. On sait résoudre ce problème pour les transducteurs lettre-à-lettre (les productions sont des lettres) [Cho77], mais pas pour les transducteurs généraux par exemple.

Dans toute la section, Σ désigne un alphabet fini et $\mathbb{S}$ un semi-anneau d'éléments neutres 0 pour l'addition et 1 pour la multiplication.

La notion de série formelle étend celle de langage : à chaque mot est associé un coefficient qui représente le "nombre de fois" où le mot apparaît dans la série.

Une **série formelle** S sur Σ à coefficients dans $\mathbb{S}$ est une application de Σ^* dans $\mathbb{S}$. Pour tout mot $\mathbf{w}$, l'image par S de $\mathbf{w}$ est notée $(S, \mathbf{w})$, c'est le **coefficient** de $\mathbf{w}$ dans S. Le coefficient du mot vide est appelé **coefficient constant** de S. La série elle-même est notée $S = \sum_{\mathbf{w}\in\Sigma^*} (S, \mathbf{w})\mathbf{w}$.

On note $\mathbb{S}\langle\langle\Sigma\rangle\rangle$ l'ensemble des séries formelles sur Σ à coefficients dans $\mathbb{S}$. Cet ensemble est muni des opérations suivantes (appelées **somme** et **produit de Cauchy**) qui lui confèrent une structure de semi-anneau :

$$\forall \mathbf{w} \in \Sigma^*, \qquad (S+T, \mathbf{w}) = (S, \mathbf{w}) + (T, \mathbf{w}) \quad \text{et} \quad (ST, \mathbf{w}) = \sum_{\mathbf{uv}=\mathbf{w}} (S, \mathbf{u})(T, \mathbf{v}).$$

La somme qui apparaît dans le calcul de $(ST, \mathbf{w})$ est finie (elle contient $|w| + 1$ termes), ce qui légitime son existence.

L'ensemble des mots dont le coefficient est non nul est le **support** de la série. Une série est dite **constante** si son support contient au plus le mot vide. On identifie les séries constantes et les éléments du semi-anneau des coefficients. Les multiplication externes à gauche ($S \mapsto aS$) et à droite ($S \mapsto Sa$) se définissent alors de façon immédiate en appliquant le produit de séries.

Le monoïde libre Σ^* s'injecte naturellement dans $\mathbb{S}\langle\langle\Sigma\rangle\rangle$, l'image d'un mot $\mathbf{w}$ est encore notée $\mathbf{w}$: c'est la série définie par $(\mathbf{w}, \mathbf{u}) = 0$ pour tout mot $\mathbf{u} \neq \mathbf{w}$ et $(\mathbf{w}, \mathbf{w}) = 1$.

Si le semi-anneau $\mathbb{S}$ est l'ensemble des langages sur un alphabet fixé Ξ, les séries obtenues s'appellent des **transductions** de Σ^* dans Ξ^* : à chaque mot de Σ^* est associé un langage sur Ξ. Le lecteur intéressé pourra trouver son bonheur dans [Ber79].

Une série formelle S est **propre** si son coefficient constant est nul.

Dans le cas d'une série S propre, la famille $(S^n)_{n\geq 0}$ est localement finie et donc sommable : un mot $\mathbf{w}$ donné ne peut apparaître que dans un nombre fini de S_n. On appelle **étoile** de S la somme de cette famille : $S^* = \sum_{n\geq 0} S^n$.

Somme, produit et étoile de séries propres sont les **opérations rationnelles** sur les séries.

3.1 Séries rationnelles

De même que pour les langages, à toute expression rationnelle sur Σ, on peut associer une série sur Σ et $(\mathbb{S}, +, \times, 0, 1)$ de manière inductive :

– $L(0) = 0, \quad L(1) = 1 \quad$ et $\quad L(a) = 1a = a$, pour toute lettre a,
– $L(E^*) = L(E)^*$ si E série propre, $\quad L(E + F) = L(E) + L(F) \quad$ et $\quad L(E \cdot F) = L(E)L(F)$.

Une série est **rationnelle** si elle peut être associée à une expression rationnelle. Comme pour les langages, cette association n'est pas unique.

3.2 Automates à multiplicités et séries reconnaissables

Un **automate à multiplicités** dans $\mathbb{S}$ sur l'alphabet Σ est un quadruplet

$$\mathcal{A} = (Q, \alpha, \mu, \beta)$$

où :
– Q est un ensemble fini d'états,
– $\mu : \Sigma^* \to \mathbb{S}^{Q \times Q}$ est un morphisme de monoïdes,
– $\alpha \in \mathbb{S}^{1 \times Q}$ est un vecteur ligne,
– $\beta \in \mathbb{S}^{Q \times 1}$ est un vecteur colonne.
La série **reconnue** par $\mathcal{A}$ est la série

$$S(\mathcal{A}) = \sum_{\mathbf{w} \in \Sigma^*} (\alpha \mu(\mathbf{w}) \beta) \mathbf{w}.$$

Pour simplifier, on dira que $\mathcal{A}$ est un automate sur $\mathbb{S}$ et sur Σ et on pourra le noter (α, μ, β) sans expliciter l'ensemble d'états.

On peut représenter graphiquement un automate à multiplicités, comme illustré en figure 1.3, en ajoutant une production pour étiqueter chaque transition : la production associée à la transition $p \xrightarrow{a} q$ est le coefficient (p, q) de la matrice $\mu(a)$. Un état i est **initial** (*resp.* **final**) si $\alpha_i \neq 0$ (*resp.* $\beta_i \neq 0$). Le **poids** d'un chemin est le produit des productions de ses transitions et le **coefficient** d'un mot la somme des poids des chemins réussis qu'il étiquette.

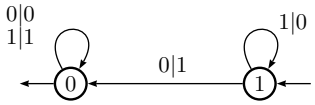


Figure 1.3 – Exemple d'automate à multiplicités.

Deux automates à multiplicités sont **équivalents** s'ils reconnaissent la même série. Une série est **reconnaissable** s'il existe un automate (à multiplicités) qui la reconnaît.

Si $\mathbb{S}$ est le semi-anneau des booléens $\mathbb{B}$, on retombe alors sur la notion d'automate définie en section 1. On qualifiera un tel automate d'**automate booléen**.

3.3 Théorème fondamental

M.-P. Schützenberger a étendu en 1961 le théorème de Kleene aux séries formelles [BR84] :

Une série formelle est rationnelle si et seulement si elle est reconnaissable.

Tout comme pour la section 1.3, je ne donnerai ici aucune démonstration de ce résultat ultra-classique, mais je rappelle quelques constructions utiles pour le montrer et qui nous servirons dans la suite de ce mémoire.

Pour les automates booléens, on pouvait définir le produit d'une famille finie d'automates. Ici, on peut définir plusieurs produits, dépendant essentiellement du choix que l'on fait pour les multiplicités des transitions dans l'automate produit.

Soient deux automates $\mathcal{A}_1$ et $\mathcal{A}_2$, d'ensembles d'états respectifs Q_1 et Q_2. Posons $\mathcal{A}_i = (\alpha^{\imath}, \mu^{\imath}, \beta^{\imath})$, pour $i = 1$ ou 2. L'**automate produit** de $\mathcal{A}_1$ et $\mathcal{A}_1$ est l'automate $\mathcal{A}_1 \times \mathcal{A}_2 = (A, M, B)$ à multiplicités dans le semi-anneau $\mathbb{S}^2$, d'ensemble d'états $Q = Q_1 \times Q_2$ et tel que :

$$\begin{aligned} \forall p, q \in Q,\ p = (p_1, p_2), q = (q_1, q_2) \quad & \alpha_p = (\alpha^{\imath}_{p_1}, \alpha^{\imath}_{p_2}), \\ & \forall a \in \Sigma,\ \mu(a)_{p,q} = (\mu^{\imath}(a)_{p_1,q_1}, \mu^{\imath}(a)_{p_2,q_2}), \\ & \beta_p = (\beta^{\imath}_{p_1}, \beta^{\imath}_{p_2}). \end{aligned}$$

Clairement cet automate satisfait

$$\forall \mathbf{u} \in \Sigma^*,\ (\mathcal{A}_1 \times \mathcal{A}_2, \mathbf{u})_1 + (\mathcal{A}_1 \times \mathcal{A}_2, \mathbf{u})_2 = (S(\mathcal{A}_1) + S(\mathcal{A}_2), \mathbf{u}) = \alpha^{\imath}\mu^{\imath}(\mathbf{u})\beta^{\imath} + \alpha^{\imath}\mu^{\imath}(\mathbf{u})\beta^{\imath}.$$

Le **produit tensoriel** de $\mathcal{A}_1$ et $\mathcal{A}_2$, noté $\mathcal{A}_1 \odot \mathcal{A}_2$ est défini de la même manière que $\mathcal{A}_1 \times \mathcal{A}_2$, excepté le fait que la multiplicité $(x_1, x_2) \in \mathbb{S}^2$ est remplacée par la multiplicité $x_1 x_2 \in \mathbb{S}$.

On étend de façon naturelle ces deux notions de produit à une famille finie quelconque d'automates.

4 Séquentialité et ambiguïté

Si l'automate booléen sous-jacent (c'est-à-dire en oubliant les productions) est déterministe, on dit que l'automate à multiplicités est **séquentiel**. Tout automate n'est pas séquentialisable, c'est-à-dire qu'il n'existe pas nécessairement un automate séquentiel équivalent. La figure 1.4 donne un exemple d'automate non séquentialisable sur un alphabet à une lettre [KLMP04]. La série reconnue par cet automate est

$$(S, a^n) = \begin{cases} 0 & \text{si } n \text{ impaire,} \\ n & \text{si } n \text{ paire.} \end{cases}$$

Il est clair que dans un automate séquentiel équivalent, l'unique chemin réussi étiqueté par a^{n+1} est obtenu en concaténant l'unique chemin réussi étiqueté par a^n et une transition étiquetée par a. Cette transition doit alors produire $-n$ ou $n+1$, suivant la parité de n, ce qui implique que l'automate ne peut être fini.

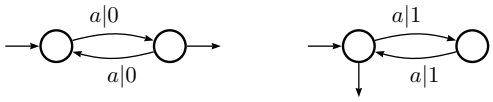


FIGURE 1.4 – Automate non séquentialisable.

A noter que qualifier un automate de **séquentialisable** fait non seulement penser qu'il existe un automate séquentiel équivalent, mais également qu'on est capable de l'obtenir, et ce n'est pas forcément la même chose. A ma connaissance, quel que soit le semi-anneau considéré, on n'a jamais montré qu'un automate possèdait un équivalent séquentiel sans être capable de construire cet équivalent. Le chapitre 2 est consacré à ce problème pour $\mathbb{S} = \mathbb{R}_{\max}$.

Un automate à multiplicités est **non-ambigu** si l'automate booléen sous-jacent l'est. Un automate à multiplicités est **finiment ambigu** si le nombre de chemins réussis étiquetés par un mot donné est uniformément borné, sinon il est **infiniment ambigu**.

Une série est **séquentielle** (*resp.* **non-ambiguë**, **finiment ambiguë**) s'il existe un automate séquentiel (*resp.* non-ambigu, finiment ambigu) qui la reconnaît. Une série est **intrinsèquement infiniment ambiguë** s'il n'existe pas d'automate finiment ambigu qui la reconnaît.

la ambigüedad es una riqueza

"Pierre Menard, autor del Quijote", Jorge Luis Borges

Cette partie expose les résultats contenus dans les articles suivants :

[CK11] G. Chapuy and I. Klimann. On the supports of recognizable series over a field and a single letter alphabet. *Inf. Process. Lett.*, 111(23-24) :1096–1098, 2011.

[KLMP03] I. Klimann, S. Lombardy, J. Mairesse, and Ch. Prieur. Deciding the sequentiality of a finitely ambiguous max-plus automaton. In *Developments in Language Theory*, pages 373–385, 2003.

[KLMP04] I. Klimann, S. Lombardy, J. Mairesse, and Ch. Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theor. Comput. Sci.*, 327(3) :349–373, 2004.

2

Séquentialité et ambiguïté sur $\mathbb{R}_{max}$

Ce chapitre est consacré à la question de la décidabilité de la séquentialité et de la désambiguisation de séries reconnaissables à coefficients dans le semi-anneau $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +)$. Un automate à multiplicités dans ce semi-anneau sera qualifié d'**automate max-plus**.

Les étapes de cette procédure de décision sont les suivantes :
- on transforme l'automate en une union finie équivalente d'automates non-ambigus, tous de même support (corollaire 2.4),
- on vérifie une certaine *propriété de dominance* (section 2) sur ce nouvel automate qui assure que la série reconnue est non-ambiguë,
- on construit un automate non-ambigu équivalent (proposition 2.5),
- on décide de la séquentiabilité de cet automate (théorème 2.6).

On note Rat l'ensemble des séries rationnelles, Seq l'ensemble des séries séquentielles, NAmb l'ensemble des séries non-ambiguës et FAmb l'ensemble des séries finiment ambiguës sur $\mathbb{R}_{\max}$ et FSeq l'ensemble des séries reconnaissables par union finie d'automates séquentiels. La figure 2.1 illustre la hiérarchie stricte de séries qui apparaît à partir des critères de séquentialité, d'ambiguïté et de rationnalité.

Dans tout ce chapitre, les automates considérés sont à coefficients dans $\mathbb{R}_{\max}$.

1 Passage d'un automate finiment ambigu à une union finie d'automates non ambigus

A. Weber a montré dans [Web94] qu'à partir d'un automate finiment ambigu sur $\mathbb{N}_{\max}$ on peut construire une union finie d'automates non ambigus équivalente. Avec Sylvain Lombardy, Jean Mairesse et Christophe Prieur, nous avons donné une construction complètement différente basée sur le revêtement de Schützenberger d'un automate [KLMP04].

Soit un automate émondé $\mathcal{A} = (\alpha, \mu, \beta)$. Le **passé** $\mathsf{Past}_{\mathcal{A}}(p)$ d'un état p est le langage des mots étiquetant un chemin allant d'un état initial à p. Son **futur** $\mathsf{Fut}_{\mathcal{A}}(p)$ est le langage des mots étiquetant un chemin allant de p à un état final.

Lemme 2.1. Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates et $\mathcal{D}$ le déterminisé de $\mathcal{A}$ obtenu par la méthode des sous-ensembles (voir p.13).

(i) Pour tout état P de $\mathcal{D}$, on a

$$\mathsf{Past}_{\mathcal{D}}(P) \subseteq \bigcap_{p \in P} \mathsf{Past}_{\mathcal{A}}(p), \quad \text{et} \quad \mathsf{Fut}_{\mathcal{D}}(P) = \bigcup_{p \in P} \mathsf{Fut}_{\mathcal{A}}(p).$$

(ii) Pour tout état (p, q) du produit tensoriel $\mathcal{A} \odot \mathcal{B}$, on a

$$\mathsf{Past}_{\mathcal{A}\odot\mathcal{B}}(p, q) = \mathsf{Past}_{\mathcal{A}}(p) \cap \mathsf{Past}_{\mathcal{B}}(q), \qquad \mathsf{Fut}_{\mathcal{A}\odot\mathcal{B}}(p, q) = \mathsf{Fut}_{\mathcal{A}}(p) \cap \mathsf{Fut}_{\mathcal{B}}(q).$$

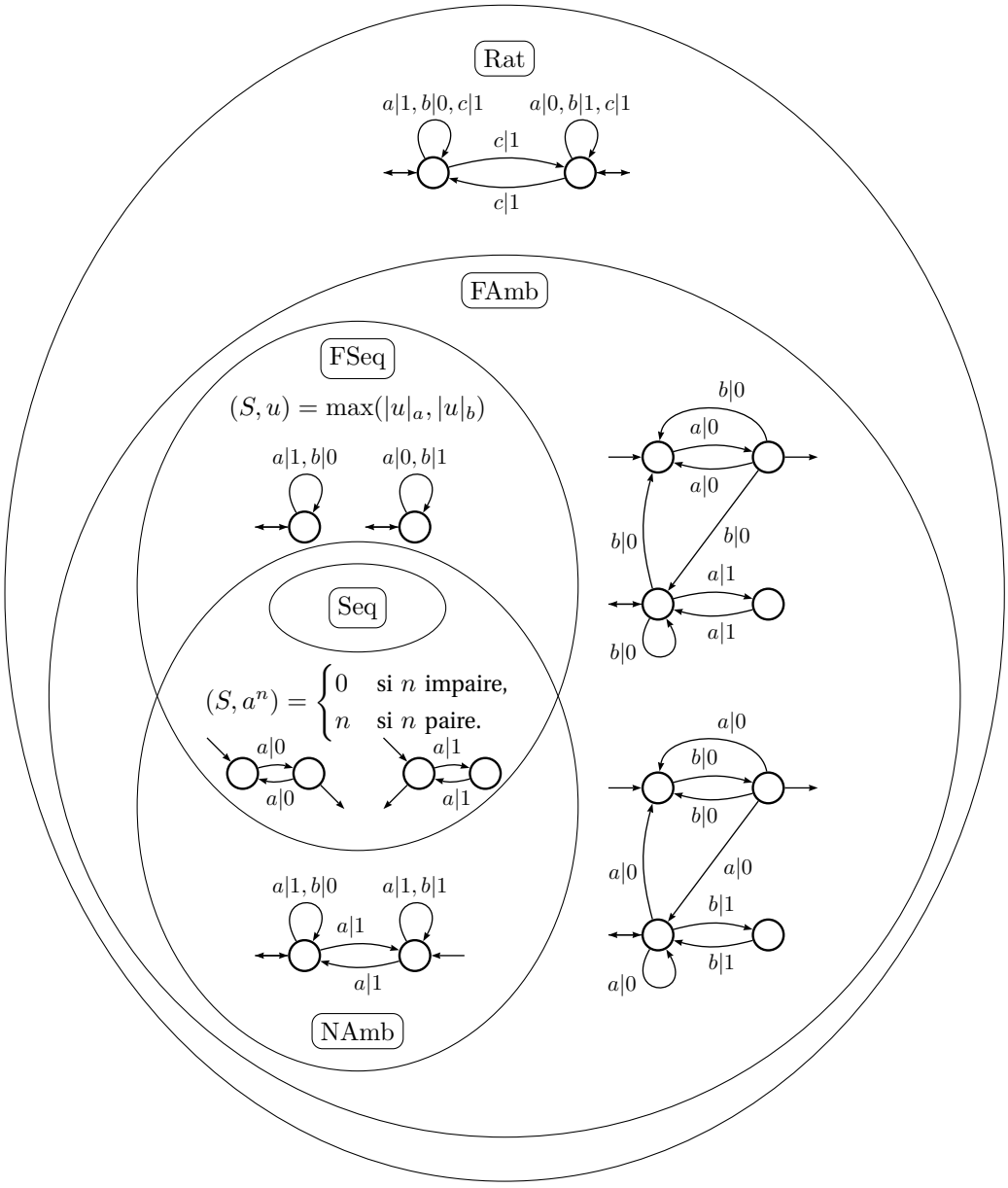


FIGURE 2.1 – Hiérarchie stricte de séries [KLMP04].

Les constructions et résultats suivants (propositions 2.2 et 2.3) sont inspirés de Schützenberger [Sch76]. Ils sont donnés explicitement dans [Sako3, chapitre II].

Soient un automate $\mathcal{A}$ et son déterminisé par la méthode des sous-ensembles $\mathcal{D}$. On appelle **revêtement de Schützenberger** la partie accessible et co-accessible du produit $\mathcal{S} = \mathcal{A} \odot \mathcal{D}$, comme illustré en figure 2.2.

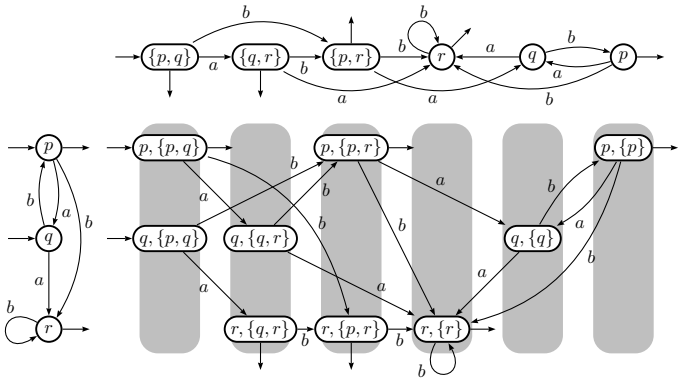


FIGURE 2.2 – L'automate $\mathcal{A}$ (à gauche), son déterminisé (en haut) et son revêtement de Schützenberger.

Proposition 2.2.

(i) Les états de $\mathcal{S}$ sont exactement les couples (p, P) acessibles et co-accessibles tels que P est un état de $\mathcal{D}$ et $p \in P$.

On appelle **colonne** un ensemble $\{(p, P) \mid p \in P\}$ d'états de $\mathcal{S}$, pour P fixé (en gris sur la figure 2.2).

(ii) La surjection canonique ψ de l'ensemble des transitions de $\mathcal{S}$ sur l'ensemble des transitions de $\mathcal{A}$ induit une bijection entre les chemins réussis de $\mathcal{S}$ et ceux de $\mathcal{A}$.

(iii) Soit P un état de $\mathcal{D}$. Alors pour tout $p \in P$, on a

$$\mathsf{Past}_{\mathcal{S}}(p, P) = \mathsf{Past}_{\mathcal{D}}(P), \qquad \mathsf{Fut}_{\mathcal{S}}(p, P) = \mathsf{Fut}_{\mathcal{A}}(p).$$

Ainsi, tous les états d'une même colonne ont le même passé.

Dans $\mathcal{S}$, les transitions de même étiquette partant de la même colonne et arrivant dans le même état sont dites **concurrentes**. De la même façon, différents états finaux appartenant à la même colonne sont dits **concurrents**. Un **ensemble concurrent** est un ensemble maximal de transitions ou d'états finaux concurrents.

Proposition 2.3. Soient $\mathcal{A}$ un automate max-plus et $\mathcal{S}$ son revêtement de Schützenberger. On construit $\mathcal{U}$ à partir de $\mathcal{S}$ en ne gardant qu'une transition dans chaque ensemble concurrent (de façon arbitraire) et en transformant tous les états finaux d'une colonne, sauf un, en états non finaux (également de façon arbitraire). Alors :

(i) $\mathsf{Past}_{\mathcal{U}}(p, P) = \mathsf{Past}_{\mathcal{S}}(p, P)$.

(ii) Les futurs des états d'une même colonne de $\mathcal{U}$ sont disjoints et

$$\bigcup_{p \in P} \mathsf{Fut}_{\mathcal{U}}(p, P) = \bigcup_{p \in P} \mathsf{Fut}_{\mathcal{S}}(p, P).$$

Il en résulte que l'automate $\mathcal{U}$ est équivalent à $\mathcal{A}$ et non-ambigu.

Le revêtement de Schützenberger $\mathcal{S}$ d'un automate finiment ambigu $\mathcal{A}$ peut être utilisé pour convertir $\mathcal{A}$ en une union finie d'automates non ambigus, chacun reconnaissant le même langage que $\mathcal{A}$. En effet, les transitions concurrentes de $\mathcal{S}$ n'appartiennent à aucun de ses circuits, ce qui permet d'affirmer que $\mathcal{S}$ contient au plus une transition dans chaque ensemble concurrent [KLMP04].

Une conséquence de ce résultat est que pour tout chemin de $\mathcal{S}$ (et donc tout chemin de $\mathcal{A}$), on peut calculer un automate non-ambigu contenant ce chemin.

Les ensembles concurrents de $\mathcal{S}$ sont partiellement ordonnés car ils n'appartiennent à aucun circuit. Soit C l'ensemble des ensembles concurrents maximaux de $\mathcal{S}$.

Considérons l'algorithme suivant.

- Soient $\mathcal{S}_1$ et $\mathcal{S}_2$ deux copies de $\mathcal{S}$. Pour chaque ensemble concurrent X de C, soit x un élément de X,
  - si x est une transition, supprimer toutes les transitions de $X - \{x\}$ de $\mathcal{S}_1$ et supprimer x de $\mathcal{S}_2$
  - si x est un état final, rendre tous les états de $X - \{x\}$ dans $\mathcal{S}_1$ et x dans $\mathcal{S}_2$ non finaux.
- Appliquer l'algorithmique inductivement sur $\mathcal{S}_1$ et $\mathcal{S}_2$.

La sortie de cet algorithme est un ensemble fini d'automates non-ambigus : $\mathcal{F}$ (de cardinal éventuellement supérieur au degré d'ambiguïté de $\mathcal{A}$). Chacun reconnaît le même langage que $\mathcal{A}$ et tout chemin de $\mathcal{S}$ apparaît dans au moins un de ces automates.

Si $\mathcal{A}$ est à multiplicités, on peut ajouter à chaque transition de $\mathcal{S}$ la production correspondante dans $\mathcal{A}$.

Clairement, comme il existe une bijection entre les chemins de $\mathcal{A}$ et ceux de $\mathcal{S}$, les automates à multiplicités $\mathcal{S}$ et $\mathcal{A}$ sont équivalents. De plus, chaque chemin de $\mathcal{S}$ apparaît dans un des automates de $\mathcal{F}$ (éventuellement plusieurs, mais cela n'a pas d'importance car le semi-anneau est idempotent), donc $\mathcal{F}$ réalise la même série que $\mathcal{A}$.

[KLMP04]

Corollaire 2.4. On peut, à partir d'un automate max-plus finiment ambigu, construire un automate équivalent, union finie d'automates max-plus de même support.

2 Décider la non-ambiguité d'une série donnée par un automate finiment ambigu

Dans cette section, nous donnons une condition nécessaire et suffisante pour qu'une série donnée par une union finie d'automates non-ambigus de même support soit non-ambiguë.

On considère une famille finie d'automates max-plus émondés non-ambigus tous de même support

$$\left(\mathcal{A}_i = (\alpha^{\imath} \in \mathbb{R}_{\max}^{Q_i}, \mu^{\imath} : \Sigma^* \to \mathbb{R}_{\max}^{Q_i \times Q_i}, \beta^{\imath} \in \mathbb{R}_{\max}^{Q_i})\right)_{i \in I},$$

et soit $\mathcal{P}$ leur produit, d'ensemble d'états $Q = \Pi_{i \in I} Q_i$, de cardinal N.

Soit θ un circuit simple de $\mathcal{P}$ dont le poids est $(x^{\imath})_{i \in I}$. L'ensemble des **coordonnées victorieuses de** θ, noté $\mathrm{Vict}\,(\theta)$, est l'ensemble des coordonnées pour lesquelles le poids de θ est maximal, c'est-à-dire :

$$\mathrm{Vict}\,(\theta) = \left\{ i \in I | x^{\imath} = \max_{j \in I} \{x^{\jmath}\} \right\}.$$

Définition et notation peuvent naturellement être étendues à un chemin (*resp.* à une composante fortement connexe) de $\mathcal{P}$, en prenant l'intersection des coordonnées victorieuses des circuits simples du chemin (*resp.* de la composante fortement connexe).

On définit la propriété de dominance (**P**) par :

Pour tout chemin réussi π de l'automate produit $\mathcal{P}$, l'ensemble des coordonnées victorieuses de π est non vide.

Le nombre de circuits simples étant fini, la propriété (**P**) est décidable.

Supposons que le produit d'automates $\mathcal{P}$ satisfasse la propriété de dominance (**P**).

Prenons les notations suivantes :

$$M = \max(\max_{i,a,p,q} \mu^{i}(a)_{p,q}, \max_{i,p} \beta_p^{i}) - \min(\min_{i,a,p,q} \mu^{i}(a)_{p,q}, \min_{i,p} \beta_p^{i}).$$

et pour $\mathbf{x} = (x_i)_{i\in I} \in \mathbb{R}_{\max}^I$, on pose

$$\check{\mathbf{x}} = \min_{i\in I}\{x_i | x_i \neq -\infty\} \quad \text{et} \quad \underline{\mathbf{x}} = \mathbf{x} - (\check{\mathbf{x}}, \dots, \check{\mathbf{x}}).$$

Soit $I = \{1, \dots, n\}$. Définissons un nouvel automate $\mathcal{V}$ dont les états appartiennent à $\mathbb{R}_{\max}^n \times Q$.

Etats initiaux. Soit un tuple $(q^{1}, \dots, q^{n})$ tel que q^{i} soit un état initial de $\mathcal{A}_i$. On pose $\mathbf{a} = (\alpha_{q^{1}}^{1}, \dots, \alpha_{q^{n}}^{n})$, alors $(\underline{\mathbf{a}}, q^{1}, \dots, q^{n})$ est un état initial de $\mathcal{V}$, de poids entrant $\check{\mathbf{a}}$.

Transitions. Si $\mathbf{p} = (z_1, \dots, z_n, p^{1}, \dots, p^{n})$ est un état de $\mathcal{V}$, alors pour toute transition de $\mathcal{P}$ de type $(p^{1}, \dots, p^{n}) \xrightarrow{a|\mathbf{x}} (q^{1}, \dots, q^{n})$ telle que $x_i \neq -\infty$ pour tout i, nous construisons une transition de $\mathcal{V}$ d'origine $\mathbf{p}$ étiquetée par a comme suit. Soient $\mathbf{y} = (z_1 + x_1, \dots, z_n + x_n)$ et V l'ensemble des coordonnées victorieuses du sous-graphe fortement connexe maximal de $(q^{1}, \dots, q^{n})$ dans $\mathcal{P}$. Comme $\mathcal{P}$ satisfait la condition (**P**), l'ensemble V est non vide. Soit $j \in V$ tel que $y_j = \min_{k\in V}\{y_k \mid y_k \neq -\infty\}$, et soit $\mathbf{y}' \in \mathbb{R}_{\max}^n$ défini par :

$$\forall i, \quad y'_i = \begin{cases} -\infty & \text{si } y_i < y_j - NM, \\ y_i & \text{sinon.} \end{cases}$$

Le tuple $(\underline{\mathbf{y}'}, q^{1}, \dots, q^{n})$ est un état de $\mathcal{V}$ et nous définissons la transition suivante dans $\mathcal{V}$:

$$(z_1, \dots, z_n, p^{1}, \dots, p^{n}) \xrightarrow{a|\check{\mathbf{y}}'} (\underline{\mathbf{y}'}, q^{1}, \dots, q^{n}).$$

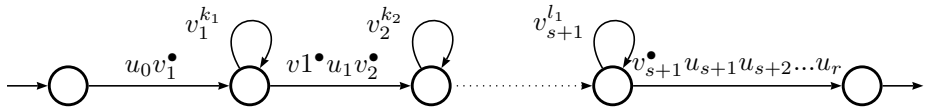
Etats finaux. Les états de $\mathcal{V}$ de la forme $\mathbf{q} = (z_1, \dots, z_n, q^{1}, \dots, q^{n})$ où pour tout i q^{i} est un état final de $\mathcal{A}_i$; le poids de l'arc sortant est $\max_{i\in I}\{z_i + \beta_{q^{i}}^{i}\}$.

[KLMP04]

Proposition 2.5. Soit une famille $(\mathcal{A}_i)_{i\in I}$ d'automates max-plus émondés et non-ambigus, de même support. Soit $\mathcal{P}$ le produit de ces automates. La série $\bigoplus_{i\in I} S(\mathcal{A}_i)$ est non ambiguë si et seulement si $\mathcal{P}$ satisfait la propriété (**P**). Dans ce cas, l'automate $\mathcal{V}$ défini ci-dessus est fini, non-ambigu et réalise la série $\bigoplus_{i\in I} S(\mathcal{A}_i)$.

Démonstration. Montrons que (**P**) est une condition nécessaire de non-ambiguïté, par l'absurde : supposons qu'on n'a pas cette propriété et qu'il existe un automate non ambigu $\mathcal{U}$ à n états qui reconnaît la série $S = \bigoplus_{i\in I} S(\mathcal{A}_i)$.

Comme on n'a pas la propriété (**P**), il existe un chemin π dans le produit $\mathcal{P}$ dont l'ensemble de coordonnées victorieuses est vide. Ce chemin peut être décomposé en $\pi_0, \theta_1, \pi_1, ..., \pi_r$ où les θ_i sont des circuits et $\bigcap_i \mathrm{Vict}(\theta)_i = \emptyset$. Soient $\mathbf{u}_i$ l'étiquette de π_i et $\mathbf{v}_i$ celle de θ_i. Soit s l'entier maximal tel que $V = \bigcap_{i\leqslant s} \mathrm{Vict}(\theta)_i \neq \emptyset$. Soit $\mathbf{w}_{k,l} = \mathbf{u}_0\mathbf{v}_1^k\mathbf{u}_1 ... v_s^k\mathbf{u}_s\mathbf{v}_{s+1}^l\mathbf{u}_{s+1}\mathbf{u}_{s+2}...\mathbf{u}_r$. Pour tout k, l, le mot $\mathbf{w}_{k,l}$ est reconnu par $\mathcal{U}$ et il étiquette un unique chemin acceptant. Soit $k_0, l_0 \geqslant n$; il existe $k_1, k_2, ...k_s$ et l_1 tels que l'unique chemin acceptant d'étiquette $\mathbf{w}_{k_0,l_0}$ dans $\mathcal{U}$ soit de la forme suivante :



Soit $K = \prod_{i \in I} k_i$. L'automate $\mathcal{U}$ étant non-ambigu, quels que soient les entiers α et β, le mot $\mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}$ est reconnu par un chemin de la même forme ; il existe donc $x = (S, w_{k_0,l_0})$, ρ et λ tels que pour tout $\alpha, \beta \in \mathbb{N}$, $(S, w_{k_0+\alpha K, l_0+\beta l_1}) = x + \alpha\rho + \beta\lambda$.

Regardons comment le poids du mot $\mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}$ est obtenu dans l'union des $\mathcal{A}_i$ et en particulier quelle est la coordonnée maximale dans $\mathcal{P}$ en suivant l'unique chemin étiqueté par ce mot. Ce chemin a la même forme que π. Pour tout β, il existe un N_β tel que, pour tout $\alpha > N_\beta$, la coordonnée maximale appartient à V et $(S, \mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}) = y + \alpha\rho_1 + \beta\lambda_1$. Ainsi, pour tout α, il existe M_α tel que, pour tout $\beta > M_\alpha$, la coordonnée maximale appartient aux coordonnées victorieuses de θ_{s+1} et $(S, \mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}) = z + \alpha\rho_2 + \beta\lambda_2$.

On a donc :

$$\forall \alpha, \beta,\ (S, \mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}) = x + \alpha\rho + \beta\lambda \tag{2.1}$$

$$\forall \beta, \forall \alpha > N_\beta,\ (S, \mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}) = y + \alpha\rho_1 + \beta\lambda_1 \tag{2.2}$$

$$\forall \alpha, \forall \beta > M_\alpha,\ (S, \mathbf{w}_{k_0+\alpha K, l_0+\beta l_1}) = z + \alpha\rho_2 + \beta\lambda_2 \tag{2.3}$$

Par conséquent, $\rho_1 = \rho = \rho_2$ et $\lambda_1 = \lambda = \lambda_2$. Il existe donc une coordonnée de V qui est victorieuse sur θ_{s+1}, ce qui contredit l'hypothèse de maximalité de s.

La propriété (**P**) est obtenue comme condition suffisante en vérifiant que l'automate $\mathcal{V}$ réalise bien la série. On montre que $\mathcal{V}$ est un automate fini en "déroulant" les circuits jusqu'à ce que la coordonnée victorieuse ne puisse plus être rattrapée par les autres ; et qu'il est non ambigu en montrant que l'ensemble des chemins réussis de $\mathcal{V}$ est en bijection avec l'ensemble des chemins réussis de $\mathcal{P}$, lui-même non ambigu. □

La dernière étape de notre transformation se fait grâce au résultat suivant qui est une adaptation d'un résultat classique de Ch. Choffrut par M. Mohri [Cho77, Cho78, Moh97, Ber79] (voir [BCPS03, WK95] pour la complexité).

Théorème 2.6. On peut décider en temps polynomial si un automate $\mathcal{A}$ non-ambigu reconnaît une série séquentielle.

On en déduit le résultat de décidabilité suivant :

[KLMP04]

Théorème 2.7. On peut décider de manière effective si une série reconnue par un automate max-plus finiment ambigu est séquentielle.

3

Liens entre séries et langages

À toute série on peut naturellement associer un langage : son support ; et à tout langage on peut naturellement associer une série : sa série caractéristique (à considérer dans notre semi-anneau préféré). Ce chapitre explore le lien qu'on peut faire entre ces langages et ces séries.

Dans la section 2, j'aborde un résultat obtenu avec Guillaume Chapuy qui concerne les supports des séries reconnaissables sur une lettre. La section 3 est une section de perspectives pour tracer un de mes futurs axes de recherche dans ce domaine.

1 Langage support et série caractéristique

A partir d'un langage, on peut construire une série naturelle qui est sa série caractéristique : c'est la série dont le support est le langage considéré et qui vaut 1 sur tous les mots de son support. Cette série dépend bien entendu du semi-anneau considéré ; cependant, quel que soit le semi-anneau, la série caractéristique d'un langage reconnaissable est reconnaissable : il suffit de mettre le neutre de la multiplication en production sur toutes les transitions d'un automate déterministe qui reconnaît le langage.

D'un autre côté, et comme on l'a vu au chapitre 1, à partir d'une série sur un semi-anneau quelconque, on construit un langage naturel qui est son support, c'est-à-dire l'ensemble des mots dont le coefficient dans la série est non nul.

Il existe des liens entre une série et son support, qui sont plus ou moins forts en fonction du semi-anneau des coefficients. Une question immédiate qui vient à l'esprit est de savoir si le support d'une série reconnaissable est reconnaissable.

La réponse à cette question est négative dans le cas général. Considérons la série sur $\mathbb{Z}$ sur un alphabet à deux lettres $\{a, b\}$ qui à un mot $\mathbf{w}$ associe la différence entre le nombre d'occurrences de la lettre a et le nombre d'occurrences de la lettre b : elle est reconnue par l'automate de la figure 3.1, mais son support est le langage non rationnel des mots qui possèdent le même nombre de a et de b.

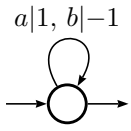


Figure 3.1 – Automate reconnaissant la série $\{a, b\} \to \mathbb{Z} : \mathbf{w} \mapsto |\mathbf{w}|_a - |\mathbf{w}|_b$.

Il existe cependant des semi-anneaux pour lesquels le support d'une série reconnaissable est toujours reconnaissable. Par exemple pour $\mathbb{R}_{\max}$ il suffit de garder l'automate booléen sous-jacent pour obtenir un automate reconnaissant le langage support. D. Kirsten a caractérisé algébriquement les semi-anneaux SR qui ont la propriété suivante : toute série reconnaissable à coefficients dans un semi-anneau SR et sur un alphabet quelconque possède un support reconnaissable [Kir09]. Il a montré qu'il suffit de regarder

les supports des séries reconnaissables sur un alphabet à deux lettres pour savoir si le semi-anneau des coefficients est SR.

Dans [Kir09], la question suivante est posée : si les séries reconnaissables sur un semi-anneau $\mathbb{S}$ et un alphabet à une lettre ont toutes des supports reconnaissables, le semi-anneau $\mathbb{S}$ est-il SR ? La réponse (négative) a cette question est en section 2 : le semi-anneau $\mathbb{Q}$ n'est pas SR (voir exemple précédent, figure 3.1, qui se plonge facilement dans $\mathbb{Q}$), mais les supports des séries reconnaissables sur $\mathbb{Q}$ et un alphabet à une lettre sont toujours reconnaissables d'après la proposition 3.1.

2 Supports de séries reconnaissables sur une lettre

Avec Guillaume Chapuy nous avons montré le résultat suivant :

[CK11]

Proposition 3.1. Le support d'une série reconnaissable sur un corps commutatif de caractéristique nulle et un alphabet à une lettre est reconnaissable.

Ce résultat ne se généralise pas pour tout corps, comme nous le montrons en section 2.2.

2.1 Cas d'un corps commutatif de caractéristique nulle

Notons qu'un automate à multiplicités sur un alphabet à une lettre $\{a\}$ est décrit par un quadruplet $\mathcal{A} = (Q, \alpha, \mu, \beta)$ où $\mu \in \mathbb{S}^{Q\times Q}$ est une matrice de transitions, les autres éléments étant semblables à la description donnée à la définition des automates à multiplicités en sous-section 3.2 (la matrice μ est en fait la matrice $\mu(a)$ telle que donnée dans cette définition).

La série reconnue par $\mathcal{A}$ s'écrit donc $S(\mathcal{A}) = \sum_{n\in\mathbb{N}} (\alpha\mu^n\beta)a^n$.

L'automate est **normalisé** s'il a un unique état initial i de poids 1, un unique état final f de poids 1 et si la i^{e} colonne et la f^{e} ligne de μ ne contiennent que des 0, c'est-à-dire si aucune transition n'arrive dans l'état initial ni ne part de l'état final.

Soit $\mathcal{A} = (Q, \alpha, \mu, \beta)$ un automate sur un corps commutatif $\mathbb{K}$ de caractéristique nulle et un alphabet à une lettre $\{a\}$. Il existe un automate $\mathcal{A}'$ équivalent normalisé sur $\mathbb{K}$ et $\{a\}$ [DKV09, Th. 2.11]. On peut donc supposer $\mathcal{A}$ normalisé sans perte de généralité.

On note respectivement i et f l'état initial et l'état final de $\mathcal{A}$: le coefficient de a^n dans la série reconnue par $\mathcal{A}$ est le coefficient (i, f) de la matrice μ^n :

$$(S(\mathcal{A}), a^n) = (\mu^n)_{i,f}.$$

On note $(u_n)_{n\in\mathbb{N}}$ la suite de ces coefficients.

Lemme 3.2. La suite $(u_n)_{n\in\mathbb{N}}$ est une suite linéaire récurrente à coefficients constants.

Démonstration. Le théorème de Cayley-Hamilton assure que toute matrice carrée sur un semi-anneau commutatif (et donc sur un corps commutatif) annule son polynôme caractéristique [AM69] : la matrice μ est donc le zéro d'un polynôme non constant à coefficient dominant 1, ce qui entraîne que

$$\mu^{|Q|} = \alpha_{|Q|-1}\mu^{|Q|-1} + \cdots + \alpha_1\mu + \alpha_0 I_{|Q|}, \qquad \text{pour certains } (\alpha_0, \ldots, \alpha_{|Q|-1}) \in \mathbb{K}^{|Q|},$$

où $I_{|Q|}$ est la matrice identité de dimension $|Q|$ et donc pour tout entier $k \geq |Q|$:

$$\mu^k = \alpha_{|Q|-1}\mu^{k-1} + \cdots + \alpha_1\mu^{k-|Q|+1} + \alpha_0\mu^{k-|Q|}.$$

En considérant les coefficients en position (i, f) de toutes ces matrices, on obtient :

$$u_k = \alpha_{|Q|-1}u_{k-1} + \cdots + \alpha_1 u_{k-|Q|+1} + \alpha_0 u_{k-|Q|}.$$

□

On applique alors un vieux résultat de Ch. Lech [Lec53, Mol89] qui stipule que dans un corps commutatif de caractéristique nulle, si les termes d'une suite linéaire récurrente à coefficients constants s'annulent infiniment souvent, alors les zéros de cette suite sont ultimement périodiques. On en déduit directement la proposition 3.1, du fait de la caractérisation des langages rationnels sur un alphabet à une lettre donnée en section 1.3.

2.2 Cas d'un corps commutatif de caractéristique p non nulle

Dans cette section nous montrons que la proposition 3.1 ne peut être étendue aux corps commutatifs de caractéristique $p \neq 0$. Plus précisément nous donnons un exemple de série reconnaissable sur un tel corps et un alphabet à une lettre dont le support n'est pas reconnaissable [CK11]. Cet exemple a été construit en adaptant l'exemple donné par Ch. Lech pour montrer que son résultat sur les suites n'est pas valide dans un corps de caractéristique $p \neq 0$.

Pour p premier, on note $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ et on considère des fractions à une indéterminée $\mathbb{F}_p(X)$: c'est le plus petit corps dans lequel s'injecte l'anneau des polynômes $\mathbb{F}_p[X]$; il est de caractéristique p.

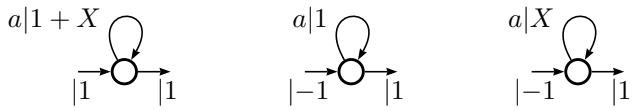


Figure 3.2 – Automate sur $\mathbb{F}p(X)$ et $\{a\}$ qui reconnaît une série dont le support n'est pas reconnaissable.

La série $S : \{a\}^* \to \mathbb{F}_p(X)$ définie par

$$(S, a^n) = (1 + X)^n - 1 - X^n$$

est reconnue par l'automate de la figure 3.2. Notons L le complémentaire de son support.

Pour tout entier n :

$$(S, a^n) = (1 + X)^n - 1 - X^n = \sum_{k=1}^{n-1} \binom{n}{k} X^k. \tag{3.1}$$

Or la plus grande puissance de p qui divise $\binom{n}{k}$ est le nombre de retenues quand on soustrait k à n en base p [Wei75]. On en déduit :

- Si n est une puissance de p, par exemple p^α, n s'écrit 10^α en base p et tout k ($1 \leq k \leq n-1$) a au moins un chiffre 1 sous un des chiffres 0 de n et donc $\binom{n}{k}$ est divisible par p et le membre de droite de l'équation (3.1) est nul car on est dans un corps de caractéristique p.
- Si n n'est pas une puissance de p, n s'écrit $1d_1 \dots d_\beta 1e_1 \dots e_\gamma$ en base p et en choisissant pour k le nombre 10^γ en base p, $\binom{n}{k}$ n'est pas divisible par p et le coefficient de X^k dans l'équation (3.1) est non nul. Le membre de droite de l'équation (3.1) est donc un polynôme non nul.

Par conséquent, le langage L est le langage des puissances de p, qui n'est pas reconnaissable, et le support de S est non reconnaissable.

Au passage, $\mathbb{F}_p(X)$ est un premier exemple de semi-anneau non-SR qui ne contient pas $\mathbb{Z}$, ce qui répond à une question formulée dans [KQ11].

3 Perspectives : problèmes de commutativité

Trouver la série maximale à coefficients réels commutant avec une série rationnelle donnée est simple : à l'instar de ce qui se passe sur les polynômes (et dans une certaine mesure sur les mots), il suffit de prendre les combinaisons linéaires finies des puissances d'une série rationnelle de base, la série de départ pouvant être obtenue de cette même façon [Reu02]. A contrario, trouver le langage maximal commutant avec un langage rationnel donné (son **centre**) est compliqué et le problème de savoir si le centre d'un langage rationnel est nécessairement rationnel est resté ouvert pendant plus de trente ans [Con71, CKO02]. En 2004, M. Kunc [Kun07] a montré qu'il existait des langages finis de centre un langage non rationnel (ni même récursivement énumérable). Ce problème de centre de langages nous fait donc sortir du domaine rationnel.

Ce problème est remarquable dans le sens où habituellement les questions que l'on peut poser sur des langages et sur des séries trouvent des solutions plus simples (et plus régulières) dans les langages que dans les séries. C'est le contraire ici.

Pour essayer de comprendre la différence de comportement des deux domaines, on peut considérer un langage L de série caractéristique S et se demander quels sont les langages qui commutent avec L et dont les séries caractéristiques commutent avec S. Même si ces deux objets semblent être les mêmes, leurs comportements vis-à-vis du produit sont tout à fait différents car, contrairement aux langages, les séries permettent de garder en mémoire le nombre de décompositions d'un mot. Par exemple :

- sur les langages : $(a + aba) \cdot (ba + 1) = a \cdot (1 + ba + baba)$,
- sur les séries (à coefficients réels) : $(a + aba) \cdot (ba + 1) \neq a \cdot (1 + ba + baba)$.

Les simulations permettent de poser la conjecture suivante :

Conjecture 1

Soit un langage rationnel L de série caractéristique S sur $\mathbb{R}$. Tout langage qui commute avec L et dont la série caractéristique commute avec S est rationnel.

7. LIST OF PROBLEMS

7.2 Groups and semigroups of automata.

1. Does there exist an algorithm that

[...]

(b) determines, by a given finite noninitial automaton A, whether the semigroup $S(A)$ (group $G(A)$) is finite, [...], free, [...]?

R.I. Grigorchuk, V.V. Nekrashevich, and V.I. Sushchanskiĭ.
Automata, dynamical systems, and groups.
Tr. Mat. Inst. Steklova, 231 :134–214, 2000.

You like au-tomato-n and I like au-tomahto-n

librement adapté de Ira et George Gershwin

Cette partie permet d'introduire les définitions et outils de base pour manipuler des (semi-)groupes d'automate ; y sont exposés également les principaux problèmes de calcul et de décision auxquels je vais m'intéresser. Elle regroupe des notions et résultats classiques, complétés par quelques autres simples introduits dans

[AKL+12] A. Akhavi, I. Klimann, S. Lombardy, J. Mairesse, and M. Picantin. On the finiteness problem for automaton (semi)groups. *Int. J. Algebra Comput.*, 22(4) :26p., 2012.

[Kli13] Ines Klimann. The finiteness of a group generated by a 2-letter invertible-reversible Mealy automaton is decidable. In *30th STACS*, volume 20 of *LIPIcs*, pages 502–513, 2013.

On pourra se référer à [BS10, BGK+08] et à leurs bibliographies pour des références classiques sur le sujet.

4

(Semi-)groupes d'automate

En informatique, les automates (booléens) servent classiquement à représenter des langages. Enrichis avec des sorties sur les transitions, ils représentent des transductions (si les sorties sont des lettres) ou des séries formelles (si elles sont numériques), comme nous l'avons vu au chapitre 1. On se contente alors de *lire* sur l'automate pour *reconnaître* des éléments ou leur associer une *sortie*. Lorsqu'entrées et sorties sur les transitions appartiennent au même ensemble, on peut composer les éléments lus sur l'automate et *engendrer* de nouveaux éléments. On obtient alors un semi-groupe, voire un groupe.

Les groupes d'automate ont été introduits dans les années 1960-1970 par des mathématiciens spécialistes de la théorie des groupes. Ils ont permis dans les années qui ont suivi de répondre à des conjectures importantes de théorie des groupes, notamment le problème de Milnor (existence de groupes à croissance intermédiaire) et de contribuer à des problèmes récurrents, tel le problème de Burnside (exemples d'automates très simples engendrant des groupes de torsion infinis finiment engendrés).

Les (semi-)groupes sont des objets potentiellement infinis. Il est naturel de chercher à les représenter par des objets finis. Un des avantages de la présentation par automates est que le problème du mot est décidable avec une telle présentation (cf. section 1) contrairement à ce qui se passe pour la présentation classique dite par générateurs et relations [EC92] (qui est cependant plus expressive).

Dans ce chapitre je donne les outils de base qui permettent de manipuler les automates pour déduire des propriétés sur les (semi-)groupes qu'ils engendrent.

1 Automates de Mealy

A partir d'ici, je redéfinis (légèrement) la notion d'automate. Un **automate** (fini, déterministe et complet) est la donnée d'un triplet

$$\big(A, \Sigma, \delta = (\delta_i : A \to A)_{i\in\Sigma}\big),$$

où
– l'**ensemble des états** A est un ensemble fini non vide,
– l'**alphabet** Σ est un ensemble fini non vide,
– les $\delta_i : A \to A$ sont les **fonctions de transition**.

Le changement par rapport à la définition donnée au chapitre 1 est qu'on ne s'intéresse plus à d'éventuels états initiaux ou finaux et qu'on impose que l'automate soit déterministe et complet.

Un **automate de Mealy** est un quadruplet

$$\big(A, \Sigma, \delta = (\delta_i : A \to A)_{i\in\Sigma}, \rho = (\rho_x : \Sigma \to \Sigma)_{x\in A}\big),$$

tel que (A, Σ, δ) et (Σ, A, ρ) sont des automates. Les applications ρ_x sont les **fonctions de production** de l'automate. Un automate de Mealy est donc un *transducteur lettre-à-lettre* séquentiel et complet (avec même alphabet d'entrée et de sortie).

Un automate de Mealy est identifié à l'ensemble de ses **transitions**

$$x \xrightarrow{i|\rho_x(i)} \delta_i(x).$$

On utilise la notation graphique usuelle des transducteurs, voir figure 4.1.

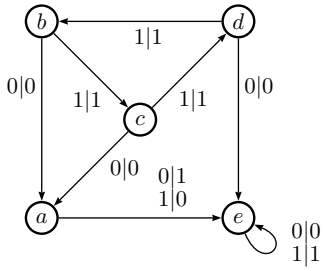


FIGURE 4.1 – Un des plus célèbres automates de Mealy : l'**automate de Grigorchuk** [GNS00] ; le groupe qu'il engendre répond à la fois au problème de Burnside et au problème de Milnor.

2 (Semi-)groupe engendré par un automate de Mealy et propriétés structurelles de certains automates

Soit $\mathcal{A} = (A, \Sigma, \delta, \rho)$ un automate de Mealy. Comme dans le cas des transducteurs, on peut considérer l'image d'un mot $\mathbf{u}$ par l'action induite par un état x, en suivant le chemin d'étiquette $\mathbf{u}$ à partir de x. Il s'agit d'un mot $\mathbf{v}$ de même longueur que $\mathbf{u}$.

Formellement, on construit les **fonctions de production étendues** $\rho_x : \Sigma^* \to \Sigma^*$ à partir des fonctions de production $\rho_x : \Sigma \to \Sigma$. Pour cela, on écrit

$$x \xrightarrow{\mathbf{u}|\mathbf{v}} y \qquad \text{avec} \qquad \mathbf{u} = u_1 \cdots u_n \quad \text{et} \quad \mathbf{v} = v_1 \cdots v_n$$

pour décrire l'existence dans $\mathcal{A}$ d'un chemin

$$x \xrightarrow{u_1|v_1} x_1 \xrightarrow{u_2|v_2} x_2 \longrightarrow \cdots \longrightarrow x_{n-1} \xrightarrow{u_n|v_n} y.$$

Par convention, l'image du mot vide est lui-même. L'application ρ_x préserve la longueur et les préfixes et satisfait

$$\forall u \in \Sigma,\ \forall \mathbf{v} \in \Sigma^*, \qquad \rho_x(u\mathbf{v}) = \rho_x(u)\rho_{\delta_u(x)}(\mathbf{v}). \tag{4.1}$$

On définit les fonctions de production étendues pour des mots sur A par simple composition :

$$\rho_{x_1 \cdots x_n} = \rho_{x_n} \circ \cdots \circ \rho_{x_1}. \tag{4.2}$$

On fait de même avec les applications $\delta_i : A^* \to A^*$:

$$\delta_{u_1 \cdots u_n} = \delta_{u_n} \circ \cdots \circ \delta_{u_1}.$$

Le **semi-groupe** $\langle \mathcal{A} \rangle_+$ **engendré** par $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est le semi-groupe des applications $\Sigma^* \to \Sigma^*$ engendré par les fonctions de production étendues ρ_x, $x \in A$. Un semi-groupe est un **semi-groupe d'automate** s'il existe un automate qui l'engendre.

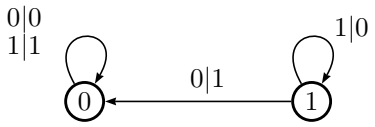


FIGURE 4.2 – Automate engendrant le semi-groupe $\mathbb{N}$ et le groupe $\mathbb{Z}$

Exemple 4.1. L'automate de la figure 4.2 engendre le semi-groupe $\mathbb{N}$: si on interprète un mot sur $\{0,1\}^*$ comme le miroir de l'écriture en base 2 d'un entier, alors ρ_0 agit sur cet entier comme l'identité et ρ_1 comme l'incrémentation.

Un automate de Mealy est **inversible** si ses fonctions de production sont des permutations de Σ. Ses fonctions de production étendues sont alors des permutations de Σ^* : elles sont inversibles et on peut donc envisager d'engendrer un groupe.

Le **groupe** $\langle\mathcal{A}\rangle$ **engendré** par un automate de Mealy inversible $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est le groupe des permutations de Σ^* engendré par les fonctions de production étendues ρ_x, $x \in A$. Un groupe est un **groupe d'automate** s'il existe un automate qui l'engendre.

Exemple 4.2. L'automate de la figure 4.2 est inversible et engendre le groupe $\mathbb{Z}$.

Un automate de Mealy est **réversible** si ses fonctions de transition sont des permutations. Le terme employé habituellement en théorie des automates est *automate à groupe* (parce que le monoïde des transitions est un groupe dans ce cas).

2.1 Plusieurs points de vue concernant l'action induite par un état

Jusqu'à présent, on a considéré qu'un état d'un automate de Mealy induit une action sur des mots finis, cette action préservant la longueur et les préfixes. De façon tout à fait naturelle, on peut également considérer cette action sur d'autres ensembles.

Notons $\mathcal{A} = (A, \Sigma, \delta, \rho)$ un automate de Mealy et soit un état $x \in A$ de cet automate.

La fonction de production étendue ρ_x agit sur les mots infinis Σ^ω : l'image d'un mot $\mathbf{u} \in \Sigma^\omega$ est la limite des images de la suite des préfixes de $\mathbf{u}$, limite correctement définie puisque ρ_x préserve les préfixes.

On peut également considérer l'ensemble des mots sur Σ comme un arbre de racine étiquetée par le mot vide et d'arité constante $\#\Sigma$; un nœud est le père d'un autre s'ils sont respectivement de la forme $\mathbf{s}$ et $\mathbf{s}i$, où $\mathbf{s} \in \Sigma^*$ est un mot et $i \in \Sigma$ une lettre. On peut alors voir ρ_x comme agissant sur cet arbre, chaque branche infinie de l'arbre correspondant à un mot infini et à la suite de ses préfixes finis.

3 Opérations sur les automates et liens entre les (semi-)groupes engendrés

3.1 Automate inverse

Soient un automate de Mealy inversible $\mathcal{A} = (A, \Sigma, \delta, \rho)$ et $A^{-1} = \{x^{-1}, x \in A\}$ une copie disjointe de son ensemble d'états. L'**automate (de Mealy) inverse** $\mathcal{A}^{-1}$ de $\mathcal{A}$ est défini par l'ensemble de transitions

$$x^{-1} \xrightarrow{j|i} y^{-1} \in \mathcal{A}^{-1} \iff x \xrightarrow{i|j} y \in \mathcal{A}. \tag{4.3}$$

La fonction de production ρ_x associée à l'état x de $\mathcal{A}$ est une bijection de Σ^* sur Σ^*, on peut donc considérer son inverse $\rho_x^{-1} : \Sigma^* \to \Sigma^*$ associée à l'état x^{-1} de $\mathcal{A}^{-1}$. On a alors

$$\langle\mathcal{A}\rangle_+ = \{\rho_{\mathbf{x}}, \mathbf{x} \in A^*\}, \qquad \langle\mathcal{A}\rangle = \{\rho_{\mathbf{x}}, \mathbf{x} \in (A \sqcup A^{-1})^*\}.$$

A noter qu'on peut toujours, à partir d'un automate de Mealy, considérer l'ensemble des transitions inverses de ses transitions (telles que définies par (4.3)). On note $\mathfrak{i}$ cette opération sur l'automate. Par $\mathfrak{i}$, on obtient toujours un transducteur lettre-à-lettre avec même alphabet d'entrée et de sortie : c'est un automate de Mealy si et seulement l'automate de départ est inversible et, dans ce cas, bien entendu : $\mathfrak{i}(\mathcal{A}) = \mathcal{A}^{-1}$.

Un automate de Mealy inversible est **biréversible** si lui et son inverse sont réversibles.

[AKL+12]

Proposition 4.1. Soit $\mathcal{A}$ un automate inversible-réversible. On a

$$\langle\mathcal{A}\rangle = \langle\mathcal{A}^{-1}\rangle = \langle\mathcal{A} \sqcup \mathcal{A}^{-1}\rangle = \langle\mathcal{A} \sqcup \mathcal{A}^{-1}\rangle_+,$$

où $\mathcal{A} \sqcup \mathcal{A}^{-1}$ est l'automate de Mealy dont l'ensemble des transitions est l'union des ensembles de transitions de $\mathcal{A}$ et $\mathcal{A}^{-1}$.
De plus, si $\langle\mathcal{A}\rangle$ ou $\langle\mathcal{A}\rangle_+$ est fini, on a

$$\langle\mathcal{A}\rangle = \langle\mathcal{A}\rangle_+.$$

Idée. Un semi-groupe fini qui se plonge dans un groupe est un groupe car tout élément est inversible et qu'il suffit d'en prendre une puissance suffisamment élevée pour obtenir l'élément neutre, en d'autres termes, dans ce cas, les poêles à frire n'ont pas de manche. □

3.2 Automate dual

La définition d'un automate de Mealy introduit une forte symétrie entre l'ensemble des états et l'alphabet de l'automate. De fait on peut inverser leurs rôles.

L'**automate dual** de $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est l'automate de Mealy $\mathfrak{d}(\mathcal{A})$ dont les transitions sont décrites par

$$i \xrightarrow{x|y} j \ \in \mathfrak{d}(\mathcal{A}) \iff x \xrightarrow{i|j} y \ \in \mathcal{A}. \tag{4.4}$$

Cette définition est consistante : le dual d'un automate de Mealy est bien toujours un automate de Mealy. Un automate de Mealy est réversible si et seulement si son dual est inversible.

La figure 4.3 représente un couple d'automates de Mealy duaux.

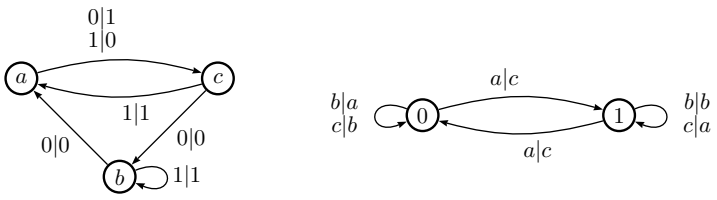


FIGURE 4.3 – L'**automate Baby Aleshin** [SVV11] et son dual.

Les propositions 4.2 et 4.3 suivantes se complètent l'une l'autre et nous donnent nos premières propriétés liées à la finitude sur les (semi-)groupes d'automate.

[AKL+12]

Proposition 4.2. Soient deux semi-groupes finis G et H. Il existe un automate de Mealy $\mathcal{A}$ tel que

$$\langle \mathcal{A} \rangle_+ = G \quad \text{et} \quad \langle \mathfrak{d}(\mathcal{A}) \rangle_+ = H.$$

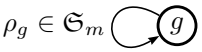
On a un énoncé similaire sur les groupes.

Idée. La preuve est faite dans le cadre des groupes. Elle est similaire pour les semi-groupes.

On construit en parallèle l'automate $\mathcal{A}$ qui engendre G et l'automate $\mathfrak{d}(\mathcal{A})$ qui engendre H. On procède par étape en s'assurant à chaque instant que $\mathcal{A}$ et $\mathfrak{d}(\mathcal{A})$ sont duaux, que $\mathcal{A}$ est bien un automate de Mealy inversible et que $\mathcal{A}$ engendre G et $\mathfrak{d}(\mathcal{A})$ engendre H.

Notons $\mathfrak{S}_k$ l'ensemble des permutations des entiers de 1 à k. Le groupe G étant fini, il est isomorphe à un sous-groupe de $\mathfrak{S}_m$ pour un certain m. De même, le groupe H est isomorphe à un sous-groupe de $\mathfrak{S}_n$ pour un certain n.

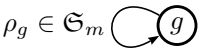
$\mathcal{A}$

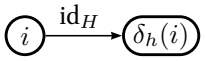


$\mathfrak{d}(\mathcal{A})$

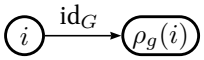
Par dualisation on obtient :

$\mathcal{A}$



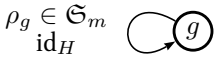


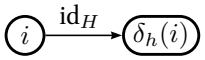
$\mathfrak{d}(\mathcal{A})$



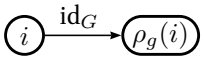
Les états de $\mathcal{A}$ agissent *tous* sur les éléments de H et de même les états de $\mathfrak{d}(\mathcal{A})$ sur les éléments de G, donc :

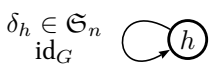
$\mathcal{A}$



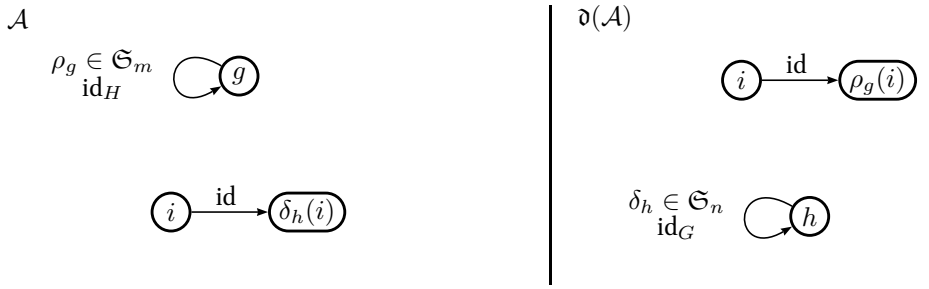


$\mathfrak{d}(\mathcal{A})$





Les états de $\mathcal{A}$ agissent *tous* sur les éléments de $\{1, \ldots, m\}$ et de même du côté du dual, donc :



□

Proposition 4.3 ([Neko5, SV11, AKL$^+$12])**.** Le (semi-)groupe engendré par $\mathcal{A}$ est fini si et seulement si le (semi-)groupe engendré par son dual $\mathfrak{d}(\mathcal{A})$ est fini.

Démonstration. Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$. On suppose que le (semi-)groupe engendré par son dual est fini.

Fixons un mot $\mathbf{w} \in A^*$. On a :

$$\rho_{\mathbf{w}}(u_1u_2\cdots u_n) := \rho_{\mathbf{w}}(u_1)\rho_{\delta_{u_1}(\mathbf{w})}(u_2)\rho_{\delta_{u_1u_2}(\mathbf{w})}(u_3)\cdots\rho_{\delta_{u_1u_2\cdots u_{n-1}}(\mathbf{w})}(u_n),$$

pour tout $u_1u_2\cdots u_n \in \Sigma^n$. Cette formule s'obtient (de façon fastidieuse) par le calcul à partir des formules (4.1) et (4.2), elle est illustré de façon plus intuitive en figure 4.5.

La fonction de production $\rho_{\mathbf{w}}$ peut donc être vue comme la fonction de production d'un transducteur lettre-à-lettre sur le graphe de Cayley de $\langle\mathfrak{d}(\mathcal{A})\rangle_+$:

Il n'y a qu'un nombre fini de tels étiquetages. □

3.3 Automates étendus

Soit $\mathcal{A}$ un automate inversible-réversible. On a vu en proposition 4.1 que $\langle\mathcal{A}\rangle = \langle\mathcal{A} \sqcup \mathcal{A}^{-1}\rangle$, c'est-à-dire qu'on ne modifie pas le groupe engendré en considérant les états et leurs inverses. On peut de même considérer les lettres *et leurs inverses*. L'**automate étendu** $\tilde{\mathcal{A}}$ de $\mathcal{A}$ est son extension à l'ensemble d'états $A \sqcup A^{-1}$ et à l'alphabet $\Sigma \sqcup \Sigma^{-1}$:

$$\tilde{\mathcal{A}} = \mathcal{A}' \sqcup (\mathcal{A}')^{-1} \qquad \text{où} \quad \mathcal{A}' = \mathfrak{d}(\mathfrak{d}(\mathcal{A}) \sqcup \mathfrak{d}(\mathcal{A})^{-1}).$$

Graphiquement :

$$x \xrightarrow{i|j} y$$

est une transition de $\mathcal{A}$ si et seulement si

$$x \xrightarrow{i|j} y \xrightarrow{i^{-1}|j^{-1}} x \quad \text{et} \quad x^{-1} \xrightarrow{j|i} y^{-1} \xrightarrow{j^{-1}|i^{-1}} x^{-1}$$

sont des transitions de $\mathcal{A}'$.

Le corollaire suivant est une conséquence des propositions 4.1 et 4.3.

[AKL$^+$12]

Corollaire 4.4. Soit $\mathcal{A}$ un automate inversible-réversible. Les groupes $\langle\mathcal{A}\rangle$ et $\langle\tilde{\mathcal{A}}\rangle$ sont tous deux finis ou tous deux infinis.

À noter que ces deux groupes ne sont pas nécessairement isomorphes. Par exemple si on considère l'automate de la figure 4.4, il engendre le groupe $K_4 \rtimes \mathbb{Z}_2$ d'ordre 16 et son automate étendu engendre un groupe d'ordre 64.

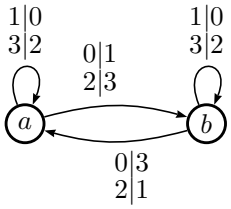


FIGURE 4.4 – Automate de Mealy engendrant un grouped'ordre 16.

3.4 Automates d'ordres supérieurs

Soient un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ et deux entiers $n, k > 0$.

L'automate de Mealy

$$\mathcal{A}_{n,k} = \left(A^n, \Sigma^k, (\delta_{\mathbf{u}} : A^n \to A^n)_{\mathbf{u}\in\Sigma^k}, (\rho_{\mathbf{x}} : \Sigma^k \to \Sigma^k)_{\mathbf{x}\in A^n} \right) \tag{4.5}$$

est l'**automate de Mealy d'ordre** (n, k) **associé à** $\mathcal{A}$.

Dans (4.5), $\rho_{\mathbf{x}} : \Sigma^n \to \Sigma^n$ est la restriction de $\rho_{\mathbf{x}} : \Sigma^* \to \Sigma^*$ à Σ^n, et de même pour $\delta_{\mathbf{u}} : \delta_{\mathbf{u}} : A^n \to A^n$ est la restriction de $\delta_{\mathbf{u}} : A^* \to A^*$ à A^n. On a en particulier $\mathcal{A}_{1,1} = \mathcal{A}$. Graphiquement, $\mathcal{A}_{n,k}$ est un automate dont les états sont les mots de A^n et les actions de ces états correspondent aux actions des éléments du semi-groupe $\langle\mathcal{A}\rangle_+$ sur des mots de Σ^k.

Le semi-groupe engendré par l'automate d'ordre $(n, 1)$ associé à $\mathcal{A}$ est un sous-semi-groupe de $\langle\mathcal{A}\rangle_+$. Le semi-groupe engendré par l'automate d'ordre $(1, k)$ associé à $\mathcal{A}$ est isomorphe à $\langle\mathcal{A}\rangle_+$. Le semi-groupe engendré par l'automate d'ordre (n, k) associé à $\mathcal{A}$ est donc isomorphe à un sous-semi-groupe de $\langle\mathcal{A}\rangle_+$.

L'automate $\mathcal{A}_{n,1}$ pourra aussi être noté $\mathcal{A}^n$ et appelé **puissance** n**-ème** de $\mathcal{A}$.

Maintenant que nous avons introduit ce formalisme, nous pouvons retrouver graphiquement la formule donnée par le calcul dans la démonstration de la proposition 4.3 :

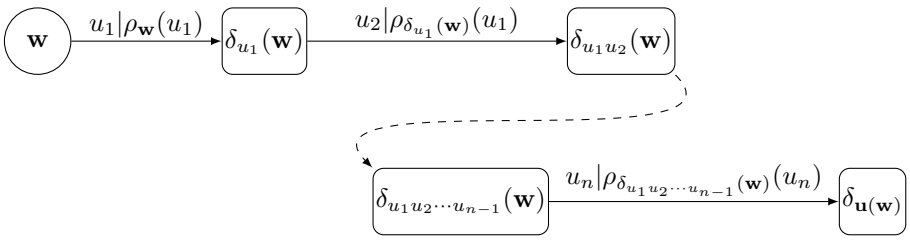


FIGURE 4.5 – Action induite par un mot $\mathbf{w} \in A^*$ sur un mot $\mathbf{u} = u_1 \cdots u_n \in \Sigma^*$.

3.5 Minimisation

Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$. Une équivalence $\equiv$ sur A est une **congruence** pour $\mathcal{A}$ si

$$\forall x, y \in A,\ \Big([x \equiv y] \Longrightarrow [\forall i \in \Sigma,\ \rho_x(i) = \rho_y(i) \text{ et } \delta_i(x) \equiv \delta_i(y)]\Big).$$

L'**équivalence de Nerode** sur A est la congruence la plus grossière pour $\mathcal{A}$. Elle est la limite de la suite d'équivalences de plus en plus fines $(\equiv_k)$ définie par

$$\begin{aligned} \forall x, y \in A, \qquad & x \equiv_0 y \Longleftrightarrow \forall i \in \Sigma,\ \rho_x(i) = \rho_y(i), \\ & \forall k \geqslant 0, x \equiv_{k+1} y \Longleftrightarrow x \equiv_k y \text{ et } \forall i \in \Sigma,\ \delta_i(x) \equiv_k \delta_i(y). \end{aligned}$$

L'ensemble d'états A étant fini, cette suite est ultimement constante ; de plus, elle est constante dès que deux termes consécutifs sont égaux. Sa limite est donc calculable.

La définition est consistante avec la définition de minimisation sur les automates booléens (voir p. 14) : la partition initiale se fait ici relativement aux fonctions de production et non sur le critère états finaux / états non finaux.

On note $[x]$ la classe d'équivalence d'un état $x \in A$ pour l'équivalence de Nerode et $[x]_k$ sa classe d'équivalence pour $\equiv_k$, appelée la k-**classe** de x.

Remarque 4.5. Deux états appartiennent à la même k-classe s'ils agissent de la même manière sur les mots de longueur au plus k et à la même classe de Nerode s'ils agissent de la même manière sur l'ensemble des mots.

Soient un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ et $\equiv$ l'équivalence de Nerode associée à $\mathcal{A}$. Le **minimisé** de $\mathcal{A}$ est l'automate de Mealy $\mathcal{A}/\equiv\ = (A/\equiv, \Sigma, \tilde{\delta}, \tilde{\rho})$, où, pour tout état $x \in A$ et toute lettre $i \in \Sigma$, on a :

$$\tilde{\delta}_i([x]) = [\delta_i(x)] \quad \text{et} \quad \tilde{\rho}_{[x]}(i) = \rho_x(i).$$

Un automate est **minimal** s'il est isomorphe à son minimisé.

Corollaire 4.6. Un automate de Mealy et son minimisé engendrent le même semi-groupe.

Comme on l'a vu à la remarque 4.5, deux états sont équivalents si leurs fonctions de production étendues ont la même action sur Σ^*. Par extension, deux mots $\mathbf{x}, \mathbf{y} \in A^*$ sont **équivalents** s'ils ont la même fonction de production étendue. On note $[\![\mathbf{x}]\!]$ l'ensemble des mots de A^* équivalents à $\mathbf{x}$, indépendamment de leur longueur.

3.6 Clôture tensorielle

On sait que si un automate engendre un semi-groupe fini, il en va de même de son dual (proposition 4.3). On peut alors modifier l'alphabet de façon à ce que chaque élément du semi-groupe engendré par le dual ait un unique représentant.

Soit $\mathcal{A} = (A, \Sigma, \delta, \rho)$ un automate de Mealy qui engendre un semi-groupe fini. Sa **clôture tensorielle** est l'automate de Mealy $\mathfrak{c}(\mathcal{A}) = (A, \Xi, \bar{\delta}, \bar{\rho})$, où $\Xi = \{[\![\mathbf{s}]\!] \mid \mathbf{s} \in \Sigma^*\} = \langle \mathfrak{d}(\mathcal{A}) \rangle_+$ et $\bar{\delta}$ et $\bar{\rho}$ sont les extensions naturelles de δ et ρ :

$$\forall x \in A, \forall \mathbf{s} \in \Sigma^*,\ \bar{\delta}_{[\![\mathbf{s}]\!]}(x) = \delta_{\mathbf{s}}(x) \text{ et } \bar{\rho}_x([\![\mathbf{s}]\!]) = [\![\rho_x(\mathbf{s})]\!].$$

Un automate de Mealy est **tensoriellement clos** s'il est isomorphe à sa propre clôture. Dans ce cas son dual est minimal.

Remarque 4.7. La clôture tensorielle d'un automate de Mealy n'engendre pas nécessairement le même (semi-)groupe que l'automate de départ, en revanche leurs duaux respectifs engendrent le même (semi-)groupe, ce qui assure que l'opération de clôture tensorielle ne modifie pas le caractère fini ou non du semi-groupe engendré.

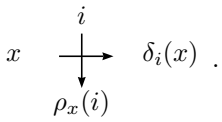
4 Diverses représentations d'un automate de Mealy

Un automate de Mealy est la donnée d'un ensemble de transitions étiquetées. Jusqu'à présent, nous avons représenté cet ensemble sous la forme d'un automate ou de son dual. On peut donner diverses autres représentations de cet ensemble, traduisant différentes visions que l'on peut en avoir.

Certaines de ses représentations visent à accentuer la symétrie entre un automate de Mealy et son dual (diagrammes en croix et graphes en hélice). D'autres permettent de considérer les actions des éléments du (semi-)groupe sur l'arbre des mots (portraits).

4.1 Diagrammes en croix

La transition $x \xrightarrow{i|\rho_x(i)} \delta_i(x)$ est notée

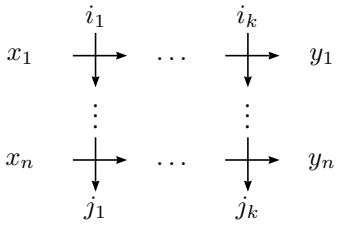


.

Cette notation est appelée **transition en croix**.

Un chemin dans un automate de Mealy $\mathcal{A}$ (*resp.* dans son dual $\mathfrak{d}(\mathcal{A})$) peut être représenté par un **diagramme en croix** horizontal (*resp.* vertical). On peut également considérer des diagrammes en croix rectangulaires de dimension $n \times k$ sur lesquels on peut lire les fonctions de production de l'automate associé d'ordre (n, k) et de son dual.

Par exemple, le diagramme en croix suivant :



correspond dans $\mathcal{A}_{n,k}$ à

$$\rho_{x_1 \cdots x_n}(i_1 \cdots i_k) = j_1 \cdots j_k,$$
$$\delta_{i_1 \cdots i_k}(x_1 \cdots x_n) = y_1 \cdots y_n.$$

4.2 Graphes en hélice

On appelle **graphe en hélice** d'un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ le graphe de sommets $A \times \Sigma$ et d'arcs les $(x, i) \to (\delta_i(x), \rho_x(i))$. Le graphe en hélice de l'automate Baby Aleshin est représenté en figure 4.6.

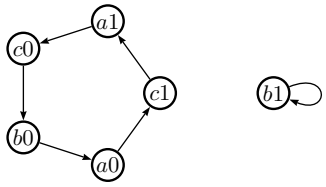


FIGURE 4.6 – Le graphe en hélice de l'automate Baby Aleshin (voir l'automate en figure 4.3).

On peut définir un graphe en hélice pour tout transducteur lettre-à-lettre d'alphabets d'entrée et de sortie égaux. Un tel transducteur est un automate de Mealy si et seulement si de tout sommet de son graphe en hélice part un unique arc.

Soient un automate de Mealy $\mathcal{A}$ et deux entiers $n, k > 0$. Le **graphe en hélice d'ordre** (n, k) de $\mathcal{A}$ est le graphe en hélice de l'automate de Mealy d'ordre (n, k) associé à $\mathcal{A}$. Le graphe en hélice d'ordre $(1, 1)$ de $\mathcal{A}$ est le graphe en hélice de $\mathcal{A}$.

On parle *des* graphes en hélice de $\mathcal{A}$ pour désigner l'ensemble de ses graphes en hélice d'ordre quelconque.

4.3 Portraits

La notion de **portrait d'un automorphisme d'arbre** vient de la théorie géométrique des groupes. Elle est présentée en détail dans [Nek05].

Les portraits sont une autre façon de considérer les automates et les actions qu'ils induisent. Ils permettent notamment de considérer naturellement l'action d'un état sur la n-ème lettre de tous les mots, n étant fixé. Ils nous seront utiles pour montrer des résultats au chapitre 7.

On considère l'ensemble des mots sur Σ comme un arbre enraciné sur le mot vide et d'arité constante (comme décrit en sous-section 2.1).

Le langage Σ^n des mots de longueur n est le **niveau** de profondeur n de cet arbre. Une **branche** de cet arbre est une suite de mots $(\mathbf{s}_k)_{k\in\mathbb{N}}$ telle que, pour tout k, $\mathbf{s}_k$ est le préfixe de longueur k de $\mathbf{s}_{k+1}$.

Un automorphisme de l'arbre Σ^* est une application bijective $\Sigma^* \to \Sigma^*$ qui préserve la racine et les arêtes de l'arbre. L'automate $\mathcal{A}$ agit sur l'arbre enraciné Σ^*.

Soient g un automorphisme de Σ^* et $\mathbf{s} \in \Sigma^*$ un mot. On peut définir de manière unique un automorphisme $g_{|\mathbf{s}} : \Sigma^* \to \Sigma^*$ appelé **section de** g par

$$\forall \mathbf{t} \in \Sigma^*, \quad g(\mathbf{st}) = g(\mathbf{s})g_{|\mathbf{s}}(\mathbf{t}).$$

Le **portrait** de g est l'étiquetage de l'arbre Σ^* dans lequel le sommet $\mathbf{s} \in \Sigma^*$ est étiqueté par $g_{|\mathbf{s}} : \Sigma \to \Sigma$. Il est noté $\mathfrak{p}_\infty(g)$. Un niveau (*resp.* une branche) d'un portrait est le niveau (*resp.* la branche) correspondant(e) de l'arbre étiqueté.

Pour un entier k donné, le k-**portrait** $\mathfrak{p}_k(g)$ de g est la restriction de $\mathfrak{p}_\infty(g)$ aux niveaux 0 à $k-1$: il représente l'action de g sur les mots de longueur au plus k.

Soit $\mathcal{A} = (A, \Sigma, \delta, \rho)$ un automate de Mealy inversible et un mot $\mathbf{u} \in A^*$.

Le **portrait** de $\mathbf{u}$ (ou ∞-**portrait** — *resp.* k-**portrait**) est le portrait (*resp.* le k-portrait) de $\rho_{\mathbf{u}}$: chaque sommet de $\mathbf{s} \in \Sigma^*$ est étiquetée par $\rho_{\delta_{\mathbf{s}}(\mathbf{u})} : \Sigma \to \Sigma$. Il est noté $\mathfrak{p}_\infty[\![\mathbf{u}]\!]$ (*resp.* $\mathfrak{p}_k[\![\mathbf{u}]\!]$). Cette notation est justifiée par le fait que deux mots équivalents ont même fonction de production étendue. Un exemple est donnée en figure 4.7.

(a) L'**automate d'Aleshin**, (b) un de ses portraits : $\mathfrak{p}_3[\![c]\!]$.

FIGURE 4.7 – Un portrait de l'automate d'Aleshin ; $\mathrm{id} = \mathrm{id}_\Sigma$ et σ permute 0 et 1.

L'application qui à un automorphisme d'arbre associe un portrait induit une structure de monoïde sur l'ensemble des portraits dont l'élément neutre du produit est le **portrait identité** : $\mathcal{I}_\infty = \mathfrak{p}_\infty(\mathrm{id}_{\Sigma^*})$. Les **portraits d'un automate** $\mathcal{A}$ sont les portraits des éléments de $\langle\mathcal{A}\rangle_+$. Le produit de deux k-portraits de $\mathcal{A}$ s'exprime en terme de mots : $\mathfrak{p}_k[\![\mathbf{u}]\!]\mathfrak{p}_k[\![\mathbf{v}]\!] = \mathfrak{p}_k[\![\mathbf{uv}]\!]$. Il permet de munir l'ensemble des k-portraits de $\mathcal{A}$ d'une structure de monoïde, dont l'élément neutre est le k-**portrait identité** $\mathcal{I}_k = \mathfrak{p}_k(\mathrm{id}_{\Sigma^*})$.

Un niveau d'un portrait est **homogène** si tous ses sommets ont même étiquette. Un portrait est **homogène** si tous ses niveaux sont homogènes. Le portrait de la figure 4.7(b) n'est pas homogène, mais ses niveaux 0 et 1 le sont. Un k-portrait $\mathfrak{p}_k(g)$ est **presqu'homogène** si $\mathfrak{p}_{k-1}(g)$ et tous les $\big(\mathfrak{p}_{k-1}(g_{|i})\big)_{i\in\Sigma}$ sont homogènes.

La donnée d'un k-portrait homogène $\mathcal{J}$ et d'un $\#\Sigma$-uplet $\tau = (\tau_i)_{i\in\Sigma}$ de permutations de Σ permet de construire un $(k+1)$-portrait $\mathcal{K}$ presqu'homogène de la manière suivante : $\mathcal{J}$ est la restriction de $\mathcal{K}$ aux niveaux 0 à $k-1$ et les feuilles du sous-arbre correspondant à la lettre $i \in \Sigma$ sont toutes étiquetées par τ_i. Ce portrait est noté $\mathcal{J}\lfloor\tau\rfloor$. Un exemple d'une telle construction est donné figure 4.8.

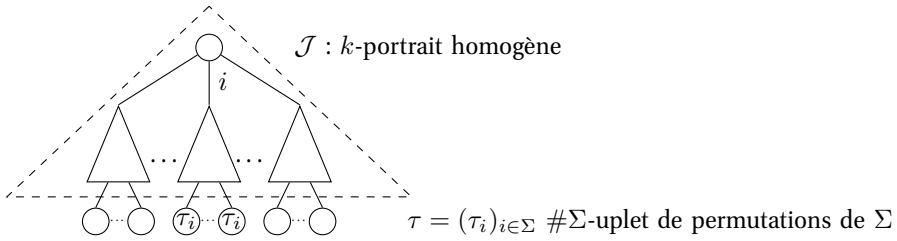


FIGURE 4.8 – Le $(k+1)$-portrait presqu'homogène $\mathcal{J}\lfloor\tau\rfloor$, $\tau = (\tau_i)_{i\in\Sigma}$.

Remarque 4.8. Le produit de deux k-portraits homogènes est un k-portrait homogène.

De plus, si l'alphabet est réduit à deux lettres $\{x, y\}$:
- le carré d'un k-portrait homogène est le k-portrait identité ;
- le carré d'un k-portrait presqu'homogène dont la racine est l'identité est le k-portrait identité ;
- le carré d'un k-portrait $\mathcal{J}\lfloor\tau_x, \tau_y\rfloor$ presqu'homogène dont la racine permute les deux lettres de l'alphabet est l'identité si et seulement si $\tau_x = \tau_y$.

Finitude et autres problèmes

Dans ce chapitre nous listons les problèmes auxquels nous allons nous intéresser dans la suite du manuscrit, aussi bien d'un point de vue théorique que pratique.

1 Problème du mot

Un des premiers problèmes de décision qu'on aborde avec les (semi-)groupes est le *problème du mot* : peut-on décider si deux mots représentent le même élément du (semi-)groupe ? Ce problème est en général indécidable [Nov55]. Il est cependant décidable dans le cadre des semi-groupes d'automate, comme montré en proposition 5.1.

Proposition 5.1. Le problème du mot est décidable pour les (semi-)groupes d'automate.

Démonstration. Deux états ont même fonction de production étendue si et seulement s'ils sont équivalents.

Soit un automate de Mealy $\mathcal{A}$ sur l'alphabet Σ. Si l'identité de Σ^* n'est pas une des fonctions de production étendues des états de $\mathcal{A}$, on peut ajouter un état qui boucle sur lui-même et dont la fonction de production est l'identité de Σ. Sans perte de généralité on peut donc supposer qu'un des générateurs du (semi-)groupe est l'identité, ce qui permet de considérer le problème du mot sur des mots de même longueur. Le problème revient alors à décider si deux états de $\mathcal{A}^n$ sont équivalents, pour n la longueur commune de ces deux mots. □

Même pour de petits automates, les (semi-)groupes engendrés peuvent être grands. Par exemple l'automate à 2 états et 3 lettres de la figure 5.1 engendre un semi-groupe d'ordre 13 597. Plus impressionnant encore, l'automate de la figure 5.2 engendre un groupe d'ordre $1\,494\,186\,269\,970\,473\,680\,896 = 2^{64} \cdot 3^4 \approx 1.5 \times 10^{21}$. Il n'est pas étonnant que les seuls facteurs premier de l'ordre de ce groupe soient 2 et 3 ; en effet les facteurs de l'ordre d'un groupe fini engendré par un automate sur n lettres sont inférieurs au sens large à n (conséquence immédiate de la proposition 9.2).

Cet exemple illustre la complexité des (semi-)groupes engendrés par automate et laisse entrevoir les difficultés à les étudier.

2 Croissance, ordre et finitude

Les questions que nous nous posons dans cette partie sont étroitement liées :
- le (semi-)groupe engendré est-il fini ?
- quel est son ordre ?
- quel est l'ordre d'un certain élément du (semi-)groupe ?

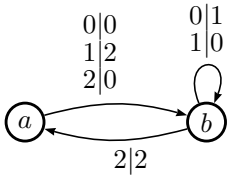


FIGURE 5.1 – Automate de Mealy engendrant un semi-groupe d'ordre 13 597.

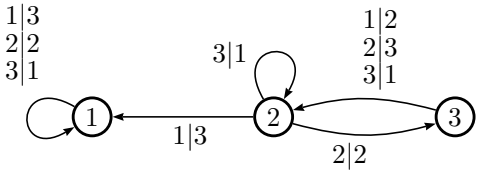


FIGURE 5.2 – Automate de Mealy engendrant un groupe d'ordre 1 494 186 269 970 473 680 896.

– à quelle vitesse le semi-groupe croît-il ?

Le problème du mot étant décidable, le problème de finitude est semi-décidable par énumération et l'ordre et la fonction de croissance d'un (semi-)groupe fini sont calculables.

Dans la suite, nous essaierons d'apporter des éléments de réponse théoriques aux questions de décidabilité (chapitres 6 et 7) et des éléments plus pratiques concernant la vitesse de décision ou de semi-décision (chapitre 8).

Vers l'infini et en deçà

librement adapté de Buzz l'éclair, Toy Story

Cette partie expose les apports des articles suivants concernant le problème de décision de la finitude d'un (semi-)groupe d'automate :

[AKL+12] A. Akhavi, I. Klimann, S. Lombardy, J. Mairesse, and M. Picantin. On the finiteness problem for automaton (semi)groups. *Int. J. Algebra Comput.*, 22(4) :26p., 2012.

[Kli13] Ines Klimann. The finiteness of a group generated by a 2-letter invertible-reversible Mealy automaton is decidable. In *30th STACS*, volume 20 of *LIPIcs*, pages 502–513, 2013.

6

Tester la finitude et l'infinitude

R. Grigorchuk, V. Nekrashevych et V. Sushchanskii ont posé la question de la décidabilité de la finitude pour les (semi-)groupes d'automates [GNS00, Problème 7.2.1(b)]. Plusieurs travaux ont apporté des éléments de réponse à cette question [Cai09, Mal09, Min09, SS05, Ant08, BBSZss, Sid00, AKL$^+$12, KMP12, Kli13]. Très récemment, P. Gillibert a montré que la finitude des semi-groupes d'automate est indécidable [Gil13]. La question de la finitude des groupes d'automate reste cependant toujours ouverte. Dans ce chapitre, je commence par traiter des tests de finitude – tests existants (§1), puis tests liés à mes travaux (§2), je suis ensuite le même schéma pour les tests d'infinitude (§3 et §4). La section 5 montre les gains apportés par ces nouveaux tests. Enfin la section 6 donne un critère non effectif de finitude.

A noter que les tests de finitude qui reposent sur l'énumération des éléments ne sont pas abordés ici, mais au chapitre 8.

1 Finitude - l'existant

Il s'agit ici de parcourir, de façon tout à fait superficielle et sans aucune démonstration, les divers tests de finitude apparaissant dans la littérature.

1.1 Machines de Cayley

Soit S un semi-groupe fini. Le **graphe de Cayley** de S est le graphe d'ensemble de sommets S et d'ensemble d'arêtes

$$\{s \xrightarrow{t} st \mid s, t \in S\}.$$

A partir de ce graphe de Cayley, on construit naturellement deux automates de Mealy en ajoutant des sorties sur les transitions :
– $\mathcal{C}(S)$: la **machine de Cayley de** S dont l'ensemble des transitions est

$$\{s \xrightarrow{t|st} st \mid s, t \in S\}.$$

– $\mathcal{C}^*(S)$: la **machine de Cayley duale de** S (qui n'est pas l'automate dual de la machine de Cayley de S) dont l'ensemble des transitions est

$$\{s \xrightarrow{t|ts} st \mid s, t \in S\}.$$

Le résultat suivant a été démontré de plusieurs façons par diverses personnes :

Théorème 6.1 ([Min09, SS05, Cai09])**.** Soit S un semi-groupe fini. Le semi-groupe engendré par $\mathcal{C}(S)$ est fini si et seulement si S est $\mathcal{H}$-trivial.

La $\mathcal{H}$-trivialité peut être définie de plusieurs façons, disons simplement qu'elle équivaut pour un semi-groupe fini au fait d'être apériodique : $\forall s \in S, \exists n \mid x^{n+1} = x^n$.

Un résultat similaire pour la machine de Cayley duale :

Théorème 6.2 ([Mino9])**.** Soit S un semi-groupe fini. Le semi-groupe engendré par $\mathcal{C}^*(S)$ est fini si et seulement si S est $\mathcal{H}$-trivial et n'admet pas de sous-semi-groupe non-trivial absorbant à droite.

Ces résultats concernent des automates de Mealy ayant une structure extrêmement particulière et dont la proportion est faible à nombres de lettres et d'états fixés.

1.2 Branchement limité

A. Antonenko s'est intéressé au problème suivant : quels sont les automates de Mealy tels que quel que soit le choix des fonctions de production, le semi-groupe engendré est fini ? Les critères développés dans [Anto8] reposent sur la structure de l'automate.

Dans un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$, l'état $x \in A$ est **sans branchement** si son image par une fonction de transition ne dépend pas de la lettre lue, c'est-à-dire :

$$\forall i, j \in \Sigma, \quad \delta_i(x) = \delta_j(x).$$

Graphiquement cela signifie qu'une seule flèche part de l'état x, étiquetée par toutes les lettres de l'alphabet Σ.

Un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est **sans branchement** si tous ses états sont sans branchement.

Proposition 6.3 ([Anto8])**.** Un automate de Mealy sans branchement engendre un semi-groupe fini.

Le résultat est immédiat par le corollaire 6.6 donné plus loin (ce n'est pas la démonstration donnée dans [Anto8]).

Le résultat de la proposition 6.3 s'étend. Un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est **à branchement limité** si tous ses états qui appartiennent à un cycle sont sans branchement.

Théorème 6.4 ([Anto8])**.** Un automate de Mealy à branchement limité engendre un semi-groupe fini.

Bien entendu ce critère ne couvre pas tous les automates engendrant des semi-groupes finis et ne concerne que des automates ayant une structure très particulière. Par exemple, l'automate de la figure 5.2 n'est pas à branchement limité et engendre un semi-groupe fini.

Ce critère est maximal dans le sens où pour tout automate qui n'est pas à branchement limité, il existe un choix de fonctions de production tel que le semi-groupe engendré est infini [Anto8].

Une des conséquences du travail d'A. Antonenko est que l'on peut élaguer un automate en supprimant les états non accessibles à partir d'un cycle, sans modifier le caractère fini ou infini du semi-groupe engendré [Anto8].

2 Finitude - md-réduction

On construit ici un critère reposant sur la notion de minimisation d'un automate. Remarquant que le dual d'un minimisé n'est pas nécessairement minimal, on introduit une notion de minimalité symétrique entre un automate et son dual.

Une paire d'automates duaux est **md-réduite** si chacun des deux automates de la paire est minimal. Par extension, on dira qu'un automate est **md-réduit** si la paire qu'il forme avec son dual est md-réduite. La **md-réduction** d'une paire d'automates duaux consiste à réduire alternativement chacun des deux automates jusqu'à ce que la paire soit md-réduite.

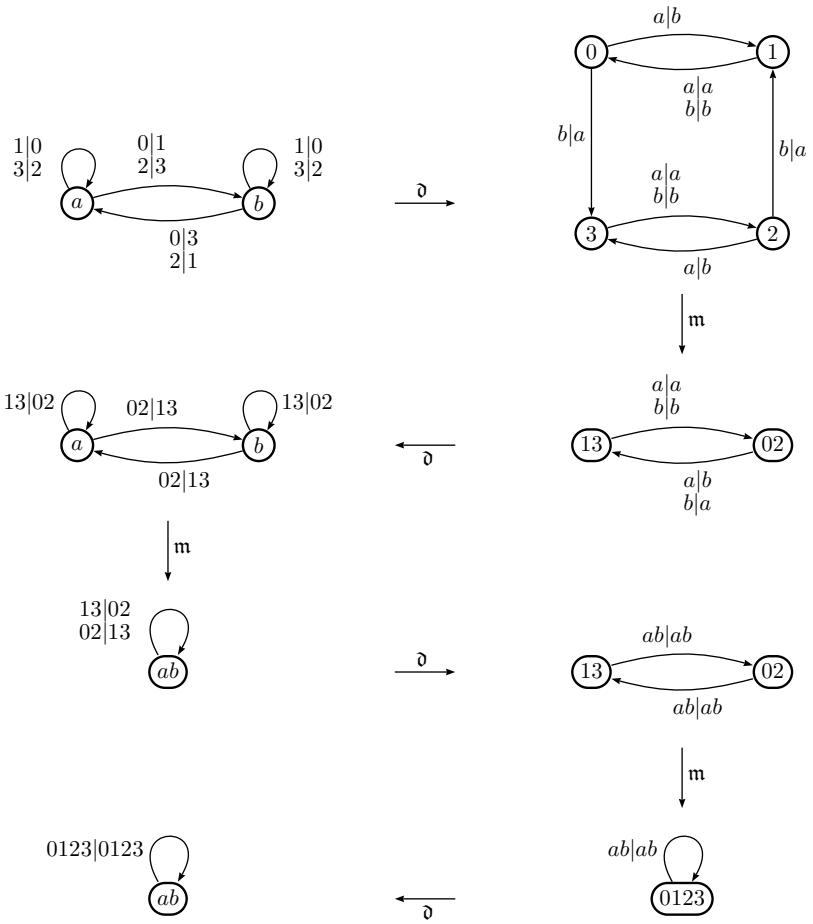


FIGURE 6.1 – La $\mathfrak{md}$-réduction d'une paire d'automates de Mealy duaux.

Exemple 6.1. Un exemple de $\mathfrak{md}$-réduction est donné en figure 6.1.

La $\mathfrak{md}$-réduction est confluente [AKL+12]. Ce fait n'est pas crucial pour la suite, mais facilitera les tournures de phrase en nous permettant de donner la définition suivante : la paire d'automates de Mealy obtenue par $\mathfrak{md}$-réduction d'un couple d'automates duaux est appelée son **$\mathfrak{md}$-réduit**.

[AKL+12]
Théorème 6.5. Une paire d'automates duaux engendre des (semi-)groupes finis si et seulement si son $\mathfrak{md}$-réduit engendre des (semi-)groupes finis.

Démonstration. La $\mathfrak{md}$-réduction est une suite de minimisations et de dualisations, aucune de ces deux opérations ne modifie le caractère fini ou infini du (semi-)groupe engendré. □

Un automate de Mealy est **$\mathfrak{md}$-trivial** si sa $\mathfrak{md}$-réduction aboutit à un automate trivial.

On déduit du théorème 6.5 la condition suffisante de finitude effective suivante, en remarquant que l'automate trivial engendre le groupe trivial.

[AKL+12]
Corollaire 6.6. Les automates de Mealy $\mathfrak{md}$-triviaux engendrent des (semi-)groupes finis.

Il existe des automates $\mathfrak{md}$-réduits non triviaux qui engendrent des (semi-)groupes finis. Des exemples sont donnés en figures 6.2 et 6.3.

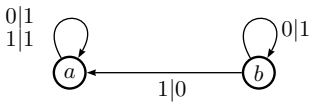


FIGURE 6.2 – Automate $\mathfrak{md}$-réduit non trivial qui engendre un semi-groupe de taille 6.

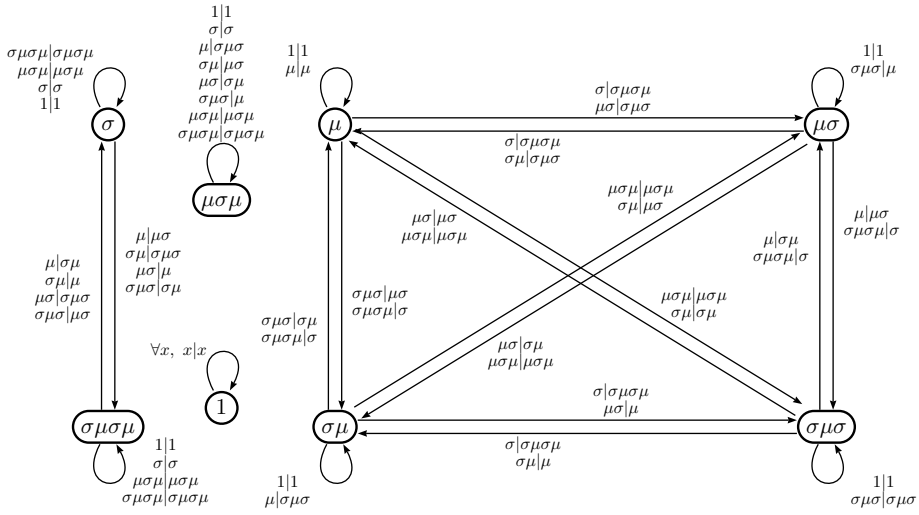


FIGURE 6.3 – Automate de Mealy biréversible $\mathfrak{md}$-réduit non trivial qui engendre le groupe fini D_4 (le plus petit groupe engendré par un automate biréversible contre-exemple pour la réciproque du corollaire 6.6). L'ensemble détat et l'alphabet de cet automate sont égaux à $\{1, \sigma, \mu, \sigma\mu, \mu\sigma, \sigma\mu\sigma, \mu\sigma\mu, \sigma\mu\sigma\mu\}$.

3 Infinitude - l'existant

Un puissant résultat de théorie des groupes permet d'affirmer que si un (semi-)groupe d'automate est d'ordre infini, soit il possède un élément d'ordre infini, soit les ordres de ses éléments ne sont pas bornés [VL93, Zel90, Zel91].

3.1 Transitivité par niveau

Un élément d'un groupe (*resp.* un groupe) agissant sur un arbre enraciné est **transitif par niveau** s'il agit transitivement sur chaque niveau de cet arbre, c'est-à-dire si son action sur ce niveau ne possède qu'une seule orbite.

Un élément transitif par niveau est d'ordre infini puisque pour tout entier n, il possède une orbite de taille $\#\Sigma^n$.

Un groupe transitif par niveau est d'ordre infini car les tailles des orbites d'un groupe fini sont bornées par le produit entre la taille du groupe et la taille de l'ensemble sur lequel il agit.

On montre de cette façon que l'automate Baby Aleshin (donné en figure figure 4.3) engendre un groupe infini [BGK+08].

3.2 Signalisateur d'orbite et automates bornés

Les travaux de S. Sidki et *al.* [BBSZss, Sidoo] sur le problème de conjugaison des automorphismes d'arbres fournissent un test d'infinitude.

Soient a un automorphisme d'arbre sur Σ^* et $\mathbf{v} \in \Sigma^*$ un mot ; on note $\mathrm{Orb}_a(\mathbf{v})$ l'orbite de $\mathbf{v}$ sous l'action de a. Le **signalisateur d'orbites** de a est l'ensemble

$$\mathrm{OS}(a) = \{a^m{}_{|\mathbf{v}} \mid \mathbf{v} \in \Sigma^*, m = \#\mathrm{Orb}_a(\mathbf{v})\}.$$

Soient $b \in \mathrm{OS}(a)$, $\mathcal{O}_1, \ldots, \mathcal{O}_k$ les orbites de b sur Σ et $x_i \in \mathcal{O}_i$: $b^{m_i}{}_{|x_i} \in \mathrm{OS}(a)$ pour $m_i = \#\mathcal{O}_i$.

Si l'automorphisme d'arbre a possède un signalisateur d'orbites fini, on peut construire le graphe $\Phi(a)$ fini d'ensemble de sommets $\mathrm{OS}(a)$ et d'ensemble d'arêtes les

$$b \xrightarrow{m_i} b^{m_i}{}_{|x_i}, \text{ pour } 1 \leq i \leq k.$$

Alors a est d'ordre fini si et seulement si toutes les arêtes de son graphe $\Phi(a)$ sont étiquetées par 1 [BBSZss].

Un automorphisme d'arbre donné comme un état x d'un automate de Mealy est **borné** si et seulement si quels que soient deux cycles non triviaux de la partie de cet automate atteignable par x, ces cycles sont disjoints et il n'existe pas de chemin allant de l'un à l'autre [Sidoo]. Cette propriété est décidable et tout automorphisme d'arbre borné possède un signalisateur d'orbite fini [BBSZss], ce qui permet de lui appliquer le résultat précédent.

3.3 Test ad hoc - automate de Grigorchuk

L'automate de Grigorchuk (voir figure 4.1) fournit un exemple au problème de Burnside : tous les éléments du groupe engendré sont d'ordres finis, mais l'ordre du groupe est infini. L'infinitude du groupe peut être obtenue en montrant que le sous-groupe strict des applications qui stabilisent la première lettre est isomorphe au groupe lui-même.

4 Infinitude - graphes en hélice

Dans cette partie nous travaillons exclusivement sur des automates inversibles-réversibles, à la recherche de liens entre finitude et structure des graphes en hélice.

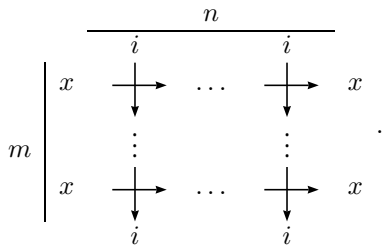
[AKL+12]

Proposition 6.7. Si le groupe engendré par un automate de Mealy inversible-réversible est fini, alors son graphe en hélice est une union de cycles disjoints.

Démonstration. Soit $\mathcal{A} = (A, \Sigma, \delta, \rho)$ un automate inversible-réversible qui engendre un groupe fini : son dual $\mathfrak{d}(\mathcal{A})$ engendre également un groupe fini.

Si on considère l'application de l'ensemble fini des sommets d'un graphe en hélice dans lui-même qui à un sommet associe son unique successeur dans ce graphe, le graphe en hélice est une union de cycles disjoints si et seulement si chaque sommet de ce graphe possède un prédécesseur.

Soient un état $x \in A$ et une lettre $i \in \Sigma$. Il existe deux entiers $m, n > 0$ tels que $\rho_x^m = \rho_{x^m} = \mathrm{id}_{\langle\mathcal{A}\rangle}$ et $\delta_i^n = \delta_{i^n} = \mathrm{id}_{\langle\mathfrak{d}(\mathcal{A})\rangle}$. Cela implique l'existence d'une transition $x^m \xrightarrow{i^n|i^n} x^m$ dans l'automate associé d'ordre (m, n). Le diagramme en croix correspondant s'écrit :



Le coin sud-est donne un prédécesseur à (x, i). □

La condition de la proposition 6.7 n'est pas suffisante : il existe des automates dont le graphe en hélice est une union de cycles disjoints et dont on sait par ailleurs qu'ils engendrent un groupe infini, comme l'automate Baby Aleshin (voir figures 4.3 et 4.6). De fait, la proposition 6.8 caractérise de façon très simple l'ensemble des automates inversibles-réversibles dont le graphe en hélice est une union de cycles disjoints.

On rappelle que l'opération $\mathfrak{i}$ sur un automate consiste à prendre les états inverses (définition donnée page 38).

[AKL+12]

Proposition 6.8. Soit $\mathcal{A}$ un automate inversible-réversible. Les propriétés suivantes sont équivalentes :

(i) $\mathcal{A}$ est biréversible,

(ii) $\mathfrak{di}\mathfrak{di}(\mathcal{A})$ est un automate de Mealy,

(iii) le graphe en hélice de $\mathcal{A}$ est une union de cycles disjoints.

Démonstration.

$(i) \Rightarrow (ii)$ $\mathcal{A}$ est biréversible, cela signifie qu'il est inversible et $\mathfrak{i}(\mathcal{A})$ est réversible, ce qui entraîne que $\mathfrak{di}(\mathcal{A})$ est inversible. A nouveau, on peut donc prendre l'inverse puis le dual et on obtient que $\mathfrak{di}\mathfrak{di}(\mathcal{A})$ est un automate de Mealy.

$(ii) \Rightarrow (i)$ L'automate $\mathcal{A}$ étant inversible-réversible, $\mathfrak{di}(\mathcal{A})$ est un automate de Mealy.
Par ailleurs, $\mathfrak{di}\mathfrak{di}(\mathcal{A})$ étant un automate de Mealy, $\mathfrak{i}\mathfrak{di}(\mathcal{A})$ est également un automate de Mealy. Comme c'est l'inverse de $\mathfrak{di}(\mathcal{A})$, on en déduit que $\mathfrak{di}(\mathcal{A})$ est inversible, donc $\mathfrak{i}(\mathcal{A})$ réversible. Ce qui entraîne que $\mathcal{A}$ est biréversible.

$(ii) \Leftrightarrow (iii)$ Dans le graphe en hélice d'un automate de Mealy, il part exactement un arc de chaque sommet. Le graphe en hélice d'un automate de Mealy est donc une union de cycles disjoints si et seulement s'il arrive au plus un arc par sommet.
On définit le graphe $\mathcal{G}$ d'ensemble de sommets $A^{-1} \times \Sigma^{-1}$ et d'arcs $(y^{-1}, j^{-1}) \to (x^{-1}, i^{-1})$ si $(x, i) \to (y, j)$ appartient au graphe en hélice $\mathcal{H}$ de $\mathcal{A}$.
Le graphe $\mathcal{G}$ est le graphe en hélice de $\mathfrak{di}\mathfrak{di}(\mathcal{A})$:
- si $\mathfrak{di}\mathfrak{di}(\mathcal{A})$ est un automate de Mealy, chaque sommet de $\mathcal{G}$ possède un successeur, donc chaque sommet de $\mathcal{H}$ possède un prédecesseur et $\mathcal{H}$ est une union de cycles disjoints,
- si $\mathcal{H}$ est une union de cycles disjoints, il en est de même pour $\mathcal{G}$ et on déduit de la remarque qui suit la définition du graphe en hélice que $\mathfrak{di}\mathfrak{di}(\mathcal{A})$ est un automate de Mealy.

□

On obtient un critère d'infinitude structurel, très simple à vérifier :

[AKL+12]

Corollaire 6.9. Tout automate inversible-réversible qui n'est pas biréversible engendre un groupe infini.

Ce critère n'a l'air de rien, mais il règle le problème de décision de la finitude pour une grande partie des automates inversibles-réversibles. En effet, il semble que la proportion d'automates biréversibles parmi les automates inversibles-réversibles décroît fortement.

TABLE 6.1 – Proportion d'automates biréversibles parmi les inversibles-réversibles (à isomorphisme près).

#lettres \ #états	2	3
2	$8/9 \simeq 88{,}89\,\%$	
3	$28/42 \simeq 66{,}67\,\%$	$335/1\,408 \simeq 23{,}79\,\%$

5 Gain de performance

Le problème de finitude pour les semi-groupes étant indécidable et celui pour les groupes restant ouvert, il est important de savoir si les critères exposés ici apportent quelque chose par rapport aux critères préexistants.

Le nombre d'automates de Mealy croît très vite en fonction du nombre d'états et de lettres, par exemple, à isomorphisme près, le nombre d'automates de Mealy sur 2 états et 2 lettres est de 76, alors que celui sur 3 états et 3 lettres est de 10 766 772. On ne peut donc tester la finitude et l'infinitude de (semi-)groupes d'automate de façon exhaustive que pour des petites dimensions.

Néanmoins, en petites dimensions, on peut souvent conclure par énumération en un temps raisonnable, même lorsqu'aucun critère connu ne s'applique. Mais le but de cette comparaison entre anciens et nouveaux critères est d'extrapoler, au moins de façon imaginaire, sur ce qui se passerait sur un automate plus grand. Et il est clair que sur un automate plus grand l'énumération n'est pas raisonnable. C'est pourquoi dans le tableau ci-dessous on ne tient pas compte de l'énumération pour déterminer la finitude d'un groupe. A contrario, on en tient compte si elle aboutit en un temps raisonnable à trouver un élément dont on peut prouver qu'il est d'ordre infini, ce qui entraîne l'infinitude du groupe.

Le tableau ci-dessous résume le nombre d'automates (à isomorphisme près) pour lesquels on est capable de prendre la décision sur la finitude du semi-groupe engendré. Pour des résultats plus détaillés, se reporter à [AKL+12].

TABLE 6.2 – Résumé des résultats expérimentaux de comparaison entre anciens et nouveaux critères.

	critères existants	**nouveaux critères**	**existants+nouveaux**	**total**
général $(3,2)$	429	1 241	1 333	4 003
inv. ou rév. $(3,3)$	111 640	134 276	227 365	236 558

Les critères existants regroupent les critères recensés avant la publication de [AKL+12]. La plupart sont implémentés dans deux paquets GAP [GAP08] : FR [Bar11] ou automgrp [MS08], mais pas tous (comme le critère d'Antonenko). Les nouveaux critères sont ceux donnés dans dans [AKL+12]. On peut également combiner ces critères. Considérons par exemple l'automate de Mealy $\mathcal{C}$ de la figure 6.4. Aucun des critères connus auparavant ne peut détecter qu'il engendre un groupe infini. Néanmoins, son dual $\mathfrak{d}(\mathcal{C})$ contient une composante connexe qui est isomorphe au dual de Baby Aleshin dont on sait qu'il engendre un groupe infini. On peut donc conclure à l'infinitude du groupe $\langle \mathcal{C} \rangle$.

6 Critère (non effectif) de finitude

Dans cette partie nous travaillons exclusivement sur des automates inversibles-réversibles. Le critère présenté ici n'est à ce jour pas effectif.

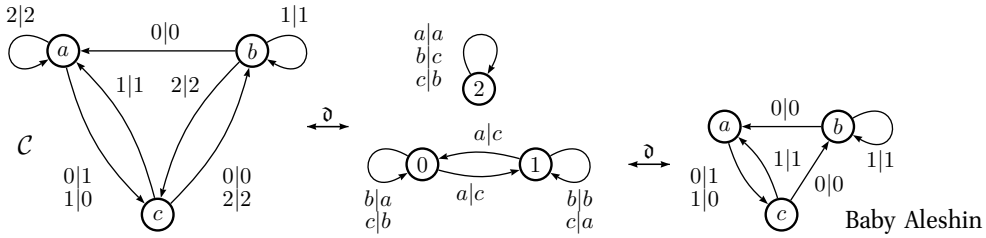


FIGURE 6.4 – L'automate de Mealy $\mathcal{A}$ engendre un groupe infini.

[AKL+12]

Théorème 6.10. Le groupe engendré par un automate inversible-réversible est fini si et seulement si les graphes en hélice de son automate étendu sont des unions de cycles disjoints uniformément bornés.

Pour montrer le théorème 6.10, nous montrons d'abord que c'est une condition nécessaire, puis que c'est une condition suffisante.

[AKL+12]

Proposition 6.11. Si un automate inversible-réversible engendre un groupe fini, alors les graphes en hélices de son automate étendu sont des unions de cycles disjoints uniformément bornés.

Démonstration. On peut montrer, avec le même type d'argument que pour la proposition 6.7, que les graphes en hélice d'un automate de Mealy sont des unions de cycles disjoints si et seulement si son graphe en hélice d'ordre $(1,1)$ est une union de cycles disjoints.

Soient $\mathcal{A}$ un automate inversible-réversible engendrant un groupe fini et $\tilde{\mathcal{A}}$ son automate étendu. D'après le corollaire 4.4, le groupe engendré par $\tilde{\mathcal{A}}$ est fini et d'après la proposition 6.7 et la remarque ci-dessus, ses graphes en hélices sont donc des unions de cycles disjoints.

D'après la proposition 4.3, le groupe $\langle \mathfrak{d}(\tilde{\mathcal{A}}) \rangle$ est également fini.

Soient $\mathcal{C}$, un cycle d'un graphe en hélice de $\tilde{\mathcal{A}}$ et $(\mathbf{u}, \mathbf{v}) \in (A \sqcup A^{-1})^* \times (\Sigma \sqcup \Sigma^{-1})^*$ un sommet de ce cycle. Chaque sommet de $\mathcal{C}$ est de la forme $(h(\mathbf{u}), g(\mathbf{v}))$, où g (*resp.* h) est un élément de $\langle \tilde{\mathcal{A}} \rangle$ (*resp.* $\langle \mathfrak{d}(\tilde{\mathcal{A}}) \rangle$). Comme les sommets sont deux à deux distincts, la longueur du cycle $\mathcal{C}$ est bornée par $\#\langle \tilde{\mathcal{A}} \rangle \times \#\langle \mathfrak{d}(\tilde{\mathcal{A}}) \rangle$. □

[AKL+12]

Proposition 6.12. Si les graphes en hélice de l'automate étendu d'un automate inversible-réversible sont des unions de cycles disjoints uniformément bornés, alors le groupe engendré par cet automate est fini.

Démonstration. La démonstration de cette proposition repose sur un résultat poussé de théorie des groupes, déjà cité en introduction de la section 3, et qui permet d'affirmer qu'un groupe d'automate dont les ordres des éléments sont bornés[1] est fini.

On dit qu'un mot sur les générateurs d'un groupe est **unitaire** s'il représente l'identité dans le groupe.

Le groupe $\langle \tilde{\mathcal{A}} \rangle$ étant infini, les ordres de ses éléments ne sont pas bornés : soit il existe un mot $\mathbf{x} \in (A \sqcup A^{-1})^*$ tel que $\rho_{\mathbf{x}}$ est d'ordre infini, soit il existe une suite de mots $(\mathbf{x}_n)_{n\in\mathbb{N}} \subseteq (A \sqcup A^{-1})^*$ telle que la suite des ordres des $(\rho_{\mathbf{x}_n})_{n\in\mathbb{N}}$ est strictement croissante. Nous allons traiter le deuxième cas (le premier est analogue).

1. Pour rappel, il existe des groupes d'automate infinis dont les éléments sont tous d'ordre fini, voir Grigorchuk [Gri80].

On note k_n l'ordre de l'élément $\rho_{\mathbf{x}_n}$: pour tout k, $1 \leq k < k_n$, il existe un mot $\mathbf{u}_k \in (\Sigma \sqcup \Sigma^{-1})^*$ tel que $\rho^k_{\mathbf{x}_n}(\mathbf{u}_k) = \mathbf{u}'_k \neq \mathbf{u}_k$. Comme $\langle \mathfrak{d}(\tilde{\mathcal{A}}) \rangle$ est un groupe, le mot $\mathbf{u}_k$ peut être étendu en un mot unitaire $\mathbf{u}_k\mathbf{v}_k$. On pose alors

$$\mathbf{w}_n = \mathbf{u}_1\mathbf{v}_1 \cdots \mathbf{u}_{k_n-1}\mathbf{v}_{k_n-1}.$$

Par construction $\rho_{\mathbf{x}_n}(\mathbf{w}_n) = \mathbf{u}'_1 \cdots \neq \mathbf{w}_n$.

Par ailleurs $\mathbf{u}_1\mathbf{v}_1$ étant unitaire, on a également

$$\begin{aligned} \rho^2_{\mathbf{x}_n}(\mathbf{w}_n) &= \rho^2_{\mathbf{x}_n}(\mathbf{u}_1\mathbf{v}_1)\rho^2_{\mathbf{x}_n}(\mathbf{u}_2\mathbf{v}_2 \cdots \mathbf{u}_{k_n-1}\mathbf{v}_{k_n-1}) \\ &= \rho^2_{\mathbf{x}_n}(\mathbf{u}_1\mathbf{v}_1)\mathbf{u}'_2 \cdots \neq \mathbf{w}_n. \end{aligned}$$

De la même façon, on montre que pour tout $k < k_n$, on a $\rho^k_{\mathbf{x}_n}(\mathbf{w}_n) \neq \mathbf{w}_n$.

Dans le graphe en hélice de $\tilde{\mathcal{A}}$ d'ordre $(|\mathbf{x}_n|, |\mathbf{w}_n|)$, on considère le cycle contenant le nœud $(\mathbf{x}_n, \mathbf{w}_n)$. Le mot $\mathbf{w}_n$ étant unitaire, les successeurs de $(\mathbf{x}_n, \mathbf{w}_n)$ dans ce cycle sont : $(\mathbf{x}_n, \rho_{\mathbf{x}_n}(\mathbf{w}_n))$, $(\mathbf{x}_n, \rho^2_{\mathbf{x}_n}(\mathbf{w}_n))$, ... Ce cycle est donc de longueur k_n. Comme $(k_n)_n$ diverge vers l'infini, les longueurs des cycles des graphes en hélice de $\tilde{\mathcal{A}}$ ne sont pas uniformément bornées. □

Le théorème 6.10 est alors un corollaire des propositions 6.11 et 6.12.

7

Le cas à deux lettres ou deux états

Dans ce chapitre nous montrons que la finitude est décidable pour les groupes engendrés par des automates de Mealy inversibles-réversibles à deux états, le passage au dual donnant le résultat pour des automates de Mealy inversibles-réversibles à deux lettres. Les techniques proposées ici ne sont pas toutes directement adaptables à un nombre supérieur d'états, ce qui ne permet pas d'étendre les résultats. Lorsqu'une extension est possible, cela est signalé dans le texte.

Tout ce chapitre repose sur l'étude de l'éventuelle connexité des puissances successives d'un automate de Mealy : on montre qu'un automate de Mealy réversible à deux états engendre un semi-groupe fini si et seulement s'il admet une puissance non connexe et que dans le cas contraire il engendre un semi-groupe libre.

Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$. On utilisera les deux faits suivants par la suite :

Remarque 7.1. Soit un entier n. Si chaque mot de A^* de longueur n équivaut à un mot de longueur strictement plus petite, alors $\mathcal{A}$ engendre un (semi-)groupe fini contenant exactement les fonctions de productions étendues associées aux mots de longueur au plus $n-1$.

Remarque 7.2. Si deux mots de A^* sont équivalents, alors leurs images par un élément du semi-groupe $\langle \mathfrak{d}(\mathcal{A}) \rangle_+$ sont également équivalentes. En effet :

$$\begin{aligned}
\rho_{\mathbf{x}} = \rho_{\mathbf{y}} &\implies \forall \mathbf{u}, \mathbf{v} \in \Sigma^*,\ \rho_{\mathbf{x}}(\mathbf{u}) = \rho_{\mathbf{y}}(\mathbf{u}) \text{ et } \rho_{\mathbf{x}}(\mathbf{uv}) = \rho_{\mathbf{y}}(\mathbf{uv}) \\
&\implies \forall \mathbf{u}, \mathbf{v} \in \Sigma^*,\ \rho_{\mathbf{x}}(\mathbf{u}) = \rho_{\mathbf{y}}(\mathbf{u}) \text{ et } \rho_{\mathbf{x}}(\mathbf{u})\rho_{\delta_{\mathbf{u}}(\mathbf{x})}(\mathbf{v}) = \rho_{\mathbf{y}}(\mathbf{u})\rho_{\delta_{\mathbf{u}}(\mathbf{y})}(\mathbf{v}) \\
&\implies \forall \mathbf{u}, \mathbf{v} \in \Sigma^*,\ \rho_{\delta_{\mathbf{u}}(\mathbf{x})}(\mathbf{v}) = \rho_{\delta_{\mathbf{u}}(\mathbf{y})}(\mathbf{v}) \\
&\implies \rho_{\delta_{\mathbf{u}}(\mathbf{x})} = \rho_{\delta_{\mathbf{u}}(\mathbf{y})}
\end{aligned}$$

Dans le cas d'un automate de Mealy à deux états, si z désigne un de ces états, on note $\bar{z}$ l'autre état.

1 Le semi-groupe engendré est libre ou fini

Un semi-groupe S est **libre** s'il existe un sous-ensemble X de S tel que tout élément de S s'écrit d'une façon unique sous la forme d'un mot sur X, le **rang** de S est le cardinal de X ; X est un ensemble de générateurs libres de S.

La notion de liberté existe aussi pour les groupes : un groupe G est libre s'il existe un sous-ensemble X de G tel que tout élément de G s'écrit d'une façon unique sous la forme réduit d'un mot sur $X \sqcup X^{-1}$. Un automate inversible peut engendrer un semi-groupe libre et un groupe non libre ; par exemple l'automate dual de Baby Aleshin (voir figure 4.3) engendre un semi-groupe libre, mais un groupe non libre : $01^{-1}01^{-1} = \mathrm{id}$.

Le but de cette section est de montrer le théorème suivant :

 [Kli13]

Théorème 7.3. Soit $\mathcal{A}$ un automate de Mealy réversible à deux états. Si $\mathcal{A}$ admet une puissance non connexe, il engendre un semi-groupe fini, sinon il engendre un semi-groupe libre de rang 2, les états de $\mathcal{A}$ étant des générateurs libres de ce semi-groupe.

Ce théorème est un cas particulier d'une conjecture de D. Savchuk [Sav09, Conjecture 3] qui stipule que tout groupe infini engendré par un automate biréversible possède un sous-semi-groupe libre non abélien.

Le théorème 7.3 est un corollaire de la proposition 7.7 et du cas particulier $p = 2$ de la proposition 7.10 ci-dessous.

Analysons la structure des composantes connexes des puissances successives de l'automate de Mealy $\mathcal{A}$.

Pour $m > 0$, $\mathbf{u}, \mathbf{v} \in A^m$ et $x, y \in A$, s'il existe un chemin de $\mathbf{u}x$ vers $\mathbf{v}y$ dans $\mathcal{A}^{m+1}$, alors il existe un chemin de $\mathbf{u}$ vers $\mathbf{v}$ dans $\mathcal{A}^m$, comme on peut le voir de façon immédiate avec des diagrammes en croix. Donc si $\mathcal{A}^n$ n'est pas connexe, il en est de même de toutes les puissances suivantes de $\mathcal{A}$. Ainsi il existe au plus un entier n tel que $\mathcal{A}^n$ est connexe et $\mathcal{A}^{n+1}$ ne l'est pas. Appelons-le **degré de connexion** de $\mathcal{A}$. Par convention, si $\mathcal{A}$ n'est pas connexe, son degré de connexion est nul et il a un degré de connexion infini si toutes ses puissances sont connexes.

On peut remarquer que l'automate Baby Aleshin (voir figure 4.3 page 39) est réversible, possède trois états, un degré de connexion égal à 1 et engendre un semi-groupe non libre puisque ses générateurs sont d'ordre 2. Ainsi le théorème 7.3 et la proposition 7.7 ne s'étendent pas à des ensembles d'états plus grands.

Un automate est de degré de connexion infini si et seulement si son dual est transitif par niveau (voir section 3.1). Le résultat de la section 1.1 a été montré dans ce contexte [BGK+08, Lemme 7], mais je donne ici une nouvelle preuve qui s'appuie fortement sur la structure de l'automate et repose sur des arguments extrêmement classiques en théorie des automates. L'intérêt de cette preuve est à la fois d'être plus intuitive pour des théoriciens des automates (du moins je l'espère) et de préparer le terrain pour les démonstrations de la section 2.

1.1 Degré de connexion fini

Nous montrons ici qu'un automate de Mealy réversible à deux états engendre un semi-groupe fini si et seulement si son degré de connexion est fini. L'idée de base est de borner les tailles des composantes connexes des puissances successives de cet automate, au-delà du degré de connexion.

 [Kli13]

Lemme 7.4. Soit un automate de Mealy réversible $\mathcal{A}$ ayant au moins deux états. Si $\mathcal{A}$ engendre un semi-groupe possédant des éléments de torsion, alors son degré de connexion est fini.

Idée. Notons A l'ensemble des états de $\mathcal{A}$. Si $\langle \mathcal{A} \rangle_+$ possède des éléments de torsion, alors il existe un mot $\mathbf{u} \in A^+$ et deux entiers $n \geq 0$ et $k > 0$ tels que $\mathbf{u}^n$ et $\mathbf{u}^{n+k}$ sont équivalents.

On montre que les états de la composante connexe contenant $\mathbf{u}^{n+2k}$ sont tous de la forme $\mathbf{v}\mathbf{w}^2$, où $|\mathbf{v}| = |\mathbf{u}|^n$ et $|\mathbf{w}| = |\mathbf{u}|^k$, ce qui entraîne que $\mathcal{A}^{(n+2k)|\mathbf{u}|}$ n'est pas connexe. □

Dans la suite de cette sous-section, $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est un automate de Mealy réversible à deux états ($A = \{x, y\}$), de degré de connexion fini n.

 [Kli13]

Lemme 7.5. Soit $\mathcal{C}$ une composante connexe de $\mathcal{A}^m$, pour un certain m et soit $\mathbf{u} \in A^m$ un état de $\mathcal{C}$. La composante connexe de $\mathbf{u}x$ est de taille $\#\mathcal{C}$ si elle ne contient pas $\mathbf{u}y$ et $2\#\mathcal{C}$ si elle contient $\mathbf{u}y$.

Démonstration. Soit $\mathcal{D}$ la composante connexe contenant $\mathbf{u}x : \mathbf{v} \in A^m$ appartient à $\mathcal{C}$ si et seulement s'il existe $z \in A$ tel que $\mathbf{v}z$ appartienne à $\mathcal{D}$, d'où : $\#C \leq \#D \leq 2\#C$.

Soient $\mathbf{v}$ appartenant à $\mathcal{C}$ et $z \in A$: $\mathbf{u}x$ et $\mathbf{v}z$ appartiennent à la même composante connexe si et seulement si $\mathbf{u}y$ et $\mathbf{v}\bar{z}$ appartiennent à la même composante connexe (où $z, \bar{z} \in \{x, y\}$ et $z \neq \bar{z}$). D'où le résultat. □

[Kli13]

Lemme 7.6. Soit un automate de Mealy réversible $\mathcal{A}$, de degré de connexion n. Pour $m \geq n$, les composantes connexes de $\mathcal{A}^m$ sont toutes de taille 2^n.

Idée. Par récurrence sur $m \geq n$.

Pour $m \in \{n, n+1\}$, la propriété est vraie (en utilisant le lemme 7.5 pour $m = n+1$).

Soit $m > n+1$. On suppose que les composantes connexes de $\mathcal{A}^{m-1}$ et $\mathcal{A}^m$ sont de taille 2^n. Si $\mathcal{C}$ est une composante connexe de $\mathcal{A}^{m+1}$, le lemme 7.5 et l'hypothèse de récurrence permettent de conclure que $\mathcal{C}$ est de taille 2^n ou 2^{n+1}. On montre ensuite par l'absurde que $\mathcal{C}$ n'est pas de taille 2^{n+1}, car cela entraînerait, par le lemme 7.5, l'existence d'une composante connexe de $\mathcal{A}^{m-1}$ de taille 2^{n-1}. □

Proposition 7.7. [BGK+08, Kli13] Le degré de connexion d'un automate de Mealy réversible à deux états est fini si et seulement s'il engendre un semi-groupe fini.

Démonstration. Soit $\mathcal{A}$ un automate de Mealy réversible à deux états.

Si le degré de connexion de $\mathcal{A}$ est nul, $\langle \mathfrak{d}(\mathcal{A}) \rangle_+$ est le semi-groupe trivial et $\langle \mathcal{A} \rangle_+$ est fini d'après la proposition 4.3.

Sinon, soit $n \geq 1$ le degré de connexion de $\mathcal{A}$: pour un $m \geq n$, les composantes connexes de $\mathcal{A}^m$ sont toutes de taille 2^n. A numérotation des états près, il ne peut donc y avoir qu'un nombre fini de composantes connexes distinctes. On en déduit par la remarque 7.1 que $\langle \mathcal{A} \rangle_+$ est fini.

La réciproque est un cas particulier du lemme 7.4. □

1.2 Degré de connexion infini

Dans cette sous-section, je montre que si un automate de Mealy réversible possédant un nombre premier d'états a un degré de connexion infini, alors il engendre un semi-groupe libre, ses états étant alors des générateurs libres. L'idée de base est de borner les tailles des classes de Nerode des puissances successives de $\mathcal{A}$.

Pour les deux prochains lemmes, $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est un automate de Mealy réversible à p états, p étant premier, de degré de connexion infini ($A = \{x_1, \ldots, x_p\}$). Cet automate engendre un semi-groupe infini (voir lemme 7.4).

Lemme 7.8. Les classes de Nerode d'une puissance donnée A^m ont toutes même taille qui est une puissance de p.

Démonstration. Soit $\mathbf{u} \in A^m$. Si $\#[\mathbf{u}] = p^m$, alors $[\mathbf{u}] = A^m$. Sinon, soit $\mathbf{v} \in A^m - [\mathbf{u}]$. Comme $\mathcal{A}^m$ est connexe, il existe $\mathbf{r} \in \Sigma^*$ tel que $\mathbf{v} = \delta_{\mathbf{r}}(\mathbf{u})$.

Par la remarque 7.2, l'image d'un mot équivalent à $\mathbf{u}$ par $\delta_{\mathbf{r}}$ est équivalent à $\mathbf{v}$. De par la réversibilité de l'automate $\mathcal{A}^m$, $\delta_{\mathbf{r}}$ est une permutation de A^m et $\#[\mathbf{u}] = \#[\mathbf{v}]$.

L'ensemble des états de $\mathcal{A}^m$ ayant pour nombre d'éléments une puissance de p, nombre premier, il en va de même de toutes les classes de Nerode. □

[Kli13]

Lemme 7.9. Il n'y a pas de relation non triviale sur A.

Démonstration. Il n'y a pas de couple de mots équivalents de longueurs différentes sur A. En effet, si $\mathbf{u}$ et $\mathbf{v}$ sont équivalents de longueurs différentes $|\mathbf{u}| < |\mathbf{v}|$, alors chaque mot de longueur $|\mathbf{v}|$ est équivalent à un mot de longueur $|\mathbf{u}|$ et le semi-groupe engendré est fini (remarques 7.1 et 7.2).

Montrons qu'il n'y a pas de couple de mots équivalents de même longueur sur A.

Soient $\mathbf{u}$ et $\mathbf{v}$ deux mots équivalents différents et de même longueur $n+1$. On montre par récurrence sur $m > n$ que $\mathfrak{m}(\mathcal{A}^m)$ possède au plus p^n états.

L'automate $\mathcal{A}^{n+1}$ possède p^{n+1} états. Les mots $\mathbf{u}$ et $\mathbf{v}$ appartiennent à la même classe de Nerode : d'après le lemme 7.8, toutes les classes de Nerode de A^{n+1} possèdent au moins p éléments et donc $\mathfrak{m}(\mathcal{A}^{n+1})$ a au plus p^n états.

Supposons que $\mathfrak{m}(\mathcal{A}^m)$ possède au plus p^n états. Toutes les classes de Nerode étant de même taille (lemme 7.8), elles ont donc au moins p^{m-n} éléments. Rappelons que $x_1 \in A$ est un des états de $\mathcal{A}$ et regardons de plus près la classe de x_1^m : $[x_1^m]$ contient les éléments deux à deux distincts

$$x_1^m, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{p^{m-n}-1}.$$

Parmi ces mots, l'un contient le plus court suffixe en x_1^* et ce n'est pas x_1^m ; sans perte de généralité supposons que ce soit $\mathbf{u}_1$. Alors $[x_1^{m+1}]$ contient les éléments suivants qui sont deux à deux distincts :

$$x_1^{m+1}, \mathbf{u}_1 x_1, \mathbf{u}_2 x_1, \ldots, \mathbf{u}_{p^{m-n}-1} x_1, x_1 \mathbf{u}_1.$$

Or d'après le lemme 7.8, $\#[x_1^{m+1}]$ est une puissance de p, donc $\#[x_1^{m+1}] \geq p^{m+1-n}$. Comme toutes les classes de Nerode de A^{m+1} ont même cardinalité, on peut conclure de $\mathfrak{m}(\mathcal{A}^{m+1})$ possède au plus $p^{m+1}/p^{m+1-n} = p^n$ éléments.

Par conséquent, il existe $k < \ell$ tels que $\mathfrak{m}(\mathcal{A}^k)$ and $\mathfrak{m}(\mathcal{A}^\ell)$ sont isomorphes. Le semi-groupe $\langle \mathcal{A} \rangle_+$ est donc fini d'après la remarque 7.1, ce qui est impossible. □

La proposition suivante est un corollaire des lemmes 7.4 et 7.9.

[Kli13]

Proposition 7.10. Soit $\mathcal{A}$ un automate de Mealy réversible possédant p états, p premier. Si l'automate $\mathcal{A}$ a un degré de connexion infini, alors il engendre un semi-groupe libre. Ce semi-groupe est alors de rang p, les états de l'automate $\mathcal{A}$ étant des générateurs libres du semi-groupe.
La réciproque est vraie pour $p = 2$.

2 Décidabilité de la finitude et de la liberté

Le but de cette section est de montrer la décidabilité de la finitude et de la liberté des semi-groupes engendrés par des automates inversibles-réversibles à deux états, en établissant la réciproque du corollaire 6.6 dans ce cas précis.

[Kli13]

Lemme 7.11. Soit $\mathcal{A} = (A, \Sigma, \delta, \rho)$ un automate de Mealy inversible-réversible possédant deux états et de degré de connexion n. Deux éléments de Σ^* dont les actions sur un mot de A^n coïncident sont équivalents.

Démonstration. Il suffit de montrer que le seul élément de $\langle\mathfrak{d}(\mathcal{A})\rangle_+$ qui fixe un mot de A^n est id_{A^*}, l'identité sur A^*.

Si $n = 0$, $\langle\mathfrak{d}(\mathcal{A})\rangle_+$ est le semi-groupe trivial et le résultat est vrai. Sinon, soient $\mathbf{u} \in A^n$ et $\mathbf{s} \in \Sigma^*$ tels que $\mathbf{u}$ soit stable par $\delta_\mathbf{s} : \delta_\mathbf{s}(\mathbf{u}) = \mathbf{u}$.

D'après le lemme 7.5, $\mathcal{A}^{n+1}$ a deux composantes connexes : $\mathbf{u}x$ appartient à l'une d'entre elles et $\mathbf{u}y$ à l'autre. Si on regarde plus loin, une composante connexe de $\mathcal{A}^m$, pour $m \geq n$, est à l'origine de deux composantes connexes de $\mathcal{A}^{m+1}$ obtenues en concaténant x ou y à la fin de l'étiquette de chaque état (mais pas nécessairement la même lettre après chaque état dans la composante connexe considérée). Par ailleurs, toutes les composantes connexes de $\mathcal{A}^{m+1}$ sont construites de cette façon. Ainsi, deux mots différents de même longueur $m > n$ ayant un préfixe commun de longueur n appartiennent à deux composantes connexes différentes de $\mathcal{A}^m$.

Soit $\mathbf{t} \in \Sigma^*$ tel que $\rho_\mathbf{u}(\mathbf{s}) = \mathbf{t}$ et soient $\mathbf{v}, \mathbf{w} \in A^*$ tels que : $\delta_\mathbf{t}(\mathbf{v}) = \mathbf{w}$.

Les mots $\mathbf{uv}$ et $\mathbf{uw}$ appartiennent à la même composante connexe ($\delta_\mathbf{s}(\mathbf{uv}) = \mathbf{uw}$) et ont un préfixe commun de longueur n, ils sont donc égaux. Ainsi : $\delta_\mathbf{t} = \mathrm{id}_{A^*}$. L'automate $\mathfrak{d}(\mathcal{A})$ étant réversible, on en déduit que $\delta_\mathbf{s} = \mathrm{id}_{A^*}$ car il existe un chemin de $\mathbf{t}$ vers $\mathbf{s}$. □

On a un résultat similaire, bien que plus faible, pour des mots plus courts dans des automates tensoriellement clos (pour rappel : l'alphabet d'un tel automate contient exactement les éléments du groupe engendré par le dual ; voir une définition précise page 43).

[Kli13]

Lemme 7.12. Soient $\mathcal{A} = (A, \Xi, \delta, \rho)$ un automate de Mealy inversible-réversible tensoriellement clos à deux états et de degré de connexion fini n, et k un entier, $1 \leq k \leq n$. Si deux éléments de Ξ^* coïncident sur un certain mot de A^k, alors ils ont même action sur tout A^k.

Démonstration. Tout mot de Ξ^* étant équivalent à une lettre de Ξ, il suffit de montrer le résultat sur les lettres.

Les composantes connexes de $\mathcal{A}^n$ sont fortement connexes car l'automate est réversible. Il existe donc toujours un chemin entre deux mots quelconques $\mathbf{u} \in A^n$ et $\mathbf{v} \in A^n$ de la même composante connexe, dont l'étiquette appartient à Ξ^*. L'automate $\mathcal{A}$ étant tensoriellement clos, tout mot sur Ξ est équivalent à une lettre de Ξ. La composante connexe de $\mathbf{u}$ et $\mathbf{v}$ est donc complète en tant que graphe et il en va de même de toute composante connexe de $\mathcal{A}^n$. De plus chaque transition a une unique étiquette : sinon ces étiquettes coïncideraient sur un mot de A^n et seraient donc équivalentes d'après le lemme 7.11, la minimalité de l'automate $\mathfrak{d}(\mathcal{A}^n)$ entraînant alors l'unicité.

Par ailleurs cet automate possède 2^n états, on en déduit que $\#\Xi = 2^n$. Par hypothèse, Ξ est l'ensemble des éléments de $\langle\mathfrak{d}(\mathcal{A})\rangle_+$, donc $\#\langle\mathfrak{d}(\mathcal{A})\rangle_+ = 2^n$.

Considérons le minimisé de $\mathfrak{d}(\mathcal{A})$ et plus particulièrement la suite d'équivalences de plus en plus fines $(\equiv_k)$ introduites en section 3.5. Chaque n-classe de Ξ est un singleton d'après le lemme 7.11, la suite $(\equiv_k)$ est donc constante à partir du rang n. Comme $\#\langle\mathfrak{d}(\mathcal{A})\rangle_+ = 2^n$ et $\#A = 2$, l'équivalence $\equiv_k$ coupe chaque $(k-1)$-classe en deux ensembles de même cardinalité :

$$\forall k, 0 \leq k \leq n, \forall s \in \Xi,\ \#[s]_k = \#[s]_{k-1}/2 = 2^{n-k}.$$

Soient k, $1 \leq k \leq n$, $\mathbf{u} \in A^k$ et $s \in \Xi$. On a :

$$[s]_k \subseteq \{t \in \Xi \mid t(\mathbf{u}) = s(\mathbf{u})\}. \tag{7.1}$$

Dans l'équation (7.1), l'ensemble de gauche possède 2^{n-k} éléments, c'est l'ensemble des éléments de Ξ qui coïncident avec s sur A^k. Etant donné que deux éléments de Ξ dont les actions coïncident sur A^n sont équivalents, l'ensemble de droite dans l'équation (7.1) est de taille au plus $\#A^{n-k} = 2^{n-k}$, on en déduit que les deux ensembles apparaissant dans l'équation (7.1) sont égaux, d'où le résultat. □

Une conséquence du lemme 7.12 est qu'un élément de Ξ^* qui fixe au moins un mot sur A de longueur k fixe tout A^k, pour un entier k donné.

Notons id l'identité sur A et σ la permutation de x et y. On peut traduire le lemme 7.12 en termes de portraits de $\mathfrak{d}(\mathcal{A})$: lorsque deux k-portraits de $\mathfrak{d}(\mathcal{A})$ possèdent une branche identique, ils sont égaux. En particulier, $\mathcal{I}_k$ étant un portrait de $\mathfrak{d}(\mathcal{A})$, si tous les nœuds d'une branche d'un k-portrait de $\mathfrak{d}(\mathcal{A})$ sont étiquetés par id, ce portrait est $\mathcal{I}_k$. Ainsi si dans un k-portrait de $\mathfrak{d}(\mathcal{A})$, tous les sommets de niveau au plus $k-1$ sont étiquetés par id, ce portrait est soit $\mathcal{I}_k$, soit $\mathcal{I}_{k-1}\lfloor\sigma,\sigma\rfloor$ (notation définie page 45). Remarquons que pour $k \leq n$, $\mathcal{I}_k$ et $\mathcal{I}_{k-1}\lfloor\sigma,\sigma\rfloor$ sont tous les deux des portraits de $\mathfrak{d}(\mathcal{A})$.

D'après le lemme 7.11, tout élément de $\langle\mathfrak{d}(\mathcal{A})\rangle_+$ dont le n-portrait est $\mathcal{I}_n$ agit trivialement sur A^*.

Quels sont les portraits possibles pour $\mathfrak{d}(\mathcal{A})$? L'automate $\mathcal{A}^n$ étant connexe et l'automate $\mathcal{A}$ tensoriellement clos, il est immédiat que toute suite finie $(\pi_i)_{1\leq i\leq n} \in \{\mathrm{id},\sigma\}^n$ étiquette une branche d'un n-portrait de A : dans $\mathcal{A}^n$, il y a une transition étiquetée en entrée par $s \in \Sigma$ allant de x^n vers $\pi_1(x)\cdots\pi_n(x)$ et la branche la plus à gauche de $\mathfrak{p}_n[\![s]\!]$ est étiquetée par $(\pi_i)_{1\leq i\leq n}$.

[Kli13]

Lemme 7.13. Soient $\mathcal{A} = (A,\Xi,\delta,\rho)$ un automate de Mealy inversible-réversible tensoriellement clos à deux états et de degré de connexion fini n, et k un entier, $1 \leq k \leq n$. Les états de $\mathcal{A}$ sont équivalents.

Démonstration. Commençons par montrer que les k-portraits de $\mathcal{A}$, $k \leq n$, sont homogènes, par récurrence sur $k \geq 1$. Un 1-portrait est restreint à sa racine et est donc homogène.

Supposons que les ℓ-portraits de $\mathfrak{d}(\mathcal{A})$ sont tous homogènes, pour $\ell \leq k < n$. Considérons une lettre $s \in \Xi$ et $\mathcal{S} = \mathfrak{p}_{k+1}[\![s]\!]$: il est presqu'homogène par construction. Plus précisément : $\mathcal{S} = \mathfrak{p}_k[\![s]\!]\lfloor\tau_1,\tau_2\rfloor$ pour certaines permutations τ_1,τ_2 de Σ.

Premier cas : δ_s permute x et y. Considérons le $(n+1)$-portrait $\mathcal{K}$ suivant :
- la restriction de $\mathcal{K}$ aux niveaux 0 à $(n-k-1)$ est $\mathcal{I}_{n-k}$,
- dans le coin inférieur gauche de $\mathcal{I}_{n-k}$, on met $\mathfrak{p}_{k+1}[\![s]\!]$: la racine de $\mathfrak{p}_{k+1}[\![s]\!]$ est le fils gauche de la feuille en bas à gauche de $\mathcal{I}_{n-k}$ (c'est possible car on peut choisir la branche la plus à gauche d'un portrait en appliquant le lemme 7.12 et $\mathfrak{p}_{k+1}[\![s]\!]$ est un portrait de $\mathfrak{d}(\mathcal{A})$),
- on complète le tout pour obtenir un portrait de $\mathfrak{d}(\mathcal{A})$.

La branche la plus à gauche de $\mathcal{K}^2$ commence par id^n. Par le lemme 7.11, on en déduit que $\mathcal{K}^2$ est le $(n+1)$-portrait identité, ce qui implique que $\tau_1 = \tau_2$ d'après la remarque 4.8 et le lemme 7.11.

Second cas : δ_s stabilise A. Soit $\mathcal{L}$ le $(k+1)$-portrait dont la racine est étiquetée par σ et tous les autres sommets par id : c'est un portrait de $\mathfrak{d}(\mathcal{A})$ car tous les $(k+1)$-portraits homogènes avec σ à la racine le sont d'après le cas précédent. En multipliant $\mathcal{S} = \mathfrak{p}_k[\![s]\!]\lfloor\tau_1,\tau_2\rfloor$ par $\mathcal{L}$, on obtient un $(k+1)$-portrait non homogène dont la racine est étiquetée par σ et qui est un portrait de $\mathfrak{d}(\mathcal{A})$. Ceci est impossible.

On montre de même que les k-portraits de $\mathcal{A}$, $k > n$, sont homogènes, en partant du portrait $\mathfrak{p}_k[\![s]\!]$.

Pour toute lettre $s \in \Xi$, $\rho_x(s)$ et $\rho_y(s)$ sont donc équivalents. L'automate étant tensoriellement clos, ils sont égaux et $\rho_x = \rho_y$. □

[Kli13]

Théorème 7.14. Un automate de Mealy inversible-réversible à deux états engendre un groupe fini si et seulement s'il est $\mathfrak{mD}$-trivial.

Démonstration. Soit $\mathcal{A}$ un automate de Mealy inversible-réversible à deux états.

D'après le corollaire 6.6, si $\mathcal{A}$ est $\mathfrak{md}$-trivial, il engendre un groupe fini.

Supposons que $\mathcal{A}$ engendre un groupe fini et considérons sa clôture tensorielle $\mathfrak{c}(\mathcal{A})$: $\mathfrak{c}(\mathcal{A})$ engendre un groupe fini d'après la remarque 4.7. Le degré de connexion de $\mathfrak{c}(\mathcal{A})$ est fini par la proposition 7.7 et $\mathfrak{c}(\mathcal{A})$ est donc $\mathfrak{md}$-trivial d'après le lemme 7.13. L'automate $\mathcal{A}$ possède deux états ; il est donc $\mathfrak{md}$-trivial si et seulement si $\mathfrak{mdmd}(\mathcal{A})$ est trivial. Or l'alphabet de $\mathfrak{dmd}(\mathcal{A})$ s'injecte dans l'alphabet de $\mathfrak{c}(\mathcal{A})$, on en déduit que $\mathcal{A}$ est $\mathfrak{md}$-trivial. □

Le théorème suivant résume les résultats de décidabilité de ce chapitre.

[Kli13]

Théorème 7.15. On peut décider si un automate de Mealy inversible-réversible à deux états d'alphabet Σ engendre un groupe fini, en temps $\mathcal{O}(\#\Sigma \log \#\Sigma)$. On peut décider s'il engendre un semi-groupe libre en temps $\mathcal{O}(\#\Sigma \log \#\Sigma)$.
On peut décider si un automate de Mealy inversible-réversible d'ensemble d'états A sur un alphabet à deux lettres engendre un groupe fini, en temps $\mathcal{O}(\#A \log \#A)$.

De quoi vous sert votre vitesse ?

Le lièvre et la tortue, Jean de la Fontaine

Cette partie expose les résultats liés à l'implémentation et obtenus dans :

[KMP12] I. Klimann, J. Mairesse, and M. Picantin. Implementing computatioons in automaton (semi)groups. In N. Moreira and R. Reis, editors, *CIAA*, number 7381 in LNCS, pages 240–252, 2012.

8

Minimisation et implémentation

Cette section décrit comment la minimisation permet d'accélérer substantiellement les procédures de calcul existantes sur les (semi-)groupes, plus spécifiquement celles concernant la finitude, la croissance et l'ordre.

Notre point de départ sont les deux paquets GAP [GAP08] suivants : FR développé par L. Bartholdi [Bar11] et automgrp développé par Y. Muntyan et D. Savchuk [MS08].

1 Croissance d'un (semi-)groupe d'automate

La croissance d'un (semi-)groupe correspond à la vitesse à laquelle celui-ci va grossir lorsqu'on compose les éléments au fur et à mesure. Donnons une définition plus formelle. Soient un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ et un mot $\mathbf{x} \in A^*$. La **longueur** de $\rho_\mathbf{x}$, notée $|\rho_\mathbf{x}|$, est la longueur minimale d'un mot équivalent à $\mathbf{x}$:

$$|\rho_\mathbf{x}| = \min\{n \mid \exists \mathbf{y} \in A^n,\ \rho_\mathbf{x} = \rho_\mathbf{y}\}.$$

La **fonction de croissance** de l'automate $\mathcal{A}$ est la série formelle qui énumère les éléments du semi-groupe en fonction de leur longueur :

$$\sum_{g \in \langle \mathcal{A} \rangle_+} t^{|g|} = \sum_{n \in \mathbb{N}} \#\{g \in \langle \mathcal{A} \rangle_+ \,;\, |g| = n\} t^n.$$

On s'intéresse au calcul des termes de cette série. Les paquets FR et automgrp utilisent l'énumération ; cette technique fonctionne pour n'importe quel (semi-)groupe dans lequel le problème du mot est résoluble : on part des générateurs et on engendre de nouveaux éléments étape par étape, jusqu'à ne plus en obtenir. Le fait que le (semi-)groupe soit engendré par un automate ne sert qu'à apporter une solution au problème du mot.

La nouvelle implémentation `AutomatonGrowth` proposée dans [KMP12] utilise complètement et en permanence la structure d'automate et la puissance de la minimisation pour obtenir de manière globale tous les éléments de longueur n à partir des éléments de longueur $n-1$. A chaque entier n, on associe un automate de Mealy $\mathcal{A}_n$ dont les états sont exactement les éléments de longueur au plus n :

$$\mathcal{A}_n = \mathfrak{m}(\mathcal{A}_{n-1} \times \mathfrak{m}(\mathcal{A}')) \quad \text{et} \quad \mathcal{A}_1 = \mathfrak{m}(\mathcal{A}),$$

où $\mathcal{A}' = \mathcal{A}$ si un état de $\mathcal{A}$ induit l'identité, et l'union de $\mathcal{A}$ et de l'automate trivial sinon.

```
AutomatonGrowth:= function(arg)
local   aut, radius, growth, sph, curr, next, r;
aut:=arg[1];   # automate de Mealy
if Length(arg)>1 then radius:=arg[2];
```

```
                else radius:=infinity;
fi;
r := 0;
curr := TrivialMealyMachine([1]);
next := Minimized(aut);
aut := Minimized(next+TrivialMealyMachine(Alphabet(aut)));
sph := aut!.nrstates - 1;  # nombre d'etats non triviaux
growth := [next!.nrstates-sph];
while sph>0 and r<radius
do  Add(growth,sph);
    r := r+1;
    curr := next;
    next := Minimized(next*aut);
    sph := next!.nrstates-curr!.nrstates;
od;
return growth;
end;
```

L'appel `AutomatonGrowth(aut)` permet de calculer la croissance du semi-groupe $\langle \mathtt{aut} \rangle_+$, tandis que l'appel `AutomatonGrowth(aut+aut^-1)` calcule la croissance du groupe $\langle \mathtt{aut} \rangle$.

Résultats expérimentaux. Commençons par faire tourner `AutomatonGrowth` et `WordGrowth` (paquet FR) sur l'automate de Grigorchuk (voir figure 4.1). Pour un rayon de 10, `AutomatonGrowth` est nettement plus rapide, 76 ms contre 9 912 ms[1]. La raison en est simple : `WordGrowth` appelle la procédure de minimisation 57 577 fois tandis que `AutomatonGrowth` ne l'appelle que 12 fois :

```
gap> aut := GrigorchukMachine;; f := sgp(aut);; radius:= 10;;
gap> ProfileFunctions([Minimized]);
gap> WordGrowth(f, radius); time;
[ 1, 4, 6, 12, 17, 28, 40, 68, 95, 156, 216 ]
9912
gap> DisplayProfile();
  count  self/ms  chld/ms  function
  57577     7712        0  Minimized
            7712           TOTAL
gap> ProfileFunctions([Minimized]);
gap> AutomatonGrowth(aut, radius); time;
[ 1, 4, 6, 12, 17, 28, 40, 68, 95, 156, 216 ]
76
gap> DisplayProfile();
  count  self/ms  chld/ms  function
     12       72        0  Minimized
              72           TOTAL
```

Comparons maintenant les temps d'exécution des diverses implémentations pour les premiers termes de la fonction de croissance des 335 automates de Mealy biréversibles à 3 lettres et 3 états (à isomorphisme près). Certains résultats n'apparaissent pas dans la table 8.1 faute d'avoir pu les obtenir en temps raisonnable.

TABLE 8.1 – Temps moyen (en ms)

rayon	1	2	3	4	5	6	7
`WordGrowth` – FR	3,4	29,0	555,0	8 616,5	131 091,4	2 530 170,3	?
`Growth` – automgrp	0,7	2,8	16,9	158,9	1 909,0	22 952,8	?
`AutomatonGrowth`	0,6	1,8	5,9	28,9	187,3	1 005,9	7 131,4

1. Programmes exécutés sur un processeur Intel Core 2 Duo 3,06 GHz.

2 Ordre d'un (semi-)groupe

FR et automgrp implémentent des procédures de calcul de l'ordre d'un (semi-)groupe d'automate fini. Ces deux paquets utilisent des approches tout à fait orthogonales. L'implémentation proposée dans [KMP12] affine l'approche de FR and reste orthogonale à celle de automgrp.

L'implémentation de automgrp**.** Le paquet GAP automgrp définit la fonction `LevelOfFaithfulAction` qui permet de calculer—parfois de façon extrêmement efficace—l'ordre du groupe engendré. Le principe est le suivant : soient $\mathcal{A}$ un automate de Mealy inversible sur l'alphabet Σ et G_k le groupe engendré par les restrictions à Σ^k des fonctions de productions étendues. Si $\#G_k = \#G_{k+1}$ pour un certain k, alors $\langle\mathcal{A}\rangle$ est fini, d'ordre $\#G_k$. Cette fonction peut être facilement adaptée à un automate de Mealy non inversible.

`LevelOfFaithfulAction` n'est pas utilisable pour calculer la fonction de croissance : à chaque étape on calcule un quotient du semi-groupe. Mais c'est une bonne stratégie pour calculer l'ordre du semi-groupe. De plus cette méthode profite du fait que GAP a été écrit pour manipuler des permutations de groupes finis.

L'implémentation de FR **et la nouvelle implémentation.** Tout algorithme qui permet de calculer la fonction de croissance peut être utilisé pour calculer l'ordre d'un (semi-)groupe fini. Il suffit de calculer la fonction de croissance jusqu'à trouver un coefficient nul. C'est l'approche utilisée dans FR. Comme nous avons proposé, dans la section précédente, une nouvelle implémentation pour la fonction de croissance, nous obtenons directement une nouvelle procédure pour calculer l'ordre. Appelons-la `AutomSGrOrder`.

Résultats expérimentaux. L'orthogonalité des deux approches précédentes peut être illustrée par l'automate de la figure 5.2. Ni la fonction `Order` de FR, ni `AutomSGrOrder` ne sont capables de calculer l'ordre du grand groupe, alors que automgrp, *via* `LevelOfFaithfulAction`, permet d'obtenir le résultat en seulement $14\,338\,\mathrm{ms}$. D'un autre côté, `AutomSGrOrder` calcule l'ordre du petit semi-groupe en $17\,\mathrm{ms}$, alors qu'une adaptation de `LevelOfFaithfulAction` (aux automates de Mealy non inversibles) met $2\,193\,\mathrm{ms}$.

3 Finitude

Toute procédure qui calcule l'ordre d'un (semi-)groupe d'automate produit une procédure de semi-décision pour le problème de finitude. Les deux paquets FR and automgrp appliquent un certain nombre de critères de finitude ou d'infinitude (voir des exemples de tels critères au chapitre 4) puis essayent en dernier recours de calculer l'ordre du groupe.

Dans [KMP12], nous saupoudrons ces procédures de minimisation : on remplace le (semi-)groupe dont on veut décider s'il est fini ou non successivement par d'autres (semi-)groupes qui sont finis si et seulement si le (semi-)groupe d'origine l'était. Il est possible d'incorporer cette astuce pour obtenir deux nouvelles implémentations, l'une dans l'esprit de FR et l'autre dans l'esprit de automgrp. Ces nouvelles implémentations sont plus efficaces que les précédentes, de plusieurs ordres de grandeur. Les deux approches restent utiles car selon le cas l'une ou l'autre sera la plus rapide.

Les nouvelles impleméntations La conception de la procédure `IsFinite1` est consistante avec celle de `AutomatonGrowth`. Ainsi `IsFinite1` est beaucoup plus proche de FR que de automgrp. Nous proposons ici une version qui fonctionne en parallèle sur l'automate et son dual.

```
IsFinite1 := function (aut, limit)
local radius, dual, curr1, next1, curr2, next2;
radius := 0;
```

```
aut := MDReduced(Prune(aut)); # automate elague puis md-reduit
dual := DualMachine(aut);
curr1 := MealyMachine([[1]],[()]);
curr2 := curr1;
next1 := aut;
next2 := dual;
while curr2!.nrstates<>next2!.nrstates and radius<limit
do  radius := radius + 1;
    curr1 := next1;
    next1 := Minimized(next1*aut);
    if curr1!.nrstates<>next1!.nrstates
    then  curr2 := next2;
          next2 := Minimized(next2*dual);
    else  return true;
    fi;
od;
if curr2!.nrstates = next2!.nrstates then  return true; fi;
return fail;
end;
```

La procédure `IsFinite2` est une amélioration de `LevelOfFaithfulAction` (automgrp) : la minimisation est faite sur le dual. Cette procédure peut être améliorée en parallélisant son exécution sur l'automate et son dual.

```
IsFinite2 := function(aut,limit)
local   Fonc1, Fonc2, next, cs, ns, lev;
aut := MDReduced(Prune(aut));
if IsInvertible(aut) then  Fonc1:=Group;
                           Fonc2:=PermList;
                     else  Fonc1:=Semigroup;
                           Fonc2:=Transformation; fi;
lev := 0; cs := 1;
ns := Order(Fonc1(List(aut!.output,Fonc2)));
aut := DualMachine(aut);
next := aut;
while cs<ns and lev<limit
do
    lev := lev+1;
        cs := ns;
    next := Minimized(next*aut);
    ns := Order(Fonc1(List(DualMachine(next)!.output,Fonc2)));
od;
if cs=ns then return true; else return fail; fi;
end;
```

TABLE 8.2 – Temps moyen (en ms) pour détecter la finitude de (semi-)groupes engendrés par des automates inversibles ou réversibles

	2 lettres 3 états	2 lettres 4 états	3 lettres 3 états
`IsFinite` – FR	0,68	36,36	1 342,12
`IsFinite` – automgrp	0,81	1,79	3,78
`IsFinite1`	0,49	0,52	0,61
`IsFinite2`	0,49	0,62	0,70

Résultats expérimentaux. La table 8.2 présente le temps moyen pour détecter la finitude de (semi-)groupes engendrés par des automates de Mealy inversibles ou réversibles sur p lettres et q états, avec $p+q \in \{5, 6\}$. Pour que ces comparaisons soient équitables, ce qui est donné est le temps minimal pour un automate et son dual.

Difficult to see, always in motion is the future.

The empire strikes back

Cette partie expose mon projet de recherche pour l'avenir, concernant les (semi-)groupes d'automate.

9

Perspectives

Je commence par donner des généralisations possibles pour des résultats de ce mémoire, puis je donne des lignes générales de recherches futures autour des (semi-)groupes d'automates.

1 Des résultats à généraliser

1.1 Décider la finitude

La question de décider la finitude reste toujours ouverte. P. Gillibert a montré très récemment que ce problème est indécidable pour les semi-groupes d'automate [Gil13], mais la question reste ouverte pour les groupes.

Je conjecture le résultat suivant pour les groupes engendrés par des automates biréversibles :

Conjecture 2

Soit un automate de Mealy $\mathcal{A}$ biréversible et $\mathfrak{md}$-réduit à p lettres et q états. Si $q > p!$, alors $\mathcal{A}$ engendre un groupe infini.

Si cette conjecture est vraie, on obtient alors un algorithme simple pour décider la finitude d'un groupe engendré par un automate biréversible. Notons qu'on a très peu d'exemples d'automates biréversibles $\mathfrak{md}$-réduits non triviaux engendrant un groupe fini (à ma connaissance en fait on n'en a globalement qu'un : celui de la figure 6.2 et quelques variations autour de ce même automate).

1.2 Fini ou libre sur deux lettres

Il y a plusieurs façons d'envisager une généralisation du théorème 7.3.

Une première direction consisterait à relier la façon dont les composantes connexes des puissances successives de l'automate grandissent avec la finitude ou la liberté du semi-groupe engendré ; il est assez simple de montrer la généralisation suivante :

Proposition 9.1. Le semi-groupe engendré par un automate de Mealy réversible est fini si et seulement si les composantes connexes de ses puissances sont bornées.

Cette proposition est à mettre en relation avec la borne uniforme sur les graphes en hélice de l'automate (théorème 6.10).

On peut essayer de faire une généralisation à l'autre bout de la fonction de croissance :

Conjecture 3

Si l'automate est connexe, le semi-groupe engendré est libre si et seulement si chaque puissance de l'automate est connexe.

Le théorème 7.3 permet d'affirmer que tout groupe infini engendré par un automate biréversible à deux lettres possède un élément d'ordre infini ; une autre direction pour généraliser ce théorème est alors la suivante :

Conjecture 4

Aucun groupe engendré par un automate biréversible ne peut répondre au problème de Burnside.

1.3 Ordre des éléments

Il existe assez peu de résultats concernant la détermination de l'ordre d'un élément d'un groupe d'automate, notamment quand cet ordre est fini. Je vais m'étendre ici sur plusieurs considérations.

Tout d'abord, il suffit de savoir déterminer l'ordre d'un générateur. En effet, tout élément est générateur d'un groupe engendré par une puissance de l'automate considéré ; on est donc capable, pour tout élément du groupe, d'exhiber un automate dont il est un état.

Un résultat préliminaire intéressant :

Proposition 9.2. Soient un automate inversible $\mathcal{A}$ sur un alphabet Σ et g un élément d'ordre fini de $\langle A \rangle$. La décomposition en facteurs premiers de l'ordre de g ne contient que des nombres inférieurs au sens large au cardinal de Σ.

Démonstration. Soient $\mathcal{A}$ un automate inversible sur un alphabet Σ et $g \in \langle \mathcal{A} \rangle$ et supposons que p premier divise l'ordre de g.

Comme cela a été dit auparavant, on peut se restreindre à g état de $\mathcal{A}$ sans perte de généralité.

En élevant g à la bonne puissance, on obtient un élément d'ordre p, donc on suppose g d'ordre p premier sans perte de généralité.

Considérons maintenant la restriction de g à Σ^n pour un certain n : son ordre divise p. Il existe donc un plus petit $n \geq 1$ tel que cette restriction soit d'ordre p. Soient $\mathbf{u} \in \Sigma^{n-1}$ et $v \in \Sigma$ tel que l'orbite de $\mathbf{u}v$ sous l'action de g soit de taille p. L'orbite de $\mathbf{u}$ sous l'action de g est de taille 1 par hypothèse sur le choix de n, donc l'orbite de v sous l'action de $\delta_{\mathbf{u}}(g)$ est de taille p. Or la taille de cette orbite divise l'ordre de la restriction de $\delta_{\mathbf{u}}(g)$ à Σ qui est une permutation de Σ. □

On en déduit en particulier que l'ordre d'un élément d'un groupe engendré par un automate sur un alphabet de 2 lettres est une puissance de 2.

En s'inspirant de la construction d'A. Antonenko, on peut montrer le résultat suivant :

Proposition 9.3. Soit un automate de Mealy. Si les états dont la fonction de production n'est pas l'identité sont sans branchement, alors toutes les fonctions de production étendues des états d'une même composante connexe ont même ordre.

1.4 Automates à branchement limité

Comme nous l'avons vu, A. Antonenko a montré que quel que soit le choix des fonctions de production, les automates à branchement limité engendrent des semi-groupes finis et que cette famille est maximale pour cette propriété.

Une question naturelle qui vient à l'esprit est de savoir si c'est également une famille maximale sur les groupes, c'est-à-dire si l'on se restreint à des fonctions de production inversibles. Cette question est d'autant plus pertinente que la génération aléatoire de groupes est en plein essor et que les groupes d'automate semblent une alternative crédible pour cela. De fait, les auteurs de [DFN13] s'intéressent à la génération aléatoire de structures à branchement limité, dans le but de générer aléatoirement des groupes finis [DFN13].

2 Lien avec d'autres pans de la théorie des automates

Comme nous l'avons vu tout au long de cette partie dédiée aux (semi-)groupes d'automates, des techniques très classiques (et très basiques) en théorie des automates permettent d'obtenir des résultats non triviaux sur les (semi-)groupes d'automates. On peut donc espérer qu'établir un lien entre les (semi-)groupes engendrés par automate et d'autres notions issue de la théorie des automates soit productif.

Cette partie est essentiellement sous forme interrogative car non explorée pour l'instant.

2.1 Semi-groupe des transitions vs. semi-groupe engendré

Si $\mathcal{A}$ est un automate de Mealy, le semi-groupe des transitions de son dual $\mathfrak{d}(\mathcal{A})$ est un quotient du semi-groupe engendré par $\mathcal{A}$.

Bien entendu, on perd de l'information en passant au semi-groupe des transitions, car on oublie la sortie. Mais on pourrait imaginer travailler sur le couple formé par le semi-groupe des transitions de $\mathcal{A}$ et le semi-groupe des transitions de $\mathfrak{d}(\mathcal{A})$. Peut-on relier des propriétés de ce couple à des propriétés des semi-groupes engendrés par $\mathcal{A}$ et $\mathfrak{d}(\mathcal{A})$?

2.2 Langage des portraits

Les portraits d'un automate de Mealy sur l'alphabet Σ forment un langage d'arbres infinis sur l'alphabet des permutations de Σ. Chacun de ces arbres est régulier car il ne contient qu'un nombre fini de sous-arbres infinis distincts. Le langage lui-même est clairement régulier si le semi-groupe engendré est fini. On peut se demander si le langage est régulier lorsque le semi-groupe engendré est infini ou s'il a d'autres propriétés intéressantes.

2.3 Automates cellulaires

Pour montrer que le problème de finitude est indécidable dans le cas général, P. Gillibert utilise une réduction d'un problème sur les tuiles de Wang, suivant le même schéma que celui utilisé dans [Kar92] par J. Kari pour montrer l'indécidabilité du problème de nilpotence pour les automates cellulaires unidimensionnels.

Une tuile de Wang est un quadruplet de couleurs, une couleur pour chaque point cardinal. On peut la voir comme un carré parallèle aux axes dont chaque côté est coloré. Un pavage de Wang est un assemblage de telles tuiles pour couvrir le plan et qui respecte les couleurs : les côtés adjacents de deux tuiles ont même couleur. Un ensemble de tuiles de Wang est NW-déterministe si étant données deux tuiles x et y dans cet ensemble, il existe au plus une tuile qui peut se placer à l'est de x et au sud de y :

	y
x	$f(x,y)$

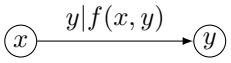
.

J. Kari a montré qu'il est indécidable si un ensemble NW-déterministe de tuiles de Wang peut paver le plan [Kar92].

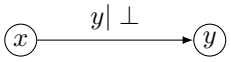
A partir d'un ensemble NW-déterministe A de tuiles de Wang, on construit l'automate de Mealy

$$\mathcal{A} = (A \sqcup \{\perp\}, A \sqcup \{\perp\}, \delta, \rho)$$

donné par les transitions



pour $x, y \in A$ si $f(x,y)$ est défini,



pour ($x = \perp$ ou $y = \perp$) ou $f(x,y)$ non défini.

P. Gillibert montre que l'ensemble de tuiles de Wang A peut paver le plan si et seulement si le semi-groupe $\langle \mathcal{A} \rangle_+$ est fini. La lettre $\perp$ est utilisée pour signaler que le pavage ne peut être continué.

Il résulte de ce travail qu'il y a des ponts assez naturels entre automates de Mealy, tuiles de Wang et automates cellulaires, qu'il serait clairement intéressant d'explorer davantage.

Bibliographie

[AHU74] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[AKL+12] A. Akhavi, I. Klimann, S. Lombardy, J. Mairesse, and M. Picantin. On the finiteness problem for automaton (semi)groups. *Int. J. Algebra Comput.*, 22(4) :26p., 2012.

[AM69] M. F. Atiyah and I. G. MacDonald. *Introduction to Commutative Algebra*. Addison-Wesley, Reading, Massachusetts, 1969.

[Ant08] A. S. Antonenko. On transition functions of Mealy automata of finite growth. *Matematychni Studii.*, 29(1) :3–17, 2008.

[Aut94] J.-M. Autebert. *Théorie des langages et des automates*. Masson, 1994.

[Bar11] L. Bartholdi. FR *Functionally recursive groups, Self-similar groups — a GAP package, Version 1.2.4.2*, 2011.

[BBSZss] I.V. Bondarenko, N.V. Bondarenko, S.N. Sidki, and F.R. Zapata. On the conjugacy problem for finite-state automorphisms of regular rooted trees. *Groups, Geometry, and Dynamics*, in press. arXiv :math.GR/1011.2227.

[BCPS03] M.-P. Béal, O. Carton, Ch. Prieur, and J. Sakarovitch. Squaring transducers : An efficient procedure for deciding functionality and sequentiality. *Theor. Comput. Sci.*, 292 :45–63, 2003.

[Ber79] J. Berstel. *Transductions and context-free languages*. Teubner, 1979.

[BGK+08] I. Bondarenko, R.I. Grigorchuk, R. Kravchenko, Y. Muntyan, V. Nekrashevych, D. Savchuk, and Z. Šunić. On classification of groups generated by 3-state automata over a 2-letter alphabet. *Algebra Discrete Math.*, 1 :1–163, 2008.

[BR84] J. Berstel and Ch. Reutenauer. *Les séries rationnelles et leurs langages*. Masson, 1984. English translation : Rational series and their languages, Springer-Verlag, 1988.

[BS10] L. Bartholdi and P.V. Silva. Groups defined by automata. 2010. arXiv :cs.FL/1012.1531.

[Cai09] A.J. Cain. Automaton semigroups. *Theor. Comput. Sci.*, 410(47-49) :5022–5038, 2009.

[Cho77] Ch. Choffrut. Une caractéristique des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theor. Comput. Sci.*, 5 :325–337, 1977.

[Cho78] Ch. Choffrut. *Contribution à l'étude de quelques familles remarquables de fonctions rationnelles*. PhD thesis, Thèse d'état, Univ. Paris VII, 1978.

[CIvR99] K. Culik II and P. C. von Rosenberg. Generalized weighted finite automata based image compression. *J. UCS*, 5(4) :227–242, 1999.

[CK11] G. Chapuy and I. Klimann. On the supports of recognizable series over a field and a single letter alphabet. *Inf. Process. Lett.*, 111(23-24) :1096–1098, 2011.

[CKO02] Ch. Choffrut, J. Karhumäki, and N. Ollinger. The commutation of finite sets : a challenging problem. *Theoretical Computer Science*, 273 :69–79, 2002.

[Con71] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman Hall, 1971.

[DFN13] S. De Felice and C. Nicaud. Random generation of deterministic acyclic automata using the recursive method. In *CSR*, 2013.

[DKV09] M. Droste, W. Kuich, and H. Vogler. *Handbook of Weighted Automata*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[EC92] D.B.A. Epstein and J.W. Cannon. *Word Processing in Groups*. Ak Peters Series. Jones & Bartlett, 1992.

[Egg63] L.C. Eggan. Transition graphs and the star-height of regular events. *Michigan Math. J.*, 10 :385–397, 1963.

[Eil74] S. Eilenberg. *Automata, languages and machines*. New York Academic Press, 1974.

[GAP08] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.12*, 2008.

[Gau95] S. Gaubert. Performance evaluation of (max,+) automata. *IEEE Trans. on Automatic Control*, 40(12), 1995.

[Gil13] P. Gillibert. The finiteness problem for automaton semigroups is undecidable. *preprint*, 2013.

[GNS00] R.I. Grigorchuk, V.V. Nekrashevich, and V.I. Sushchanskiĭ. Automata, dynamical systems, and groups. *Tr. Mat. Inst. Steklova*, 231 :134–214, 2000.

[Gri80] R. I. Grigorchuk. On Burnside's problem on periodic groups. *Funktsional. Anal. i Prilozhen.*, 14(1) :53–54, 1980.

[Has88] K. Hashiguchi. Algorithms for determining relative star height and star height. *Inf. Comput.*, 78(2) :124–169, 1988.

[HU79] J. Hopcroft and J. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley, 1979.

[Kar92] Jarkko Kari. The nilpotency problem of one-dimensional cellular automata. *SIAM J. Comput.*, 21(3) :571–586, 1992.

[Kir05] D. Kirsten. Distance desert automata and the star height problem. *ITA*, 39(3) :455–509, 2005.

[Kir09] D. Kirsten. An algebraic characterization of semirings for which the support of every recognizable series is recognizable. In *MFCS*, pages 489–500, 2009.

[KL09] D. Kirsten and S. Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *26th STACS*, pages 589–600, 2009.

[Kli13] Ines Klimann. The finiteness of a group generated by a 2-letter invertible-reversible Mealy automaton is decidable. In *30th STACS*, volume 20 of *LIPIcs*, pages 502–513, 2013.

[KLMP04] I. Klimann, S. Lombardy, J. Mairesse, and Ch. Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theor. Comput. Sci.*, 327(3) :349–373, 2004.

[KMP12] I. Klimann, J. Mairesse, and M. Picantin. Implementing computatioons in automaton (semi)groups. In N. Moreira and R. Reis, editors, *CIAA*, number 7381 in LNCS, pages 240–252, 2012.

[KQ11] D. Kirsten and K. Quaas. Recognizability of the support of recognizable series over the semiring of the integers is undecidable. *Inf. Process. Lett.*, 111 :500–502, April 2011.

[Kro94] D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Int. J. Algebra Comput.*, 4(3) :405–425, 1994.

[Kun07] M. Kunc. The power of commuting with finite sets of words. *Theory of Computing Systems*, 40(4) :521–551, juin 2007.

[Lec53] Ch. Lech. A note on recurring series. *Arkiv för Matematik*, 2 :417–421, 1953.

[LS03] S. Lombardy and J. Sakarovitch. On the star height of rational languages : a new presentation for two old results. In *Words, languages & combinatorics, III (Kyoto, 2000)*, pages 266–285. World Sci. Publishing, 2003.

[Mal09] V. Maltcev. Cayley automaton semigroups. *Int. J. Algebra Comput.*, 19(1) :79–95, 2009.

[Min09] A. Mintz. On the Cayley semigroup of a finite aperiodic semigroup. *Int. J. Algebra Comput.*, 19(6) :723–746, 2009.

[Moh97] M. Mohri. Finite-state transducers in language and speech processing. *Comput. Linguist.*, 23(2) :269–311, 1997.

[Mol89] R. A. Mollin. *Number theory and applications*. Kluwer Academic publ., 1989.

[MPR05] M. Mohri, F. Pereira, and M. Riley. Weighted automata in text and speech processing. *CoRR*, abs/cs/0503077, 2005.

[MS08] Y. Muntyan and D. Savchuk. automgrp *Automata Groups — a GAP package, Version 1.1.4.1*, 2008.

[Nek05] V. Nekrashevych. *Self-similar groups*, volume 117 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2005.

[Nov55] P.S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov*, 44 :1–143, 1955. in Russian.

[Pin97] J.-É. Pin. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume 1, chapter 10, pages 679–746. Springer, 1997.

[Reu02] Ch. Reutenauer. *M. Lothaire, Algebraic combinatorics on words*, chapter Centralizers of noncommutative series and po-

lynomials. Cambridge University Press, 2002.

[Sak03] J. Sakarovitch. *Éléments de théorie des automates*. Vuibert informatique, 2003.

[Sak09] J. Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

[Sav09] D. Savchuk. *Asymptotic, algorithmic and geometric aspects of groups generated by automata*. PhD thesis, Texas A&M University, 2009.

[Sch76] M.P. Schützenberger. Sur les relations rationnelles entre monoïdes libres. *Theor. Comput. Sci.*, 3 :243–259, 1976.

[Sid00] S.N. Sidki. Automorphisms of one-rooted trees : growth, circuit structure, and acyclicity. *J. Math. Sci. (New York)*, 100(1) :1925–1943, 2000. Algebra, 12.

[SS05] P.V. Silva and B. Steinberg. On a class of automata groups generalizing lamplighter groups. *Int. J. Algebra Comput.*, 15(5-6) :1213–1234, 2005.

[SV11] D. M Savchuk and Y. Vorobets. Automata generating free produtcs of groups of order 2. *J. Algebra*, 336(1) :53–66, 2011.

[SVV11] B. Steinberg, M. Vorobets, and Y. Vorobets. Automata over a binary alphabet generating free groups of even rank. *Int. J. Algebra Comput.*, 21(1-2) :329–354, 2011.

[VL93] M. Vaughan-Lee. *The restricted Burnside problem*, volume 8 of *London Mathematical Society Monographs. New Series*. Oxford University Press, 1993.

[Web94] A. Weber. Finite-valued distance automata. *Theor. Comput. Sci.*, 134 :225–251, 1994.

[Wei75] A. Weil. *Collected papers – Ernst Eduard Kummer*, volume 1 (Contributions to number theory). Springer-Verlag, 1975.

[WK95] A. Weber and R. Klemm. Economy of description for single-valued transducers. *Information and Computation*, 118(2) :327–340, 1995.

[Zel90] E.I. Zel′manov. Solution of the restricted Burnside problem for groups of odd exponent. *Izv. AN SSSR Math+*, 54(1) :42–59, 221, 1990.

[Zel91] E.I. Zel′manov. Solution of the restricted Burnside problem for 2-groups. *Mat. Sb.*, 182(4) :568–592, 1991.

Table des matières

1 Automates, langages et séries 12
1 Langages et automates 12
2 Structures algébriques 15
3 Séries formelles et automates 16
4 Séquentialité et ambiguïté 19
2 Séquentialité et ambiguïté sur $\mathbb{R}_{\max}$ 22
1 Passage d'un automate finiment ambigu à une union finie d'automates non ambigus 22
2 Décider la non-ambiguité d'une série donnée par un automate finiment ambigu 25
3 Liens entre séries et langages 28
1 Langage support et série caractéristique 28
2 Supports de séries reconnaissables sur une lettre 29
3 Perspectives: problèmes de commutativité 31
4 (Semi-)groupes d'automate 36
1 Automates de Mealy 36
2 (Semi-)groupe engendré par un automate de Mealy et propriétés structurelles de certains automates 37
3 Opérations sur les automates et liens entre les (semi-)groupes engendrés 38
4 Diverses représentations d'un automate de Mealy 43
5 Finitude et autres problèmes 47
1 Problème du mot 47
2 Croissance, ordre et finitude 47
6 Tester la finitude et l'infinitude 50
1 Finitude - l'existant 50
2 Finitude - md-réduction 51
3 Infinitude - l'existant 53
4 Infinitude - graphes en hélice 54
5 Gain de performance 56
6 Critère (non effectif) de finitude 56
7 Le cas à deux lettres ou deux états 59
1 Le semi-groupe engendré est libre ou fini 59
2 Décidabilité de la finitude et de la liberté 62
8 Minimisation et implémentation 68
1 Croissance d'un (semi-)groupe d'automate 68
2 Ordre d'un (semi-)groupe 70
3 Finitude 70
9 Perspectives 74
1 Des résultats à généraliser 74
2 Lien avec d'autres pans de la théorie des automates 76