



HAL
open science

Une introduction aux (semi-)groupes d'automate

Ines Klimann

► **To cite this version:**

| Ines Klimann. Une introduction aux (semi-)groupes d'automate. Master. Une introduction aux (semi-)groupes d'automate, MPRI, 2013, pp.21. cel-01058503

HAL Id: cel-01058503

<https://cel.hal.science/cel-01058503>

Submitted on 27 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Une introduction aux (semi-)groupes d'automate

Ines Klimann klimann@liafa.univ-paris-diderot.fr

MPRI 2012/2013 - filière "Modélisation par automates finis"

1 Introduction

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 Burnside (exemples d'automates très simples engendrant des groupes de torsion infinis finiment engendrés, c'est-à-dire des groupes infinis finiment engendrés dont tous les éléments sont d'ordres finis [stage proposé sur cette thématique, encadré par Matthieu Picantin]) et le problème de Milnor (existence de groupes à croissance intermédiaire).

Les (semi-)groupes d'automate sont un objet d'étude en soi et, comme pour toute famille de (semi-)groupes, on peut se poser des questions de décidabilité concernant les (semi-)groupes d'automate. Le but de ce cours est d'explorer une partie de l'existant sur la décidabilité de la finitude de tels (semi-)groupes. Nous nous centrerons en particulier sur les résultats liés à la structure de l'automate. Il existe d'autres critères provenant de la théorie géométrique des groupes, que nous n'aborderons pas ici.

Pierre Gillibert a montré très récemment que le problème de finitude des semi-groupes d'automate est indécidable [5], tandis que le problème de finitude des groupes sur des automates inversibles-réversibles sur 2 lettres est décidable [7].

Il y a donc deux directions de recherche concernant la finitude des (semi-)groupes d'automates :

- trouver la frontière entre décidable et indécidable [stage proposé sur ce sujet],
- trouver des semi-algorithmes efficaces en moyenne pour tester la finitude ou l'infinitude.

Dans ce cours, nous allons nous concentrer sur le deuxième point : je donnerai des conditions nécessaires ou suffisantes de finitude et nous verrons comment accélérer les calculs pour les semi-algorithmes existant déjà [2, 1, 8].

Pour illustrer la difficulté de l'étude de la finitude des (semi-)groupes d'automates, regardons la figure 1 : ce petit automate engendre un groupe d'ordre $1\,494\,186\,269\,970\,473\,680\,896 = 2^{64} \cdot 3^4 \approx 1.5 \times 10^{21}$.

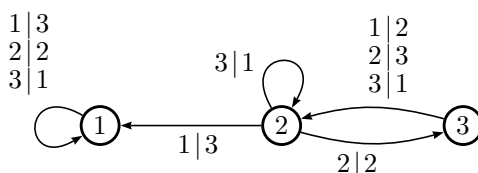


FIGURE 1 – Automate de Mealy engendrant un groupe d'ordre $1\,494\,186\,269\,970\,473\,680\,896$.

2 Premiers éléments

2.1 Automates de Mealy

Soit S un ensemble fini non vide. On note \mathfrak{T}_S l'ensemble des applications de S dans S et \mathfrak{S}_S l'ensemble des permutations de S .

automate

Définition 1. En oubliant les états initiaux et finaux, un *automate* (fini, déterministe et complet) est la donnée d'un triplet

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

où

- l'ensemble des états A est un ensemble fini non vide,
- l'alphabet Σ est un ensemble fini non vide,
- les fonctions de transition δ_i sont des applications : $\delta_i \in \mathfrak{T}_A$.

On identifie cet automate à un élément de \mathfrak{T}_A^Σ .

automate de Mealy

Définition 2. Un automate de Mealy est un quadruple

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

tel que (A, Σ, δ) et (Σ, A, ρ) sont des automates.

Les applications ρ_x sont les *fonctions de production* de l'automate.

La terminologie standard vue jusqu'à présent est *transducteur lettre-à-lettre* séquentiel et complet (avec même alphabet d'entrée et de sortie).

Un automate de Mealy est identifié à un élément de $\mathfrak{T}_A^\Sigma \times \mathfrak{T}_\Sigma^A$.

Les *transitions* d'un automate de Mealy sont les

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

Un automate de Mealy est identifié à son ensemble de transitions.

On utilise la notation graphique usuelle des automates : un graphe dont les sommets sont les états et les arcs correspondent aux transitions de l'automate, voir figure 2.

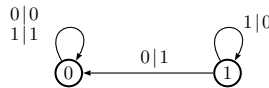


FIGURE 2 – Un automate de Mealy.

2.2 (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. Cet automate peut être vu comme un automate à deux bandes définissant une application de Σ^* vers Σ^* .

fcts prod. étendues

Plus formellement, on construit les *fonctions de production étendues* $\rho_x : \Sigma^* \rightarrow \Sigma^*$ à partir des fonctions

de production $\rho_x : \Sigma \rightarrow \Sigma$. Pour cela, on écrit

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

pour décrire l'existence d'un chemin

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

dans \mathcal{A} .

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

$$\forall u \in \Sigma, \forall \mathbf{v} \in \Sigma^*, \quad \rho_x(u\mathbf{v}) = \rho_x(u)\rho_{\delta_u(x)}(\mathbf{v}). \quad (1)$$

On peut aussi définir les fonctions de production étendues $\rho_x : \Sigma^* \rightarrow \Sigma^*$ par récurrence à l'aide de la formule (1).

Par ailleurs, on peut aussi étendre ces fonctions sur A^* par composition :

$$\rho_{\mathbf{u}} = \rho_{u_1 \cdots u_n} = \rho_{u_n} \circ \cdots \circ \rho_{u_1}.$$

On peut bien entendu faire de même avec les applications $\delta_i : A^* \rightarrow A^*$.

*semi-groupe
engendré*

Définition 3. Le *semi-groupe* $\langle \mathcal{A} \rangle_+$ engendré par $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est le semi-groupe des applications $\Sigma^* \rightarrow \Sigma^*$ engendré par les fonctions de production étendues $\rho_x, x \in A$.

Un semi-groupe est un *semi-groupe d'automate* s'il existe un automate qui l'engendre.

Exemple 1. L'automate de la figure 2 engendre le semi-groupe \mathbb{N} . Soit un mot $u \in \{0, 1\}^*$. On interprète u comme le miroir de l'écriture en base 2 d'un entier, notons \bar{u} cet entier. Alors : $\rho_0(u) = u$ et $\rho_1(u) = v$, où $\bar{v} = \bar{u} + 1$. L'application associée à l'état 0 est donc l'identité et l'application associée à l'état 1 est l'incrément. Le semi-groupe engendré est donc isomorphe à \mathbb{N} .

Si les fonctions de production sont des permutations de Σ , alors les fonctions de production étendues sont des permutations de Σ^* . Elles sont donc inversibles et on peut envisager d'engendrer un groupe.

*automate
inversible*

Définition 4. Un automate de Mealy est *inversible* si ses fonctions de production sont des permutations.

Un automate inversible est identifié à un élément de $\mathfrak{S}_A^\Sigma \times \mathfrak{S}_\Sigma^A$.

*groupe
engendré*

Définition 5. 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 $\rho_x, x \in A$.

Un groupe est un *groupe d'automate* s'il existe un automate qui l'engendre.

Exemple 2. L'automate de la figure 2 engendre le groupe \mathbb{Z} .

*automate
réversible*

Définition 6. Un automate de Mealy est *réversible* si ses fonctions de transition sont des permutations.

Un automate réversible est identifié à un élément de $\mathfrak{S}_A^\Sigma \times \mathfrak{S}_\Sigma^A$. Le terme employé habituellement en théorie des automates est *automate à groupe*.

Un automate *inversible-réversible* est identifié à un élément de $\mathfrak{S}_A^\Sigma \times \mathfrak{S}_\Sigma^A$.

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

2.3.1 Automate inverse

automate
inverse

Définition 7. Soit un automate de Mealy inversible $\mathcal{A} \in \mathfrak{T}_A^\Sigma \times \mathfrak{G}_\Sigma^A$. Soit $A^{-1} = \{x^{-1}, x \in A\}$ une copie disjointe de l'ensemble A des états. L'automate (de Mealy) inverse \mathcal{A}^{-1} de \mathcal{A} est défini par l'ensemble de transitions

$$x^{-1} \xrightarrow{j|i} y^{-1} \in \mathcal{A}^{-1} \iff x \xrightarrow{i|j} y \in \mathcal{A}. \quad (2)$$

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 $\rho_x^{-1} : \Sigma^* \rightarrow \Sigma^*$ associée à l'état x^{-1} de \mathcal{A}^{-1} . On a alors

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

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 (2)). On note i cette opération. Par i , on obtient toujours un transducteur lettre-à-lettre avec même alphabet d'entrée et de sortie, mais ce n'est pas nécessairement un automate de Mealy : c'est un automate de Mealy si et seulement l'automate de départ est inversible, dans ce cas bien entendu : $i(\mathcal{A}) = \mathcal{A}^{-1}$.

automate
biréversible

Définition 8. Un automate de Mealy inversible est *biréversible* si lui et son inverse sont réversibles.

En particulier, et de façon immédiate, un automate biréversible est inversible-réversible.

Proposition 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

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

Démonstration. Le premier point découle directement des définitions.

Supposons que le semi-groupe $\langle \mathcal{A} \rangle_+$ soit fini et soit x un de ses éléments : il existe deux entiers k et n tels que $x^{k+n} = x^k$. On a donc $x^n = 1$ dans le groupe $\langle \mathcal{A} \rangle$. L'inverse x^{n-1} de x appartient donc au semi-groupe $\langle \mathcal{A} \rangle_+$ et on a égalité entre groupe et semi-groupe.

(À noter que cette démonstration est valable pour tout semi-groupe qui est sous-semi-groupe d'un groupe : un tel semi-groupe fini est toujours un groupe.) \square

2.3.2 Automate dual

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

automate
dual

Définition 9. 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

$$i \xrightarrow{x|y} j \in \mathfrak{d}(\mathcal{A}) \iff x \xrightarrow{i|j} y \in \mathcal{A}. \quad (3)$$

Cette définition est consistante : le dual d'un automate de Mealy est bien toujours un automate de Mealy (c'est-à-dire un transducteur lettre-à-lettre séquentiel et complet).

Un automate est réversible si et seulement si son dual est inversible.

Les propositions 2 et 3 suivantes sont complémentaires l'une de l'autre et nous donnent nos premières propriétés liées à la finitude sur les (semi-)groupes d'automate.

Proposition 2 ([1]). Soit deux semi-groupes finis G et H . Il existe un automate de Mealy \mathcal{A} tel que

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

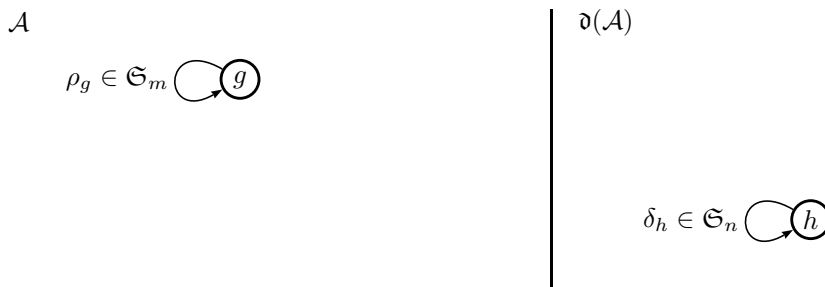
On a un énoncé similaire sur les groupes.

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

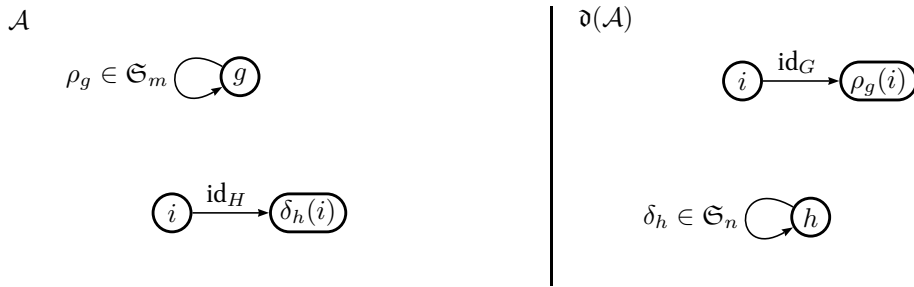
Commençons par une preuve *avec les mains* pour comprendre ce qui se passe.

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 bien entendu en s'assurant que \mathcal{A} engendre G et $\mathfrak{d}(\mathcal{A})$ engendre H .

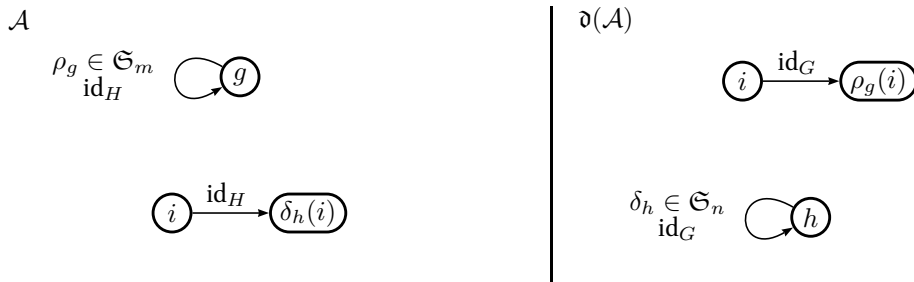
Le groupe G étant fini, il est isomorphe à un sous-groupe de \mathfrak{S}_m pour un certain m . De même, le groupe H est isomorphe à un sous-groupe de \mathfrak{S}_n pour un certain n .



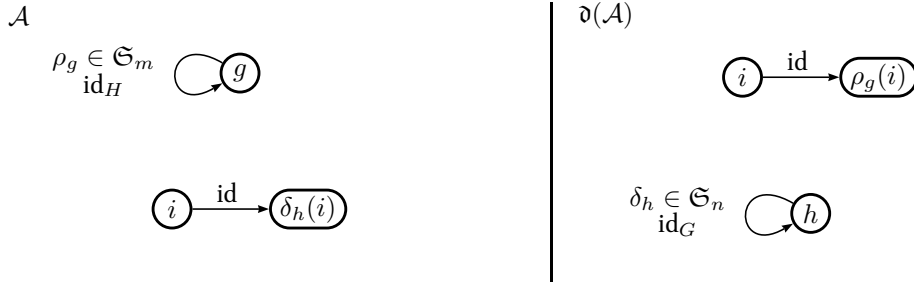
Par dualisation on obtient :



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 :



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



Formalisons la preuve précédente.

Tout groupe fini est un sous-groupe d'un groupe de permutations. Soit Σ_1 et A_2 deux ensembles finis tels que G est un sous-groupe de \mathfrak{S}_{Σ_1} et H est un sous-groupe de \mathfrak{S}_{A_2} . Soit $A_1 \subseteq \mathfrak{S}_{\Sigma_1}$ un ensemble de générateurs de G et $\Sigma_2 \subseteq \mathfrak{S}_{A_2}$ un ensemble de générateurs de H .

On pose $A = A_1 \times A_2$ et $\Sigma = \Sigma_1 \times \Sigma_2$ et on considère l'automate de Mealy d'ensemble d'états A sur l'alphabet Σ dont les transitions sont données par

$$(a, b) \xrightarrow{(i,j)|(a(i),j)} (a, j(b)).$$

On note δ et ρ les fonctions de transition et de production correspondantes. Clairement pour $(a, b) \in A_1 \times A_2$ et $(a, b') \in A_1 \times A_2$, on a $\rho_{(a,b)} = \rho_{(a,b')}$ et on peut noter cette fonction $\rho_a : \Sigma^* \rightarrow \Sigma^*$. On a alors pour tout $a \in A_1$ et tout $(i_1, j_1), \dots, (i_n, j_n) \in \Sigma^*$:

$$\rho_a((i_1, j_1) \cdots (i_n, j_n)) = (a(i_1), j_1) (a(i_2), j_2) \cdots (a(i_n), j_n).$$

Ainsi le groupe engendré par $(\rho_a : \Sigma^* \rightarrow \Sigma^*)_{a \in A_1}$ est isomorphe au groupe engendré par $(a : \Sigma_1 \rightarrow \Sigma_1)_{a \in A_1}$, c'est-à-dire $\langle \mathcal{A} \rangle = G$. De la même façon $\langle \mathfrak{d}(\mathcal{A}) \rangle = H$. \square

Proposition 3 ([10, 12, 1]). 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. La preuve est faite pour les semi-groupes, elle s'étend aux groupes directement par la proposition 1.

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 :

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

pour tout $u_1 u_2 \cdots u_n \in \Sigma^*$. 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 $\langle \mathfrak{d}(\mathcal{A}) \rangle_+$ par rapport aux lettres de Σ :



Or il n'y a qu'un nombre fini de tels transducteurs, égal au nombre d'applications de $\langle \mathfrak{d}(\mathcal{A}) \rangle_+$ vers \mathfrak{T}_{Σ} . On en conclut

$$\#\langle \mathcal{A} \rangle_+ \leq (\#\Sigma)^{(\#\Sigma)} (\#\langle \mathfrak{d}(\mathcal{A}) \rangle_+).$$

\square

2.3.3 Automates étendus

Soit \mathcal{A} un automate inversible-réversible. On a vu en proposition 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.

automate
étendu

Définition 10. Soit \mathcal{A} un automate inversible-réversible. 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 $\Sigma \sqcup \Sigma^{-1}$:

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

Le corollaire suivant est une conséquence des propositions 1 et 3.

Corollaire 1. 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 3, il engendre un groupe d'ordre 16 et son automate étendu engendre un groupe d'ordre 64.

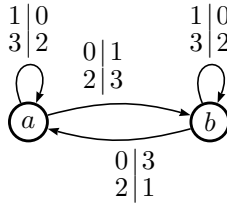


FIGURE 3 – Automate inversible-réversible engendrant le groupe $K_4 \rtimes \mathbb{Z}_2$ d'ordre 16.

2.3.4 Automates d'ordres supérieurs

aut. d'ordre
supérieur

Définition 11. Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho) \in \mathfrak{T}_A^\Sigma \times \mathfrak{T}_\Sigma^A$ et deux entiers $n, k > 0$.

L'automate de Mealy

$$\mathcal{A}_{n,k} = (A^n, \Sigma^k, (\delta_{\mathbf{x}} : A^n \rightarrow A^n)_{\mathbf{x} \in \Sigma^k}, (\rho_{\mathbf{u}} : \Sigma^k \rightarrow \Sigma^k)_{\mathbf{u} \in A^n}) \quad (4)$$

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

Il s'identifie à un élément de $\mathfrak{T}_{A^n}^{\Sigma^k} \times \mathfrak{T}_{\Sigma^k}^{A^n}$.

Dans l'équation (4), $\rho_{\mathbf{u}} : A^n \rightarrow A^n$ est la restriction de $\rho_{\mathbf{u}} : A^* \rightarrow A^*$ à A^n , et de même pour $\delta_{\mathbf{x}}$. On a en particulier $\mathcal{A}_{1,1} = \mathcal{A}$.

Graphiquement, $\mathcal{A}_{n,k}$ est un automate dont les états sont des mots de longueur n sur A dans le semi-groupe engendré par \mathcal{A} et les actions de ces états correspondent aux actions des éléments du semi-groupe sur des mots de longueur k sur Σ .

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

puissance
d'un aut.

L'automate $\mathcal{A}_{n,1}$ est parfois noté \mathcal{A}^n et appelé puissance n -ème de \mathcal{A} .

3 Problème du mot

Le premier problème 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 [11].

Ce problème est décidable dans le cadre des semi-groupes d'automate, comme montré en proposition 4, ce qui rend le problème de finitude semi-décidable par énumération.

Lemme 1. Étant donné un automate de Mealy, on peut décider si les fonctions de production étendues de deux de ses états sont égales.

Je ne détaille pas la preuve ici, mais il suffit de regarder la procédure de minimisation introduite en section 4 pour s'en convaincre.

Proposition 4. Le problème du mot est décidable pour les (semi-)groupes d'automate.

Démonstration. Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$. 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 groupe est l'identité, ce qui permet de considérer le problème du mot sur des mots de même longueur.

Soit les générateurs $u_1, u_2, \dots, u_n, v_1, v_2, \dots, v_n \in A$ pour lesquels on se demande si

$$u_1 u_2 \cdots u_n \stackrel{?}{=} v_1 v_2 \cdots v_n.$$

On se place dans l'automate puissance \mathcal{A}^n : $u_1 u_2 \cdots u_n$ et $v_1 v_2 \cdots v_n$ sont des états de cet automate. On applique donc le lemme 1 pour obtenir le résultat. \square

Cependant, même pour de petits automates, les (semi-)groupes engendrés peuvent être grands, comme montré dans l'introduction avec l'automate de la figure 1.

Dans la suite, on s'intéresse à des constructions permettant de décider de la finitude ou de l'infinitude d'un (semi-)groupe engendré par automate.

4 Critère de finitude : la mδ-réduction

On construit ici un critère reposant sur la notion de minimisation d'un automate.

congruence **Définition 12.** Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$. Une équivalence \equiv sur A est une *congruence* pour \mathcal{A} si

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

équivalence de Nérode L'équivalence de Nérode sur A est la congruence la plus fine pour \mathcal{A} .

L'équivalence de Nérode est la limite de la suite d'équivalences de plus en plus fines (\equiv_k) définie par

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

L'ensemble des 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.

On note $[x]$ la classe d'équivalence d'un état $x \in A$ pour l'équivalence de Nérode.

automate
minimisé

Définition 13. Soit un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ et \equiv l'équivalence de Nérode 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 équivalent à son minimisé.

La définition est consistante avec la définition classique de minimisation sur les automates booléens : la partition initiale a lieu ici en fonction des fonctions de production ; pour rappel, dans le cas des automates booléens elle se fait sur le critère états finaux / états non finaux.

Proposition 5. Un automate de Mealy et son minimisé engendrent le même semi-groupe.

Démonstration. On montre par récurrence sur n que ρ_x et $\tilde{\rho}_{[x]}$ sont égaux sur Σ^n .

Laissé en exercice. □

On remarque que le dual d'un minimisé n'est pas nécessairement minimal. On introduit ici une notion de minimalité symétrique entre un automate et son dual.

paire / aut.
m ∂ -réduit(e)
m ∂ -réduction

Définition 14. Une paire d'automates duaux est *m ∂ -réduite* si chacun des deux automates de la paire est minimal. Par extension, on dira qu'un automate est *m ∂ -réduit* si la paire qu'il forme avec son dual est m ∂ -réduite.

La *m ∂ -réduction* d'une paire d'automates duaux consiste à réduire alternativement chacun des deux automates jusqu'à ce que la paire soit m ∂ -réduite.

Exemple 3. Un exemple de m ∂ -réduction est donné en figure 4.

Même si elle ne le semble pas au premier abord, la m ∂ -réduction est confluente [1]. Ce fait n'est pas crucial pour la suite, mais facilitera les tournures de phrase en nous permettant de donner la définition suivante.

m ∂ -réduit

Définition 15. La paire d'automates de Mealy obtenue par m ∂ -réduction d'un couple d'automates duaux est appelée son *m ∂ -réduit*.

Théorème 1 ([1]). Une paire d'automates duaux engendre des (semi-)groupes finis si et seulement si son m ∂ -réduit engendre des (semi-)groupes finis.

Démonstration. Laissée en exercice. □

On note par ailleurs que $\partial\text{m}\partial(\mathcal{A})$ est un quotient de \mathcal{A} . Donc si le groupe engendré par \mathcal{A} est fini, celui engendré par $\partial\text{m}\partial(\mathcal{A})$ est plus petit.

Le théorème 1 n'est pas en lui-même un critère de finitude puisqu'il faut savoir si le m ∂ -réduit obtenu engendre des groupes finis. Néanmoins il peut être efficacement combiné à d'autres critères de finitude. On déduit de ce théorème une condition suffisante de finitude effective donnée par le corollaire 2, en remarquant que l'automate trivial engendre le groupe trivial.

Corollaire 2 ([1]). Si la m ∂ -réduction d'une paire d'automates de Mealy duaux aboutit à une paire d'automates triviaux, les automates de départ engendrent des (semi-)groupes finis.

La démonstration de la proposition 6 ci-dessous est une application directe de ce critère.

Il existe des paires m ∂ -réduites non triviales d'automates duaux qui engendrent des (semi-)groupes finis. Un exemple est donné en figure 5.

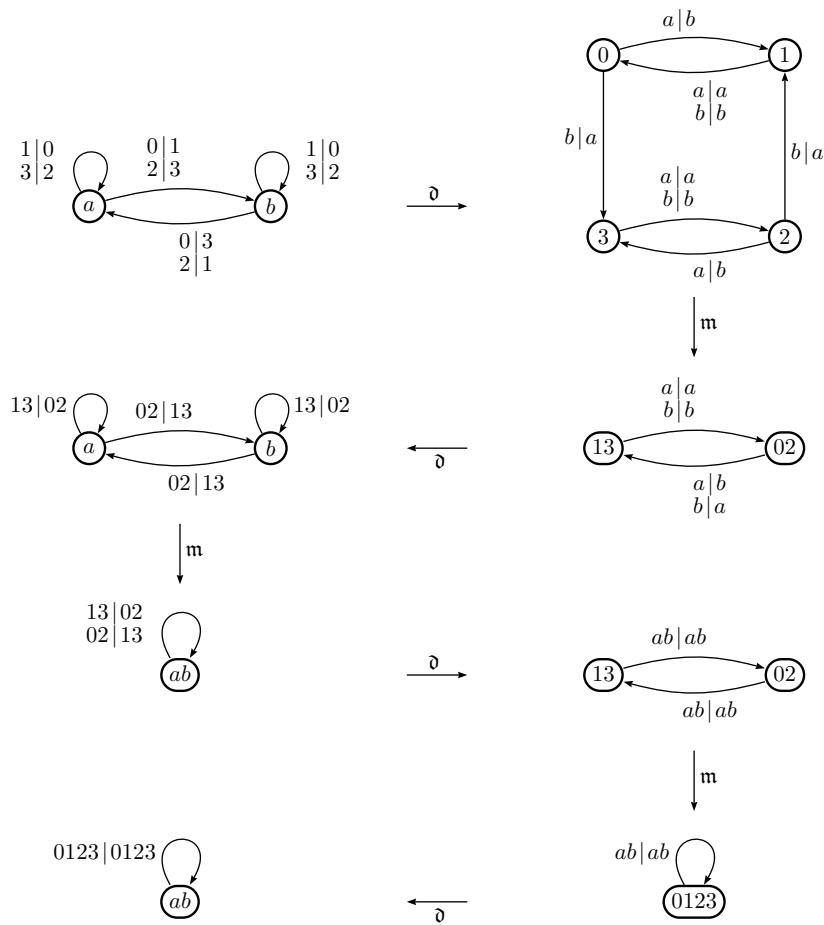


FIGURE 4 – La $m\delta$ -réduction d'une paire d'automates de Mealy duaux.

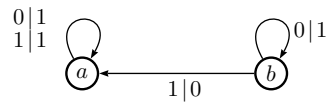


FIGURE 5 – Automate $m\delta$ -réduit (non trivial) qui engendre un semi-groupe de taille 6.

5 Critère structurel de finitude : branchement limité

Antonenko [2] s'est intéressé au problème suivant : quels sont les automates de Mealy tels que pour toutes les fonctions de production possibles, le semi-groupe engendré est fini ?

Les critères développés dans [2] reposent sur la structure de l'automate. La proposition 7 donne le résultat général. Il est cependant plus intuitif d'étudier en premier la proposition 6.

*état sans
branchement*

Définition 16. 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 transition part de l'état x , étiquetée par toutes les lettres de l'alphabet Σ .

aut. sans
branchement

Définition 17. Un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est *sans branchement* si tous ses états sont sans branchement.

On pourra alors en abusant légèrement noter $\delta(x)$ l'image d'un état $x \in A$ par une des fonctions de transition δ_i .

Proposition 6 ([2]). Un automate de Mealy sans branchement engendre un (semi-)groupe fini.

Démonstration. Soit un automate de Mealy sans branchement. Tous les états de son dual sont équivalents, le m δ -réduit de la paire est donc une paire d'automates triviaux et on peut conclure par le corollaire 2.

[Ce n'est pas la démonstration donnée dans [2].] □

Le résultat de la proposition 6 s'étend aux automates dont aucun branchement n'est suivi d'un cycle.

aut. à branchement
limité

Définition 18. Un automate de Mealy $\mathcal{A} = (A, \Sigma, \delta, \rho)$ est à *branchement limité* si tous ses états atteignables à partir d'un cycle sont sans branchement.

Proposition 7 ([2]). Un automate de Mealy à branchement limité engendre un (semi-)groupe fini.

Démonstration. avec les mains

Quitte à dérouler un peu les cycles, on peut supposer que toutes les branches qui précèdent l'entrée dans un cycle ont même longueur b . En s'autorisant à agrandir ces mêmes cycles (en les parcourant un certain nombre de fois), on peut également supposer que tous les cycles ont même longueur ℓ .

On découpe alors un mot en ses b premières lettres, puis en paquets de ℓ lettres consécutives. Sur les b premières lettres, il n'y a qu'un nombre fini d'actions possibles. Sur la suite du mot : sur chaque paquet de ℓ lettres, il n'y a qu'un nombre fini d'actions possibles et au bout d'un moment on a forcément vu toutes les actions qui devaient apparaître.

Formalisation laissée en exercice. □

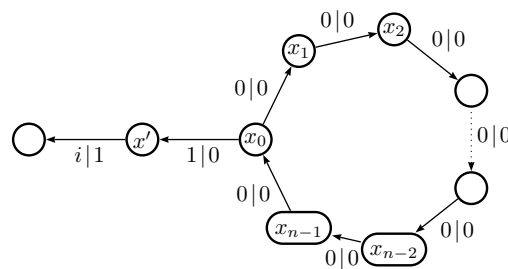
Bien entendu ce critère ne couvre pas tous les automates engendrant des (semi-)groupes finis. Par exemple, l'automate de la figure 1 n'est pas à branchement limité et engendre un semi-groupe fini.

Ce critère est maximal dans le sens où pour tout automate qui n'est pas à branchement limité, il existe une fonction de production telle que le groupe engendré est infini.

La démonstration se fait en traitant séparément les trois cas suivants :

- (i) il existe un cycle \mathcal{C} étiqueté uniquement par des 0 et tel qu'un des états de ce cycle branche vers un état extérieur à \mathcal{C} ;
- (ii) il existe un cycle \mathcal{C} étiqueté uniquement par des 0 et tel qu'un des états de ce cycle branche vers un état de \mathcal{C} ;
- (iii) les cycles qui possèdent des états qui branchent ne sont pas uniquement étiquetés par des 0.

Montrons le cas (i) pour donner une idée :



Soit deux entiers k et ℓ tels que $k + \ell$ soit divisible par n . Alors $\rho_{x_k}(0^\ell 1 v) = 0^{\ell+1} 1 v'$.

On montre que l'élément suivant du groupe engendré par l'automate est d'ordre infini :

$$\bar{\rho} = \rho_{x_1} \circ \rho_{x_2} \circ \dots \circ \rho_{x_{n-1}} \circ \rho_{x_0}.$$

En effet pour un entier k quelconque, on a

$$\begin{aligned} \bar{\rho}(0^{kn} 1 \dots) &= \rho_{x_1} \circ \rho_{x_2} \circ \dots \circ \rho_{x_{n-1}} \circ \rho_{x_0}(0^{kn} 1 \dots) \\ &= \rho_{x_1} \circ \rho_{x_2} \circ \dots \circ \rho_{x_{n-1}}(0^{kn+1} 1 \dots) \\ &= \dots \\ &= 0^{(k+1)n} 1 \dots \end{aligned}$$

Donc au final :

$$\bar{\rho}^k(1^\omega) = 0^{kn} 1 \dots .$$

6 Critère d'infinitude : graphe en hélice

Dans cette partie nous travaillons exclusivement sur des automates inversibles-réversibles. Nous introduisons de nouvelles représentations d'automates de Mealy permettant de considérer un automate et son dual simultanément.

graphe en hélice

Définition 19. 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 $(x, i) \rightarrow (\delta_i(x), \rho_x(i))$.

On remarque qu'on peut définir un graphe en hélice pour tout transducteur lettre-à-lettre ayant même alphabet d'entrée et de sortie. Un tel transducteur est un automate de Mealy si et seulement si de tout sommet part un unique arc.

Proposition 8. Si le groupe engendré par un automate inversible-réversible est fini, alors son graphe en hélice est une union de cycles disjoints.

Pour montrer ce résultat, nous avons besoin d'une autre représentation d'une paire d'automates duaux.

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

$$x \begin{array}{c} \begin{array}{c} i \\ \downarrow \\ \hline \end{array} \rightarrow \delta_i(x) \\ \downarrow \\ \rho_x(i) \end{array} .$$

transition en croix

Cette notation est appelée *transition en croix*. Un automate de Mealy est identifié à l'ensemble de ses transitions en croix (de cardinalité $|A| \times |\Sigma|$).

diagramme en croix

Un chemin dans un automate de Mealy (*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 $\mathcal{A}_{n,k}$ d'ordre (n, k) et de son dual.

Par exemple, le diagramme en croix suivant :

$$\begin{array}{ccc} x_1 & \begin{array}{c} \begin{array}{c} i_1 \\ \downarrow \\ \hline \end{array} \rightarrow \dots \rightarrow \begin{array}{c} \begin{array}{c} i_k \\ \downarrow \\ \hline \end{array} \\ \downarrow \\ \rho_x(i) \end{array} & y_1 \\ \vdots & & \vdots \\ x_n & \begin{array}{c} \begin{array}{c} \downarrow \\ \hline \end{array} \rightarrow \dots \rightarrow \begin{array}{c} \begin{array}{c} \downarrow \\ \hline \end{array} \\ \downarrow \\ \rho_x(i) \end{array} & y_n \\ j_1 & & j_k \end{array}$$

correspond dans $\mathcal{A}_{n,k}$ à

$$\rho_{x_1 \dots x_n}(i_1 \dots i_k) = j_1 \dots j_k,$$

$$\delta_{i_1 \dots i_k}(x_1 \dots x_n) = y_1 \dots y_n .$$

(ii) \Rightarrow (i) L'automate \mathcal{A} étant supposé inversible-réversible, $\partial\mathcal{A}$ est bien un automate de Mealy.

Par ailleurs, $\partial\partial\mathcal{A}$ étant un automate de Mealy, $\partial\mathcal{A}$ est également un automate de Mealy. Comme c'est l'inverse de $\partial\mathcal{A}$, on en déduit que $\partial\mathcal{A}$ est inversible, donc \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 $(y^{-1}, j^{-1}) \rightarrow (x^{-1}, i^{-1})$ si $(x, i) \rightarrow (y, j)$ appartient au graphe en hélice \mathcal{H} de \mathcal{A} .

Le graphe \mathcal{G} est le graphe en hélice de $\partial\partial\mathcal{A}$:

- si $\partial\partial\mathcal{A}$ 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 19 que $\partial\partial\mathcal{A}$ est un automate de Mealy.

□

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

Corollaire 3. Tout automate inversible-réversible qui n'est pas biréversible engendre un groupe infini.

7 Condition nécessaire et suffisante de finitude (non effective)

Dans cette partie nous travaillons exclusivement sur des automates inversibles-réversibles. Le critère présenté ici n'est à ce jour pas effectif, cette section sert donc quelque peu du cadre général de ce cours, mais les techniques utilisées dans les preuves peuvent s'avérer intéressantes dans d'autres occasions.

graphes en hélice

Définition 20. Soit 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} .

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

On peut noter que le graphe en hélice de \mathcal{A} tel que défini à la définition 19 est le graphe en hélice d'ordre $(1, 1)$ de \mathcal{A} .

Théorème 2. 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 2, nous allons utiliser des résultats intermédiaires.

Lemme 2. 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.

La démonstration de ce lemme repose sur le même type d'argument que celle de la proposition 8 et est laissée en exercice.

Proposition 10. Si un automate inversible-réversible engendre un groupe fini, alors les cycles des graphes en hélices de son automate étendu sont uniformément bornés.

Démonstration. Soit \mathcal{A} un automate inversible-réversible engendrant un groupe fini et $\tilde{\mathcal{A}}$ son automate étendu. D'après le corollaire 1, le groupe engendré par $\tilde{\mathcal{A}}$ est fini et d'après le lemme 2, ses graphes en hélices sont des unions de cycles disjoints.

D'après la proposition 3, le groupe $\langle \partial(\tilde{\mathcal{A}}) \rangle$ est également fini.

Soit \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 $\langle \partial(\tilde{\mathcal{A}}) \rangle$). Comme les sommets sont deux à deux distincts, la longueur du cycle \mathcal{C} est bornée par $\#(\tilde{\mathcal{A}}) \times \#(\partial(\tilde{\mathcal{A}}))$. \square

Proposition 11. Si les cycles des graphes en hélice de l'automate étendu d'un automate inversible-réversible sont 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 qui permet d'affirmer qu'un groupe d'automate dont les ordres des éléments sont bornés¹ est fini.

mot unitaire

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).

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 $\rho_{\mathbf{x}_n}^k(\mathbf{u}_k) = \mathbf{u}'_k \neq \mathbf{u}_k$.

Comme $\langle \partial(\tilde{\mathcal{A}}) \rangle$ 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

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

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

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 : $(\mathbf{x}_n, \rho_{\mathbf{x}_n}(\mathbf{w}_n))$, $(\mathbf{x}_n, \rho_{\mathbf{x}_n}^2(\mathbf{w}_n))$, ... 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. \square

Le théorème 2 est alors un corollaire des propositions 10 et 11.

8 Accélération des semi-algorithmes existants

Cette section décrit comment la minimisation permet d'accélérer substantiellement les procédures existantes servant à tester la finitude et à calculer la croissance ou l'ordre d'un (semi-)groupe fini [8].

Notre point de départ sont les deux paquets GAP [4] suivants : FR développé par Bartholdi [3] et automgrp développé par Muntyan et Savchuk [9].

8.1 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

longueur

$\mathcal{A} = (A, \Sigma, \delta, \rho)$ et un mot $x \in A^*$. La longueur de ρ_x , notée $|\rho_x|$, est la longueur minimale d'un mot équivalent à x :

$$|\rho_x| = \min\{n \mid \exists y \in A^n, \rho_x = \rho_y\}.$$

fonction de croissance

La fonction de croissance de l'automate \mathcal{A} est la série formelle qui énumère les éléments du semigroupe 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 groupe soit engendré par un automate ne sert qu'à apporter une solution au problème du mot.

On peut en fait utiliser 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 :

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

où $\mathcal{A}' = \mathcal{A}$ si l'identité est un générateur de \mathcal{A} , 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'états 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 $\langle \text{aut} \rangle_+$, tandis que l'appel AutomatonGrowth(aut+aut⁻¹) calcule la croissance du groupe $\langle \text{aut} \rangle$.

Résultats expérimentaux. Nous commençons par faire tourner AutomatonGrowth et WordGrowth (paquet FR) sur l'automate de Grigorchuk. Pour un rayon de 10, AutomatonGrowth est nettement plus rapide, 76 ms contre 9 912 ms². La raison en est simple : WordGrowth appelle la procédure de minimisation 57 577 fois tandis que AutomatonGrowth ne l'appelle que 12 fois :

1. Il existe des groupes d'automate infinis dont les éléments sont tous d'ordre fini, par exemple le groupe engendré par l'automate de Grigorchuk [6].
2. Programmes exécutés sur un processeur Intel Core 2 Duo 3,06 GHz.

```

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 1 faute d'avoir pu les obtenir en temps raisonnable.

TABLE 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

8.2 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 décrite ici 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 en quelque sorte 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 1. Ni la fonction `Order` de `FR`, ni `AutomSGOrder` ne sont capables de calculer l'ordre du groupe, alors que `automgrp`, *via* `LevelOfFaithfulAction`, permet d'obtenir le résultat en seulement 14 338 ms. D'un autre côté, `AutomSGOrder` calcule l'ordre du semi-groupe engendré par le dual en 17 ms, alors qu'une adaptation de `LevelOfFaithfulAction` (aux automates de Mealy non inversibles) met 2 193 ms (ce semi-groupe est d'ordre 234).

8.3 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 puis essayent en dernier recours de calculer l'ordre du groupe.

On peut saupoudrer 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 implémentations La conception de la procédure `IsFinite1` est consistante avec celle de `AutomatonGrowth`. Ainsi `IsFinite1` est beaucoup plus proche de `FR` que de `automgrp`. La version écrite ici 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 2 – Temps moyen (en ms) pour détecter la finitude de (semi-)groupes

	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

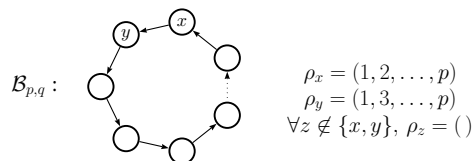
Résultats expérimentaux. La table 2 présente le temps moyen pour détecter la finitude de (semi-)groupes engendrés par des automates de Mealy inversibles ou réversibles sur p lettres et q états, avec $p+q \in \{5, 6\}$. Pour que ces comparaisons soient équitables, ce qui est donné est le temps minimal pour un automate et son dual.

9 Cas à deux états

Si on fixe le nombre d'états et la taille de l'alphabet, le nombre d'automates de Mealy correspondant est potentiellement très grand : il y a $q^{qp}(p!)^q$ automates de Mealy inversibles à q états et p lettres. Si on regarde à isomorphisme de structure près, il y a 76 automates de Mealy à 2 états et 2 lettres et 10 766 772 automates de Mealy à 3 états et 3 lettres. Engendrer ces automates est un défi en soi.

L'accélération des calculs permet de faire des tests systématiques sur de grandes familles d'automates. Ces tests permettent de vérifier et/ou d'élaborer des conjectures. Voici deux conjectures établies par calculs dans [8] :

Conjecture 1. Tout groupe fini engendré par un automate biréversible sur p lettres et q états est d'ordre au plus $\#\langle \mathcal{B}_{p,q} \rangle = p!^q$, où :



Si cette conjecture se révèle vraie, la problème de finitude serait décidable pour les automates biréversibles.

Une autre conjecture (devenue depuis un résultat) obtenue par expérimentation :

Théorème 3. Un automate de Mealy réversible à deux états engendre un semi-groupe fini ou libre.

Nous allons montrer ici une partie de ce résultat.

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

degré de
connexion

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 . Donc si \mathcal{A}^n n'est pas connexe, il en est de même de toutes les puissances suivantes de \mathcal{A} . Il existe donc 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 7) est réversible, possède trois états et a un degré de connexion égal à 2 ; il engendre le semi-groupe infini non libre $\mathbb{Z}_2^{*3} = \mathbb{Z}_2 * \mathbb{Z}_2 * \mathbb{Z}_2$ [10]. Ainsi le théorème 3 ne s'étend pas à des ensembles d'états plus grands.

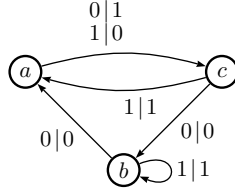


FIGURE 7 – L'automate Baby Aleshin engendre le groupe $\mathbb{Z}_2^{*3} = \mathbb{Z}_2 * \mathbb{Z}_2 * \mathbb{Z}_2$ [10].

Proposition 12. 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.

Ce résultat se montre par une série de petits résultats.

Lemme 3. 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. \square

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 .

Lemme 4. 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 : $N \leq \#\mathcal{D} \leq 2N$.

Soient \mathbf{v} appartenant à \mathcal{C} et $z, \bar{z} \in A, z \neq \bar{z}$: $\mathbf{u}x$ 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. D'où le résultat. \square

Lemme 5. 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 4 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 4 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 4, l'existence d'une composante connexe de \mathcal{A}^{m-1} de taille 2^{n-1} . \square

Preuve de la proposition 12. 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 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 prêt, il ne peut donc y avoir qu'un nombre fini de composantes connexes distinctes. On en déduit que $\langle \mathcal{A} \rangle_+$ est fini.

La réciproque est un cas particulier du lemme 3. \square

Références

- [1] A. Akhavi, I. Klimann, S. Lombardy, J. Mairesse, and M. Picantin. On the finiteness problem for automaton (semi)groups. *International Journal of Algebra and Computation*, (accepted), 2011. <http://arxiv.org/abs/1105.4725>.
- [2] A. S. Antonenko. On transition functions of Mealy automata of finite growth. *Matematychni Studii*, 29(1) :3–17, 2008.
- [3] L. Bartholdi. FR *Functionally recursive groups, Self-similar groups — a GAP package, Version 1.2.3*, 2011.
- [4] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.12*, 2008.
- [5] P. Gillibert. communication personnelle, 2012.
- [6] R. I. Grigorchuk. On Burnside's problem on periodic groups. *Funktsional. Anal. i Prilozhen.*, 14(1) :53–54, 1980.
- [7] I. Klimann. The finiteness of a group generated by a 2-letter invertible-reversible Mealy automaton is decidable. In *STACS*, 2013. <http://arxiv.org/abs/1208.6324>.
- [8] I. Klimann, J. Mairesse, and M. Picantin. Implementing computations in automaton (semi)groups. In N. Moreira and R. Reis, editors, *CIAA*, number 7381 in LNCS, pages 240–252, 2012.
- [9] Y. Muntyan and D. Savchuk. *automgrp Automata Groups — a GAP package, Version 1.1.4.1*, 2008.
- [10] V. Nekrashevych. *Self-similar groups*, volume 117 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2005.
- [11] P.S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov*, 44 :1–143, 1955. in Russian.
- [12] D. M Savchuk and Y. Vorobets. Automata generating free products of groups of order 2. *J. Algebra*, 336(1) :53–66, 2011.