UNIVERSITE DE LIEGE
Faculté des Sciences
Institut de Mathématique

Les groupes automatiques

Mémoire présenté par
Julien LEROY
en vue de l'obtention
du grade de licencié en
sciences mathématiques

Année académique 2006–2007

Mes premiers remerciements vont bien entendu à Monsieur M. Rigo, non seulement pour son aide et ses précieux conseils tout au long de ce travail, mais aussi pour m'avoir donné l'envie de poursuivre ma formation mathématique.

Je tiens également à remercier mes proches pour leur soutient et leurs encouragements.

Enfin, merci à Anne pour son aide et sa patience.

Introduction

Les automates finis sont un outil classique en théorie des langages formels. Il s'agit d'algorithmes élémentaires (ne nécessitant qu'une mémoire finie) qui permettent de tester l'apparition de certains critères syntaxiques dans tout mot fourni en entrée. La notion même de mot se rencontre dans divers contextes mathématiques, par exemple, en dynamique symbolique : une trajectoire est codée par la suite des régions traversées, ou encore, en théorie des groupes : un produit arbitraire d'éléments (par exemple, si on dispose d'un nombre fini de générateurs) peut être interprété comme un mot (sur un alphabet fini, celui des générateurs).

Dans ce mémoire, nous considérons ce dernier exemple et nous définissons la classe des *groupes automatiques*. Il s'agit de groupes finiment générés pour lesquels on dispose d'automates finis réalisant deux types de tests. Le premier consiste à vérifier si deux mots quelconques w_1 et w_2 écrits sur l'alphabet des générateurs représentent effectivement le même élément du groupe, l'automate correspondant est qualifié d'*automate égalité*. Le second consiste quant à lui à vérifier, pour tout générateur g, si deux mots quelconques w_1 et w_2 écrits sur l'alphabet des générateurs sont tels que $w_1 g$ et w_2 représentent le même élément. On parlera alors d'*automates multiplicateurs*. L'ensemble de ces données, à savoir les générateurs et les automates correspondants, constitue une *structure automatique* sur le groupe et cette dernière détermine univoquement le groupe. Nous la voyons donc comme une présentation du groupe.

D'une manière générale, les présentations de groupes par générateurs et relations sont assez difficiles à manipuler. À l'opposé et bien évidemment, grâce aux automates qui les sous-tendent, les présentations par une structure automatique permettent de construire relativement aisément des algorithmes pour résoudre bon nombre de problèmes, notamment le problème du mot.

Ce travail s'intéresse donc aux structures automatiques et à certaines de leurs variantes et s'agence plus spécifiquement comme suit.

Le premier chapitre de ce travail rappelle les définitions et les résultats fondamentaux de la théorie des automates et des langages formels et les généralise au cas plus général des langages à plusieurs variables. Nous introduisons des automates sur plusieurs alphabets et nous définissons alors une certaine régularité sur les langages à n variables. Nous nous intéressons ensuite aux prédicats sur un alphabet et nous montrons la correspondance qu'il existe entre les langages et les prédicats. Enfin, nous développons le cas particulier des langages à 2 variables sur un couple d'alphabets identiques,

c'est-à-dire aux *relations* sur l'ensemble des mots sur un alphabet, ce que nous utilisons énormément puisque les langages acceptés par les automates multiplicateurs et égalité sont des relations sur l'ensemble des mots écrits sur l'alphabet des générateurs.

Le deuxième chapitre est principalement axé sur l'ouvrage de C. Choffrut intitulé *A short introduction to automatic group theory* et traite de la notion la plus habituellement rencontrée de groupe automatique, à savoir les groupes automatiques *synchrones*. Dans ce type de groupe automatique, les automates multiplicateurs et égalité qui sont utilisés sont synchrones en ce sens qu'ils lisent les deux mots d'entrée à la même vitesse.

Nous utilisons dans ce chapitre le concept de *graphe de Cayley* d'un groupe qui est la généralisation des tables de Cayley des groupes finis aux groupes infinis discrets et nous définissons une distance sur ce graphe. Nous introduisons alors la *propriété du compagnon de voyage* qui stipule que si deux mots écrits sur l'alphabet des générateurs représentent des éléments qui ne diffèrent que de la multiplication à droite d'au plus un générateur, alors les chemins respectifs qu'ils décrivent dans le graphe de Cayley sont *proches* l'un de l'autre à tout instant.

Cette propriété est une clef de bon nombre de propriétés pour ces groupes. Nous démontrons par exemple que le caractère automatique d'un groupe est *propre* à ce groupe et ne dépend donc pas de l'ensemble des générateurs choisis. Nous fournissons également plusieurs méthodes afin de simplifier la structure automatique d'un groupe, en utilisant par exemple l'ordre lexicographique. Enfin, nous démontrons certains résultats concernant la stabilité de l'ensemble des groupes automatiques synchrones, comme par exemple la stabilité pour le produit direct de deux groupes automatiques ou encore le passage aux sous-groupes d'index fini. En particulier, nous démontrons que le problème du mot est décidable en un temps quadratique en la longueur de l'instance.

Le troisième et dernier chapitre est basé sur l'ouvrage de David B. A. Epstein intitulé *Word Processing in Groups* et développe une classe de groupes plus large que celle des groupes automatiques synchrones, à savoir celle des groupes automatiques *asynchrones*. La différence est que, contrairement au cas synchrone, les automates multiplicateurs et égalité que nous utilisons sont *asynchrones* en ce sens qu'ils ne lisent pas les deux mots d'entrée à la même vitesse. Nous démontrons cependant que la différence entre les vitesses de lecture des deux mots peut être supposée bornée ; nous appellons ces automates des *automates asynchrones à lecture bornée*.

La majorité des résultats démontrés au deuxième chapitre pour les groupes automatiques synchrones restent vrais pour les groupes automatiques asynchrones ; les démonstrations sont cependant pour la plupart fort différentes. Néanmoins, même si le problème du mot reste décidable pour ces groupes, il n'existe pas encore d'algorithme qui permet de le décider avec une complexité temporelle polynomiale. De plus, d'autres problèmes qui sont décidables pour les groupes automatiques synchrones ne le sont plus. Nous terminons le chapitre en donnant un exemple de groupe automatique asynchrone mais pas synchrone ; la classe des groupes automatiques synchrones est donc un sous-ensemble strict de l'ensemble des groupes automatiques asynchrones.

Chapitre 1

Automates et relations

Dans ce chapitre, nous rappelons les notions élémentaires de la théorie des automates et langages formels comme les mots finis, les langages réguliers ou la notion même d'automate. Nous étendons ensuite la notion de langage à celle des *langages à n variables*. Nous définissons alors une certaine régularité sur ces langages et nous montrons la correspondance qu'il existe entre les langages et les prédicats. Nous nous intéressons alors au cas particulier des *relations* sur l'ensemble des mots sur un alphabet.

1.1 Premières définitions

Les éléments d'un alphabet fini Σ sont des *lettres* ou *symboles*.

Une suite finie d'éléments de Σ est appelée un *mot* sur Σ. Nous notons par la juxtaposition

$$w_0 w_1 \cdots w_n$$

la suite $(w_0, w_1, \ldots, w_n)$.

La *longueur* d'un mot w est le nombre de symboles constituant ce mot ; nous la notons $|w|$. L'unique mot de longueur zéro est le mot correspondant à la suite vide. Ce mot s'appelle le *mot vide* et se note ε.

L'ensemble des mots sur Σ est muni de l'opération de *concaténation* qui aux mots $u = u_0 u_1 \cdots u_l$ et $v = v_0 v_1 \cdots v_k$ associe le mot $uv = u_0 u_1 \cdots u_l v_0 v_1 \cdots v_k$. Cette opération est associative et possède un neutre, le mot vide. Muni de cette opération, l'ensemble des mots jouit d'une structure de monoïde. Nous l'appelons le *monoïde libre (engendré par Σ)* et nous le notons Σ^*. Nous développons dans la section 2.1 la notion de groupe (resp. monoïde) libre.

Soit u un mot sur Σ. Le mot x sur Σ est un *préfixe* (resp. *suffixe*) de u s'il existe un mot v sur Σ tel que $u = xv$ (resp. $u = vx$). De la même manière, x est un *facteur* de u s'il existe deux mots v et w sur Σ tels que $u = vxw$. L'ensemble des préfixes (resp. suffixes) d'un mot u est noté $\mathrm{Pref}(u)$ (resp. $\mathrm{Suff}(u)$).

Un sous-ensemble L du monoïde Σ^* est appelé *langage*. La concaténation de deux langages L et M est simplement le langage

$$LM = \{uv | u \in L, v \in M\}.$$

Nous étendons la notion de *préfixe* et de *suffixe* aux langages en définissant, pour tout langage L,

$$\mathrm{Pref}(L) = \bigcup_{w \in L} \mathrm{Pref}(w)$$

et

$$\mathrm{Suff}(L) = \bigcup_{w \in L} \mathrm{Suff}(w)$$

Nous disons alors qu'un langage L est *préfixiel* (resp. *suffixiel*) si $\mathrm{Pref}(L) = L$ (resp. $\mathrm{Suff}(L) = L$).

Soient u et v deux mots sur un alphabet Σ. Nous définissons le *shuffle* de u et v par le langage

$$u \sqcup\!\sqcup v = \{u_1 v_1 \cdots u_n v_n \in \Sigma^* \mid u = u_1 \cdots u_n,\ v = v_1 \cdots v_n,\ u_i,\ v_i \in \Sigma^*,\ n \geq 1\}.$$

Par exemple, si $u = abc$ et $v = xy$ nous avons

$$u \sqcup\!\sqcup v = \{abcxy, abxcy, abxyc, axbcy, axbyc, axybc, xabcy, xabyc, xaybc, xyabc\}.$$

Le *shuffle* de deux langages L et M est le langage

$$L \sqcup\!\sqcup M = \bigcup_{(u,v) \in L \times M} u \sqcup\!\sqcup v.$$

Un *automate fini déterministe* (ou AFD) sur l'alphabet Σ est la donnée d'un quadruple $\mathcal{A} = (Q, q_0, F, \delta)$, où

i) Q est un ensemble fini dont les éléments sont les états de l'automate ;

ii) $q_0 \in Q$ est un état privilégié appelé état initial ;

iii) $F \subseteq Q$ désigne l'ensemble des états finals ;

iv) $\delta : Q \times \Sigma \longrightarrow Q$ est la fonction de transition de l'automate.

Nous étendons naturellement la fonction δ à $Q \times \Sigma^*$ en posant

$$\delta(q, \varepsilon) = q$$

et

$$\delta(q, \sigma w) = \delta(\delta(q, \sigma), w), \forall \sigma \in \Sigma, w \in \Sigma^*.$$

Nous notons parfois par $q.a$ la transition $\delta(q, a)$ pour $a \in \Sigma$. Si w est un mot sur Σ, alors $q.w$ désigne l'ensemble des états accessibles à partir de l'état q en lisant le mot w.

Le langage accepté par un automate fini déterministe $\mathcal{A}$ est l'ensemble

$$L(\mathcal{A}) = \{w \in \Sigma^* | \delta(q_0, w) \in F\}.$$

Un *automate fini non déterministe* (ou AFND) sur un alphabet Σ est la donnée d'un quadruple $\mathcal{A} = (Q, I, F, \Delta)$, où I est l'ensemble des états initiaux, Q et F sont définis comme dans le cas déterministe et où $\Delta \subseteq Q \times \Sigma^* \times Q$ est l'ensemble des transitions de l'automate.

Le langage accepté par un automate fini non déterministe $\mathcal{A}$ est l'ensemble des mots acceptés par cet automate, un mot w étant accepté s'il existe un chemin dans le graphe associé à $\mathcal{A}$ débutant dans un état initial, d'exécution w et aboutissant dans un état final.

Le théorème de Rabin et Scott stipule que l'ensemble des langages acceptés par AFND coïncide exactement avec l'ensemble des langages acceptés par AFD. Dès lors, dans la suite de ce travail, nous ne précisons en général pas le caractère déterministe ou non déterministe des automates envisagés. Nous pouvons cependant toujours supposer disposer d'un unique état initial et que tous les états sont accessibles à partir de cet état initial ; nous disons qu'un tel atomate est *normalisé*. Nous supposons également pouvoir toujours atteindre un des états finals à partir de n'importe quel état, i.e. nous avons éliminé les puits éventuels de l'automate ; nous appelons cela un automate *partiel*. Pour uniformiser les notations, nous nous autoriserons le cas échéant à définir les transitions d'un AFD $\mathcal{A} = (Q, q_o, F, \delta)$ sur l'alphabet Σ par l'ensemble de ses transitions $\Delta = \{(q, \sigma, \delta(q, \sigma)) | q \in Q, \sigma \in \Sigma\}$

Un langage L est dit *régulier* s'il existe un automate fini $\mathcal{A}$ tel que $L = L(\mathcal{A})$.

Nous disposons des résultats suivants concernant les langages réguliers ; leur démonstration ne seront cependant pas développées dans ce travail[1].

Théorème 1.1.1. *L'ensemble des langages réguliers est stable pour les opérations booléennes, la concaténation et pour l'image directe et inverse par morphisme.*

En pratique, si nous disposons de deux automates déterministes $\mathcal{A}$ et $\mathcal{A}'$ acceptant respectivement les langages L et L', il existe une méthode très commode pour construire un automate déterministe acceptant les langages $L \cup L'$ ou $L \cap L'$. Pour cela, si nous notons par Q et Q' les ensembles d'états respectifs de $\mathcal{A}$ et de $\mathcal{A}'$, alors nous considérons l'ensemble $Q \times Q'$ comme ensemble d'états. Nous définissons ensuite les transitions de l'automate par $(q, q').a = (q.a, q'.a)$. Si q_0 et q'_0 sont les états initiaux respectifs de $\mathcal{A}$ et de $\mathcal{A}'$, alors nous choisissons comme état initial le couple (q_0, q'_0). Enfin, si F et F' sont les ensembles d'états finals de $\mathcal{A}$ et de $\mathcal{A}'$ respectivement, nous choisissons comme ensemble d'états finals l'ensemble

$$\{(f, f') \in Q \times Q' \mid f \in F,\ f' \in F'\}$$

[1] Voir [4] pour les démonstrations.

pour que l'automate accepte l'intersection des langages et l'ensemble

$$\{(q, q') \in Q \times Q' \mid q \in F\} \cup \{(q, q') \in Q \times Q' \mid q' \in F'\}$$

pour que l'automate en accepte l'union.

Proposition 1.1.2. *Si L est un langage régulier, alors $Pref(L)$ et $Suff(L)$ sont également des langages réguliers.*

Proposition 1.1.3. *Si L et M sont des langages réguliers, alors $L \sqcup\!\sqcup M$ est également un langage régulier.*

1.2 Prédicats et langages réguliers

Il existe une correspondance entre les prédicats à une variable et les langages sur un alphabet. Nous définissons ici les *langages réguliers à n variables* afin d'obtenir cette même correspondance avec les prédicats à n variables.

Soit Σ un alphabet. Un *prédicat* P sur Σ est une fonction à valeurs booléennes sur Σ^*, i.e. qui ne peut prendre que les valeurs *vrai* (T) et *faux* (F).

Par exemple, $P_1 =$ "Le mot w a un nombre paire de a" est un prédicat sur l'alphabet $\{a, b\}$.

Il existe une correspondance entre les prédicats sur un alphabet et les langages sur ce même alphabet. En effet, si L est un langage sur Σ, nous pouvons définir le prédicat P_L comme la fonction sur Σ^* qui à un mot w sur Σ associe la valeur T si le mot appartient au langage et F sinon. Réciproquement, si P est un prédicat sur Σ, alors nous définissons le langage L_P par l'ensemble

$$L_P = \{w \in \Sigma^* \mid P(w) = T\}.$$

Ainsi, nous disons qu'un prédicat P est *régulier* si le langage correspondant L_P est régulier.

Par exemple, le prédicat "w est la représentation d'un multiple de 3" est un prédicat régulier sur l'alphabet $\{0, 1\}$. En effet, il est clair que l'automate représenté à la figure 1.1 accepte son langage correspondant.

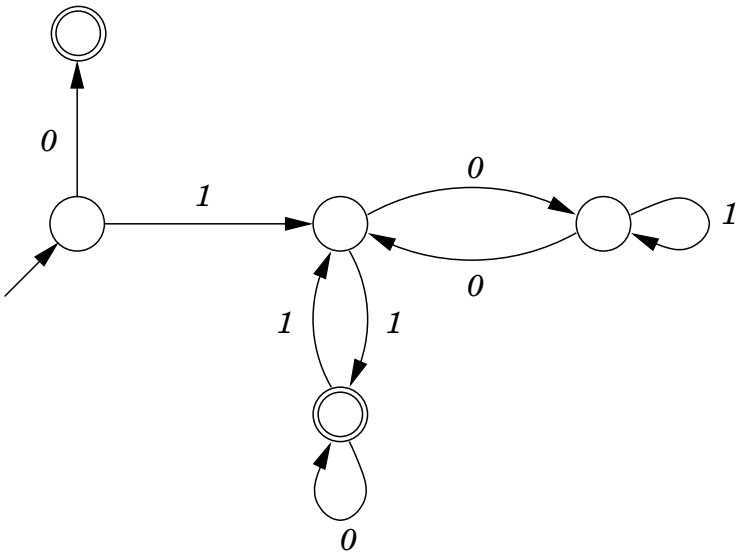

FIG. 1.1 – Automate acceptant les représentations binaires des multiples de 3

Les opérateurs du premier ordre sur les prédicats sont $\neg$, $\wedge$, $\vee$, $\exists$ et $\forall$. Si nous considérons les prédicats comme des langages, nous étendons ces opérateurs comme

suit : si L et L' sont deux langages sur Σ, alors $\neg L = \Sigma^* \backslash L$, $L \wedge L' = L \cap L'$ et $L \vee L' = L \cup L'$. Les quantificateurs $\forall$ et $\exists$ sont définis uniquement pour les prédicats à plusieurs variables ; nous les définissons dans la suite.

Le lemme suivant stipule que l'ensemble des prédicats réguliers est fermé pour les opérateurs $\neg$, $\wedge$ et $\vee$.

Lemme 1.2.1. *Si L et L' sont deux langages réguliers sur l'alphabet Σ, alors les langages $\neg L$, $L \wedge L'$ et $L \vee L'$ sont également réguliers*

Démonstration. Le résultat est évident vu les définitions des opérateurs données ci-dessus. □

Pour définir l'équivalence des quantificateurs $\forall$ et $\exists$ dans les langages, nous devons étendre la définition d'un langage à celle d'un langage à plusieurs variables.

Ainsi, si $\Sigma_1, \ldots, \Sigma_n$ sont des alphabets, $n \geq 2$, nous appellons *langage à n variables* sur $(\Sigma_1, \ldots, \Sigma_n)$ toute partie L de $\Sigma_1^* \times \cdots \times \Sigma_n^*$

Un *prédicat à n variables* sur $(\Sigma_1, \ldots, \Sigma_n)$ est une fonction à valeurs booléennes sur $\Sigma_1^* \times \cdots \times \Sigma_n^*$.

La correspondance qui existe entre les prédicats et les langages est toujours valable pour les prédicats et langages à n variables.

Nous pouvons maintenant étendre les quantificateurs $\forall$ et $\exists$ à l'ensemble des langages à n variables. Ainsi, si L est un langage à n variables sur $(\Sigma_1, \ldots, \Sigma_n)$, nous définissons les langages à $n-1$ variables $\exists(L)$ et $\forall(L)$ par

$$\exists(L) = \{(w_1, \ldots, w_{n-1}) \in \Sigma_1^* \times \cdots \times \Sigma_{n-1}^* \mid \exists w \in \Sigma_n^* \ : \ (w_1, \ldots, w_{n-1}, w_n) \in L\}$$

et

$$\forall(L) = \{(w_1, \ldots, w_{n-1}) \in \Sigma_1^* \times \cdots \times \Sigma_{n-1}^* \mid \forall w \in \Sigma_n^* \ : \ (w_1, \ldots, w_{n-1}, w_n) \in L\}.$$

Nous pourrions maintenant nous demander s'il existe une certaine régularité parmi les prédicats à n variables en ce sens que le langage associé à un prédicat est accepté par un automate quelconque. Cependant, si un n-uple $(w_1, \ldots, w_n)$ de mots appartient à un langage à n variables $L \subset \Sigma_1^* \times \cdots \times \Sigma_n^*$, tous les mots du n-uple n'ont pas nécessairement la même longueur et il est donc impossible d'en vérifier l'appartenance à un langage par automate. Il serait donc intéressant de parler de *mot de n-uples de lettres* plutot que de n-uple de mots. Ainsi, si toutes les composantes d'un n-uple de mots ont une même longueur, il devient possible de vérifier l'appartenance à un langage par un automate dont les labels de transition seraient des n-uples de symboles.

Considérons donc pour tout $1 \leq i \leq n$, l'alphabet $\Sigma_i^{\$_i} = \Sigma_i \cup \{\$_i\}$, où le symbole $\$_i$ n'appartient pas à l'alphabet Σ_i ; nous l'appelons le *symbole de fin de mot*. Pour tout n-uple $(w_1, \ldots, w_n)$ de mots, nous considérons alors le n-uple $(w_1, \ldots, w_n)^{\$} = (w_1\$_1^{k_1}, \ldots, w_n\$_n^{k_n})$, où pour tout i, nous avons $k_i = \max\{|w_j| \mid 1 \leq j \leq n\} - |w_i|$. En d'autres termes, à chaque mot du n-uple, nous ajoutons des occurences du symbole

de fin de mot correspondant à l'alphabet du mot jusqu'à obtenir une distance de $\max\{|w_j| \mid 1 \leq j \leq n\}$; tous les mots du n-uple ont donc une même longueur minimale.

Par exemple, au couple $(monsieur, madame)$ de mots, nous associons le couple $(monsieur, madame)^{\$} = (monsieur, madama\$_2\$_2)$ et nous pouvons donc l'écrire sous la forme d'un mot dont les lettres sont des couples de lettres, i.e. nous l'écrivons sous la forme

$$(m,m)(o,a)(n,d)(s,a)(i,m)(e,e)(u,\$_2)(r,\$_2)$$

et nous pouvons alors vérifier, grâce à un automate, s'il appartient effectivement à un langage à deux variables.

Nous pouvons maintenant passer à la définition d'un langage régulier à n variables. Si L est un langage à n variables sur $(\Sigma_1, \ldots, \Sigma_n)$, nous considérons le langage associé

$$L^{\$} = \{(w_1, \ldots, w_n)^{\$} \in \Sigma_1^{\$_1} \times \cdots \times \Sigma_n^{\$_n} \mid (w_1, \ldots, w_n) \in L\}.$$

Nous disons alors qu'un langage à n variables sur $(\Sigma_1, \ldots, \Sigma_n)$ est *régulier* si le langage associé $L^{\$}$ est accepté par un automate sur $\Sigma_1^{\$_1} \times \cdots \times \Sigma_n^{\$_n}$.

Nous disposons du théorème suivant dont le premier point est la généralisation du lemme 1.2.1 pour les prédicats à n variables.

Théorème 1.2.2. *Soient L et L' deux langages réguliers à n variables sur $(\Sigma_1, \ldots, \Sigma_n)$. Les conditions suivantes sont satisfaites.*

1. *Les langages à n variables $\neg L$, $L \wedge L'$ et $L \vee L'$ sont réguliers ;*
2. *Les langages à $n-1$ variables $\forall(L)$ et $\exists(L)$ sont réguliers sur $(\Sigma_1, \ldots, \Sigma_{n-1})$;*
3. *Pour tout alphabet Σ_{n+1}, le langage à $n+1$ variables*

$$\{(w_1, \ldots, w_n, w_{n+1}) \in \Sigma_1^* \times \cdots \times \Sigma_{n+1}^* \mid (w_1, \ldots, w_n) \in L\}$$

   est régulier sur $(\Sigma_1, \ldots, \Sigma_{n+1})$.
4. *Pour toute permutation ν de $\{1, \ldots, n\}$, le langage à n variables*

$$L_\nu = \{(w_{\nu(1)}, \ldots, w_{\nu(n)}) \in \Sigma_{\nu(1)}^* \times \cdots \times \Sigma_{\nu(n)}^* \mid (w_1, \ldots, w_n) \in L\}$$

   est régulier sur $(\Sigma_{\nu(1)}, \ldots, \Sigma_{\nu(n)})$.

Démonstration. Le premier point est évident ; les constructions développées après le théorème 1.1.1 s'appliquent toujours.

Montrons que le langage à $n-1$ variables $\exists(L)$ est régulier. Soit $\mathcal{A}$ un automate sur $\Sigma_1^{\$_1} \times \cdots \times \Sigma_n^{\$_n}$ acceptant le langage à n variables $L^{\$}$. Nous construisons un automate $\mathcal{A}'$ sur $\Sigma_1^{\$_1} \times \cdots \times \Sigma_{n-1}^{\$_{n-1}}$ qui accepte $\exists(L)$ comme suit. Il s'agit en fait du même automate que $\mathcal{A}$, à la différence que nous avons remplacé tous les labels de transition, qui sont des n-uples dans $\mathcal{A}$, par des $(n-1)$-uples en supprimant simplement la dernière composante du n-uple.

Cet automate accepte clairement le langage à $n-1$ variables $\exists(L)$ et puisque

$$\forall = \neg \circ \exists \circ \neg,$$

le langage à $n-1$ variables $\forall(L)$ est également régulier.

Les deux derniers points sont évidents, vu la forme des automates qui acceptent un langage à n variables.

□

Corollaire 1.2.3. *La classe des prédicats réguliers est fermée pour les opérations* $\neg$, $\wedge$, $\vee$, $\exists$ *et* $\forall$.

Vu le quatrième point du théorème précédent, intervertir les composantes d'un langage régulier à n variables le laisse régulier ; il est donc possible de quantifier sur n'importe quelle composante du langage et pas uniquement sur la dernière.

Le troisième point nous permet également de considérer des composantes qui ne figurent pas explicitement dans le prédicat. Nous souhaitons par exemple montrer que le langage à 3 variables

$$\{(w_1, w_2, w_3) \in \Sigma^* \times \Sigma^* \times \Sigma^* \mid (w_3 < w_2) \ \wedge \ ((w_2, w_3, w_1) \in L)\},$$

où L est un langage régulier à 3 variables et où $<$ représente une relation qui peut être vérifiée par automate, est régulier sur (Σ, Σ, Σ). Nous pouvons écrire ce langage sous la forme

$$\{(w_1, w_2, w_3) \in \Sigma^* \times \Sigma^* \times \Sigma^* \mid w_3 < w_2\} \cap \{(w_1, w_2, w_3) \in \Sigma^* \times \Sigma^* \times \Sigma^* \mid (w_2, w_3, w_1) \in L\}.$$

La composante w_1 ne figure pas explicitement dans le membre de droite de cette expression et les variables ont été permutées dans le membre de gauche. Il suffit donc d'appliquer les points 4 puis 3 au premier membre et le point 4 au second. Le premier point nous permet alors de conclure.

1.3 Relations

Nous nous intéressons ici au cas particulier des langages à deux variables sur un couple (Σ, Σ) d'alphabets identiques, i.e. aux *relations* sur Σ^*.

Une *relation (binaire)* sur le monoïde libre Σ^* est un sous-ensemble R de $\Sigma^* \times \Sigma^*$.

Nous pouvons par exemple citer les relations $R_1 = \{(u, v) \in \Sigma^* \times \Sigma^* \mid |u| = |v|\}$ et $R_2 = \{(u, v) \in \Sigma^* \times \Sigma^* \mid u < v\}$, où $<$ représente un ordre sur le monoïde Σ^*.

Comme nous l'avons fait pour les langages à n variables, nous nous intéressons aux relations qui pourraient être acceptées par des automates et à chaque couple (u, v) de mots sur Σ, nous associons donc le couple $(u, v)^\$$ de mots sur $\Sigma^\$ = \Sigma \cup \{\$\}$ défini, comme pour les n-uples de mots, par

$$(u, v)^\$ = \begin{cases} (u\$^n, v) \ si \ n = |v| - |u| \geq 0, \\ (u, v\$^n) \ si \ n = |u| - |v| \geq 0. \end{cases}$$

Par exemple, aux couples $(aabb, ba)$ et (cb, bac) de mots sur l'alphabet $\Sigma = \{a, b, c\}$, nous associons respectivement les couples $(aabb, ba\$\$)$ et $(cb\$, bac)$.

Nous disons que le couple de mots (z, t) sur Σ est un *préfixe synchrone* du couple de mots (u, v) si z et t sont respectivement préfixes de u et de v et si en plus nous avons

$$|z| = |t| \leq min\{|u|, |v|\}$$

ou

$$min\{|z|, |t|\} = min\{|u|, |v|\}.$$

Dans l'exemple précédent, les préfixes synchrones de $(aabb, ba)$ et (cb, bac) sont respectivement

$$(\varepsilon, \varepsilon), (a, b), (aa, ba), (aab, ba), (aabb, ba)$$

et

$$(\varepsilon, \varepsilon), (c, b), (cb, ba), (cb, bac).$$

Nous pouvons maintenant appliquer ce qui vient d'être fait aux relations binaires. À la relation binaire R sur le monoïde Σ^*, nous associons la relation

$$R^\$ = \{(u, v)^\$ | (u, v) \in R\}.$$

Cet ensemble peut alors être vu comme un sous-ensemble du monoïde libre $(\Sigma^\$ \times \Sigma^\$)^*$ et nous nous intéressons alors à savoir s'il est ou non accepté par un automate fini sur $\Sigma^\$ \times \Sigma^\$$. Ainsi, nous disons que la relation R sur Σ^* est *synchrone* si le langage à deux variables[2] R est régulier sur (Σ, Σ).

[2]Nous pourrions également généraliser ce qui vient d'être fait aux relations p-aires pour $p \geq 2$. Une *relation p-aire* sur le monoïde libre Σ^* est un sous-ensemble R de $(\Sigma^*)^p$. Ici aussi, nous complétons chaque p-uple de mots sur Σ par des occurences du symbole \$ de telle sorte que tous les mots du p-uple aient une même longueur minimale et une composante de longueur maximale sans \$. Ainsi, nous disons que la relation R sur Σ^* est *synchrone* si c'est un langage régulier à p variables sur $(\Sigma)^p$.

Par exemple, la relation

$$R = \{(a^i, b^i) | i \geq 0\}$$

est clairement synchrone sur l'alphabet $\{a, b\} \times \{a, b\}$. En effet, il est facile de voir que l'automate représenté à la figure 1.2 accepte la relation R. Notons que dans cet exemple, nous avons $R = R^{\$}$.

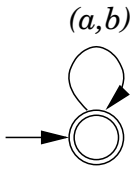

FIG. 1.2 – AFND acceptant le langage $R = \{(a^i, b^i) | i \geq 0\}$ sur l'alphabet $\{a, b\} \times \{a, b\}$.

Si R et S sont deux relations sur le monoïde Σ^*, alors la composée $R \circ S$ est la relation

$$R \circ S = \{(u, v) \in \Sigma^* \cup \Sigma^* \mid \exists\ w \in \Sigma^* : (u, w) \in R \ \ et \ \ (w, v) \in S\}.$$

Les relations synchrones satisfont beaucoup de propriétés. Nous ne développons ici que celles qui sont nécessaires à la suite. Le théorème suivant est simplement une généralisation du théorème 1.1.1 aux relations synchrones.

Théorème 1.3.1. *L'ensemble des relations synchrones est stable pour les opérations booléennes et pour la composition de relations. La projection d'une relation synchrone sur chacune des composantes est un langage régulier. De plus, si L est un langage régulier, alors la relation $\{(u, u) \mid u \in L\}$ est synchrone. Enfin, si M et L sont des langages réguliers, alors $M \times L$ est une relation synchrone.*

Démonstration. La stabilité pour les opérations booléennes est une conséquence du théorème 1.2.2.

Considérons un automate fini acceptant la relation synchrone R. Les projections π_1 et π_2 définies par $\pi_1(a, b) = a$ et $\pi_2(a, b) = b$, $a, b \in \Sigma^{\$}$, sont des morphismes entre $(\Sigma^{\$} \times \Sigma^{\$})^*$ et $(\Sigma^{\$})^*$. Le langage à deux variables R étant régulier sur (Σ, Σ), le langage $R^{\$}$ est accepté par un automate sur $\Sigma^{\$} \times \Sigma^{\$}$. Nous concluons en utilisant le théorème 1.1.1.

Soit maintenant L un langage régulier sur Σ et soit $\mathcal{A}$ un automate fini déterministe acceptant L. L'automate fini construit à partir de $\mathcal{A}$, où l'on a remplacé chaque label de transition a par le label

$$\begin{pmatrix} a \\ a \end{pmatrix},$$

accepte la relation $\{(u, u) | u \in L\}$ et celle-ci est donc synchrone.

Ensuite, notons par $\mathcal{A}$ et $\mathcal{B}$ deux automates sur l'alphabet Σ acceptant respectivement les langages L et M. Nous pouvons supposer sans restriction que ces deux automates sont déterministes. Nous avons

$$\mathcal{A} = (Q_{\mathcal{A}}, q_{0,\mathcal{A}}, F_{\mathcal{A}}, \delta_{\mathcal{A}})$$

et

$$\mathcal{B} = (Q_{\mathcal{B}}, q_{0,\mathcal{B}}, F_{\mathcal{B}}, \delta_{\mathcal{B}}).$$

Un automate sur l'alphabet $\Sigma^{\$} \times \Sigma^{\$}$ acceptant le langage à deux variables $(L \times M)^{\$}$ est l'automate ayant

i) $Q_{\mathcal{A}} \times Q_{\mathcal{B}}$ comme ensemble d'états ;

ii) $(q_{0,\mathcal{A}}, q_{0,\mathcal{B}})$ comme état initial ;

iii) $F_{\mathcal{A}} \times F_{\mathcal{B}}$ comme ensemble d'états finals ;

et où la fonction de transition δ définie sur $(Q_{\mathcal{A}} \times Q_{\mathcal{B}}) \times (\Sigma^{\$} \times \Sigma^{\$})$ et à valeurs dans $(Q_{\mathcal{A}} \times Q_{\mathcal{B}})$ est définie par

$$\begin{cases} \delta((q,q'),(a,b)) = (\delta_{\mathcal{A}}(q,a), \delta_{\mathcal{B}}(q',b)), \\ \delta((q,q'),(a,\$)) = (\delta_{\mathcal{A}}(q,a), q'), \\ \delta((q,q'),(\$,b)) = (q, \delta_{\mathcal{B}}(q',b)). \end{cases}$$

Cet automate ne fait en fait que simuler le comportement de $\mathcal{A}$ sur sa première composante et celui de $\mathcal{B}$ sur sa deuxième composante.

Enfin, montrons la stabilité pour la composition. Soient deux relations synchrones R et S sur Σ^*. Considérons alors les relations tertiaires $R \times \Sigma^*$ et $\Sigma^* \times S$. Par le point précédent, ces deux relations sont clairement synchrones. Nous concluons en remarquant que

$$R \circ S = \pi_{1,3}((R \times \Sigma^*) \cap (\Sigma^* \times S)),$$

où $\pi_{1,3}$ est la projection sur la première et la troisième composante.

En effet, il est clair que $R \circ S$ est inclus dans le membre de droite. Réciproquement, tout couple (u,v) apartenant à $\pi_{1,3}((R \times \Sigma^*) \cap (\Sigma^* \times S))$ est la projection par $\pi_{1,3}$ d'un triplet (u,w,v) appartenant à $(R \times \Sigma^*) \cap (\Sigma^* \times S)$. Nous avons donc un mot w sur Σ qui est tel que

$$\begin{cases} (u,w) \in R \\ (w,v) \in S \end{cases}$$

et le couple (u,v) appartient donc à la relation $R \circ S$.

□

Chapitre 2

Groupes automatiques synchrones

Nous introduisons dans ce chapitre les *groupes automatiques synchrones*, ou plus simplement *groupes automatiques*. Nous montrons comment un alphabet peut générer un groupe et comment nous pouvons définir une certaine structure, appelée *structure automatique*, sur un groupe par l'intermédiaire d'un alphabet. Nous développons ensuite quelques résultats concernant ces groupes. Nous définissons également le *graphe de Cayley* d'un groupe sur lequel nous définissons une distance, ce qui nous permet d'illustrer certaines propriétés des groupes automatiques.

2.1 Groupes et langages

Dans cette section, nous montrons comment il est possible de générer un groupe gâce à un alphabet et à un langage sur cet alphabet. Nous définissons ensuite la notion de *groupe automatique synchrone*, que nous appelons plus généralement *groupe automatique*.

Rappelons pour commencer qu'un *semigroupe* est un ensemble muni d'une application associative qui n'admet pas nécessairement d'identité. Un *homomorphisme* (ou *morphisme* φ de semigroupes est une application qui respecte les structures, i.e. qui satisfait la condition

$$\varphi(xy) = \varphi(x)\varphi(y).$$

2.1.1 Ensembles de générateurs de groupes

Montrons maintenant comment un langage peut générer un groupe.

Soient G un groupe et Σ une partie finie de G. Nous notons la multiplication d'éléments de Σ par la concaténation de ceux-ci. Nous définissons alors un homomorphisme de semigroupes π défini sur Σ^* et à valeurs dans G. Si w est un mot sur Σ, nous disons que $\pi(w)$ est l'élément de G *représenté* par w ; nous notons cet élément $\bar{w}$. Si l'homomorphisme obtenu est surjectif, i.e. si tout élément du groupe est représenté par un

mot sur Σ, nous disons que Σ est un *ensemble de générateurs de semigroupe* pour G, où que Σ *génère* G comme un semigroupe. Nous disons aussi que Σ *génère* G comme un groupe si $\Sigma \cup \Sigma^{-1}$ génère G comme un semigroupe, l'ensemble Σ^{-1} étant l'ensemble des inverses dans G des éléments de Σ.

Par exemple, le singleton $\{1\}$ génère le groupe additif $\mathbb{Z}$ comme un groupe alors que l'ensemble $\{3, -4\}$ le génère comme un semigroupe.

Nous pouvons étendre cette définition au cas plus général où Σ est un alphabet fini quelconque. Formellement, si G est un groupe et Σ un alphabet fini et si nous disposons d'une application p définie sur Σ et à valeurs dans G, nous étendons cette application en un homomorphisme de semigroupes π défini sur Σ^*. Par conséquent, si $w = a_1 \cdots a_n$ est un mot sur Σ, alors $\pi(w)$ est l'élément

$$p(a_1) \dots p(a_n)$$

de G.

Si cet homomorphisme est surjectif, i.e. si $\pi(\Sigma)$ génère G comme un semigroupe, nous disons que Σ est un *ensemble de générateurs de semigroupe* pour G. Si $\pi(\Sigma)$ génère G comme un groupe, nous disons alors que Σ est un *ensemble de générateurs de groupe* pour G.

Il est souvent requis que l'alphabet Σ considéré soit stable pour le passage à l'inverse. Dans la suite de ce travail, nous supposerons donc, sauf mention explicite du contraire, qu'il existe une involution[1] i sur Σ telle que

$$\forall a \in \Sigma,\ \pi(i(a)) = (\pi(a))^{-1},$$

i.e. nous supposons que pour toute lettre a de Σ, il existe une lettre $i(a)$ qui est envoyée sur l'inverse de $\pi(a)$.

Pour simplifier les notations, nous noterons par a^{-1} l'élément $i(a)$ de Σ.

Étant donné un langage L sur Σ, nous notons également par π la restriction de l'homomorphisme π à L. Le cas le plus intéressant est lorsque cette restriction est encore surjective. Nous verrons dans la suite que lorsque le langage en question est régulier, nous pouvons définir de nouvelles notions et obtenir des résultats intéressants.

2.1.2 Groupes automatiques synchrones

Nous introduisons maintenant les notions de *structure rationnelle* et de *structure automatique* sur un groupe. Celles-ci nous mènent à la définition même de *groupe automatique*.

Si l'alphabet Σ est un ensemble de générateurs de semigroupe pour un groupe G et si le langage L est régulier sur Σ, nous disons que le couple (Σ, L) forme une *structure*

[1] Rappelons qu'une involution sur Σ est une application de Σ dans lui-même qui est telle que $i(i((a)) = a$ pour toute lettre a de Σ.

rationnelle sur G si la restriction de l'homomorphisme de semigroupes π à L est encore surjective.

Nous définissons la relation d'équivalence $\sim_\pi$ sur Σ^* par

$$u \sim_\pi v \Leftrightarrow \pi(u) = \pi(v), \forall\ u, v \in \Sigma^*.$$

Deux mots sont donc en relation par $\sim_\pi$ s'ils ont la même image par π dans G.

Nous pouvons alors voir le langage L comme un ensemble de représentants des classes d'équivalence pour π^{-1}, sachant qu'il n'est pas *a priori* requis qu'il n'existe qu'un unique représentant par classe.

Étant donnée une structure rationnelle (Σ, L) sur un groupe G, nous définissons les relations

$$R_a = \{(u, v) \in \Sigma^* \times \Sigma^* | \overline{ua} = \bar{v},\ u, v \in L\}, a \in \Sigma,$$

et

$$R_\varepsilon = \{(u, v) \in \Sigma^* \times \Sigma^* | \bar{u} = \bar{v},\ u, v \in L\}.$$

Nous appelons L le *langage des représentants* et les R_a, $a \in \Sigma$, les *relations multiplicateur*. La relation R_ε est une relation multiplicateur particulière appelée la *relation égalité*.

Plus généralement nous définissons les *relations multiplicateurs* pour tout mot $w = a_1 \cdots a_l$ sur Σ par

$$R_w = \{(u, v) \in \Sigma^* \times \Sigma^* \mid \overline{uw} = \bar{v},\ u, v \in L\}.$$

Remarquons simplement que $R_w = R_{a_1} \circ \cdots \circ R_{a_l}$.

En effet, si le couple (u, v) appartient à R_w, alors nous avons trivialement

$$\begin{cases} (u, ua_1) \in R_{a_1} \\ (ua_1, v) \in R_{a_2 \cdots a_l} \end{cases}$$

et le couple (u, v) appartient donc à $R_{a_1} \circ R_{a_2 \cdots a_l}$. En répétant ce procédé l fois, nous obtenons que (u, v) appartient à $R_{a_1} \circ \cdots \circ R_{a_l}$.

Réciproquement, si le couple (u, v) appartient à $R_{a_1} \circ \cdots \circ R_{a_l}$, alors il existe un mot t_1 sur Σ tel que

$$\begin{cases} (u, t_1) \in R_{a_1} \\ (t_1, v) \in R_{a_2} \circ \cdots \circ R_{a_l} \end{cases}$$

Ensuite, puisque le couple (t_1, v) appartient à $R_{a_2} \circ \cdots \circ R_{a_l}$, il existe un mot t_2 sur Σ tel que

$$\begin{cases} (t_1, t_2) \in R_{a_2} \\ (t_2, v) \in R_{a_3} \circ \cdots \circ R_{a_l} \end{cases}$$

En continuant de la sorte, nous trouvons des mots $t_1, \ldots, t_l$ sur Σ tels que

$$\begin{cases} \pi(ua_1) = \pi(t_1) \\ \pi(t_i a_{i+1}) = \pi(t_{i+1}),\ 1 \leq i \leq l-1 \\ \pi(t_l) = \pi(v) \end{cases}$$

Nous avons donc

$$\pi(v) = \pi(t_l) = \pi(t_{l-1})\pi(a_l) = \pi(t_{l-2})\pi(a_{l-1}a_l) = \cdots = \pi(u)\pi(w)$$

et le couple (u, v) appartient donc à R_w.

Les relations R_ε et R_σ, $\sigma \in \Sigma \cup \{\varepsilon\}$ sont donc des langages à deux variables sur (Σ, Σ) et nous nous attachons donc à savoir s'il sont réguliers où non.

Une *structure automatique* sur un groupe G est une structure rationnelle sur G pour laquelle les relations R_σ sont synchrones pour tout σ appartenant à $\Sigma \cup \{\varepsilon\}$.

Une structure automatique sur un groupe peut également se définir par un ensemble d'automates, ceux qui acceptent les relations multiplicateurs et celui qui accepte le langage des représentant. En général, nous noterons par $\mathcal{A}$ ce dernier et par $\mathcal{M}_\sigma$ ceux qui acceptent les relations multiplicateurs pour $\sigma \in \Sigma \cup \{\varepsilon\}$.

Nous appelons $\mathcal{A}$ *l'automate accepteur*, $\mathcal{M}_\varepsilon$ *l'automate égalité* et les $\mathcal{M}_a$, $a \in \Sigma$, les *automates multiplicateurs* ; nous supposons ces automate normalisés. Suivant les cas, soit nous les supposerons déterministes, soit nous supposerons qu'ils ne possèdent qu'un unique état final f. Ces deux suppositions sont bien entendu légitimes, mais rien n'assure que nous pouvons les faire simultanément.

Bien que ces automates ne nous seront pas d'une grande utilité dans ce chapitre, ils nous serviront énormément pour les groupes automatiques asynchrones, où nous utilisons la définition d'une structure automatique par ses automates et non pas par ses relations.

Un *groupe automatique synchrone* est simplement un groupe qui admet une structure automatique. Nous définissons au chapitre suivant la notion de *groupe automatique non synchrone*. Pour l'instant, quand nous parlons de *groupe automatique*, nous comprenons *groupe automatique synchrone*.

2.2 Exemple : Les groupes libres

Il existe beaucoup de groupes qui sont automatiques. Nous pouvons citer comme exemple le groupe des tresses ou encore les groupes libres. Intéressons-nous de plus près à ces derniers.

2.2.1 Partie libre d'un groupe

Soient G un groupe et X une partie de G. Si nous pouvons trouver $x_1, \ldots, x_n$ dans X et $\varepsilon_1, \ldots, \varepsilon_n$ dans $\{-1, 1\}$ tels que le produit $x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$ soit l'identité de G, notée 1, nous disons que l'expression

$$x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n} = 1 \tag{2.1}$$

est une *relation*[2] entre les éléments de X. Si dans cette relation il existe un entier $i < n$ tel que $x_i = x_{i+1}$ et $\varepsilon_i = -\varepsilon_{i+1}$, nous obtenons une relation réduite équivalente en supprimant $x_i^{\varepsilon_i} x_{i+1}^{\varepsilon_{i+1}}$ de l'expression. En continuant ainsi tant qu'une telle simplification est possible (ce qui ne peut arriver qu'un nombre fini de fois), nous obtenons à la fin une relation *irréductible*, toujours équivalente à (2.1). Il est possible que cette expression finale soit simplement $1 = 1$, auquel cas la relation de départ est dite *triviale*. Par exemple, la relation $a^{-1}baa^{-1}b^{-1}a = 1$ est triviale.

Si l'expression (2.1) est non triviale, nous disons que les éléments $x_1, \ldots, x_n$ de X sont *liés* par cette relation. Remarquons alors que pour tout morphisme φ entre G et un groupe quelconque G', les images des éléments de X sont aussi liés par une relation non triviale : celle obtenue formellement à partir de (2.1) en remplaçant x_i par $\varphi(x_i)$ pour tout i.

S'il n'existe aucune relation non triviale liant les éléments de X, nous disons que X est une partie *libre* du groupe G.

Si x et y sont deux éléments distincts de X qui commutent, nous pouvons obtenir la relation non triviale suivante

$$xyx^{-1}y^{-1} = 1. \tag{2.2}$$

Ainsi, deux éléments distincts d'une partie libre d'un groupe ne peuvent jamais commuter. En particulier, une partie libre ne peut jamais contenir à la fois un éléments x et son inverse x^{-1} : s'ils sont distincts, ils commutent, ce qui contredirait la remarque précédente, et s'ils sont égaux, nous avons entre éléments de X la relation non triviale $x^2 = 1$.

[2] À ne pas confondre avec les relation multiplicateurs et égalité qui sont des ensembles.
Nous retrouverons cette notion dans la suite lorsque nous définirons un groupe par *générateurs et relations*.

2.2.2 Groupes libres

Un groupe G est dit *libre* sur X si X est une partie à la fois libre et génératrice de G ; on dit alors que X est une *base* de G.

Proposition 2.2.1. *Soit X une partie de G. Le groupe G est libre sur X si et seulement si chaque $g \in G$, $g \neq 1$, s'écrit de manière unique sous la forme*

$$g = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n},$$

où $x_1, \ldots, x_n$ sont dans X, $\varepsilon_1, \ldots, \varepsilon_n$ sont dans $\{-1, 1\}$, et il n'existe pas d'indice $i \in \{1, \ldots, n-1\}$ tel que $x_i = x_{i+1}$ et $\varepsilon_i = -\varepsilon_{i+1}$.

Démonstration. Supposons que le groupe G est libre sur X et procédons par l'absurde. Soit g un élément de G et soient $x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$ et $y_1^{\delta_1} \cdots y_m^{\delta_m}$ deux décompositions différentes de g, où les x_i et les y_j sont des éléments de X pour tout i, j. Nous obtenons alors la relation non triviale

$$x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n} y_m^{-\delta_m} \cdots y_1^{-\delta_1} = 1$$

et la partie X n'est donc pas libre, d'où une contradiction.

Réciproquement, comme tout élément peut se décomposer en des éléments de X, cet ensemble est bien un partie génératrice de G. Si X n'est pas libre, il existe une relation non triviale $x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n} = 1$ entre les éléments $x_1, \ldots, x_n$ de X. Si maintenant l'élément g de G se décompose en le produit

$$y_1^{\delta_1} \cdots y_m^{\delta_m},$$

où les y_i sont des éléments de X pour tout $i \leq m$, il se décompose également en le produit

$$y_1^{\delta_1} \cdots y_m^{\delta_m} x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$$

et la décomposition n'est donc pas unique.

□

2.2.3 Construction d'un groupe libre sur un ensemble X

Étant donné un ensemble X quelconque, il est toujours possible de construire un groupe $F(X)$ libre sur X. Nous considérons un autre ensemble X' disjoint de X et mis en bijection avec X par une application j ; nous notons alors, pour tout x appartenant à X, $x^{-1} = j(x)$. Nous considérons ensuite les mots sur l'ensemble $Y = X \cup X'$, muni de la concaténation. Ceux-ci sont de la forme $w = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$, où $x_1, \ldots, x_n$ sont des éléments de X et $\varepsilon_1, \ldots, \varepsilon_n$ appartiennent à $\{-1, 1\}$. Les mots de longueur 1 sont les éléments de Y et l'unique mot de longueur zéro est noté 1. Un mot est dit *réduit* s'il n'existe pas d'indice i dans $\{1, \ldots, n-1\}$ tel que $x_i = x_{i+1}$ et $\varepsilon_i = -\varepsilon_{i+1}$. Les mots de longueur 1 et celui de longueur nulle sont évidemment réduits.

Le groupe $F(X)$ est précisément l'ensemble des mots réduits muni d'une loi de composition précisée dans la suite. Nous pouvons voir l'ensemble $F(X)$ comme l'ensemble des combinaisons formelles d'éléments de X que l'on ne peut pas simplifier.

Définissons maintenant pas récurrence sur $\min\{m, n\}$ le produit $u.v$ des deux mots réduits u et v de longueurs respectives m et n. Si ce minimum vaut zéro, alors l'un des deux mots u et v vaut 1 et dans ce cas, nous posons $u.1 = u$ et $1.v = v$.

Sinon, nous avons les deux mots réduits $u = u_1^{\varepsilon_1} \cdots u_m^{\varepsilon_m}$ et $v = v_1^{\varepsilon'_1} \cdots v_n^{\varepsilon'_n}$. Supposons que nous sachions définir le produit de deux mots lorsque l'un des deux est de longueur strictement inférieure à $\min\{m, n\}$. Si la concaténation uv est encore un mot réduit, alors nous posons $u.v = uv$. Si ce n'est pas le cas, c'est que $u_m = v_1$ et $\varepsilon'_1 = -\varepsilon_m$. Nous posons alors $u.v = u_1^{\varepsilon_1} \cdots u_{m-1}^{\varepsilon_{m-1}} . v_2^{\varepsilon'_2} \cdots v_n^{\varepsilon'_n}$.

Si G est un groupe libre sur X, le caractère générateur de X pour G nous donne $G = F(X)$.

2.2.4 Monoïdes libres

Un monoïde M est dit *libre* sur X si X est une partie génératrice de M qui est isomorphe à un alphabet Σ. Remarquons que cette définition rejoint celle d'un groupe libre sur une partie X en ce sens qu'un élément m de M se décompose toujours de manière unique en une suite d'éléments de l'alphabet.

Par exemple, le monoïde $M = \{ab, ba, a\}^*$ n'est pas libre sur $\{ab, ba, a\}$. En effet, considérons l'élément aba de M. Nous pouvons le décomposer en deux suites distinctes (ab, a) et (a, ba). De plus, si $\Sigma = \{\alpha, \beta, \gamma\}$ est un alphabet isomorphe à $\{ab, ba, a\}$, où ab, ba et a sont respectivements envoyés sur α, β et γ, alors nous obtenons la relation $\gamma\beta = \alpha\gamma$, ce qui est impossible dans un alphabet.

Il s'ensuit que si Σ est un alphabet fini, alors le monoïde Σ^* (resp. $(\Sigma \times \Sigma)^*$) est libre sur Σ (resp. $\Sigma \times \Sigma$). Par contre, le monoïde $\Sigma^* \times \Sigma^*$ n'est pas libre. En effet, si Σ est l'alphabet $\{a, b\}$, alors le couple (a, ab) de mots peut par exemple s'écrire $(a, a)(\varepsilon, b)$ ou $(\varepsilon, a)(a, b)$.

Montrons maintenant qu'un groupe libre est automatique. Pour ne pas compliquer les notations, considérons le groupe libre $F(\{a, b\})$ construit sur l'ensemble $\{a, b\}$. Nous devons trouver une structure automatique sur $F(\{a, b\})$.

L'homomorphisme envisagé est l'identité et nous prenons comme alphabet $\Sigma = \{a, b, \alpha, \beta\}$, où a et b sont les générateurs du groupe et où α et β sont leurs inverses. Nous choisissons alors comme langage des représentants L l'ensemble des *mots réduits* sur Σ, i.e., l'ensemble des mots ne possédant pas de facteurs de la forme $a\alpha$, $b\beta$, αa ou βb. Ce langage est accepté par l'automate représenté à la figure 2.1 et est donc régulier. De plus, il est clair que nous avons $F(a, b) = L$ et le couple (Σ, L) est donc une structure rationnelle pour ce groupe.

Il reste à montrer que les relations R_ε, R_a, R_b, R_α et R_β sont synchrones.

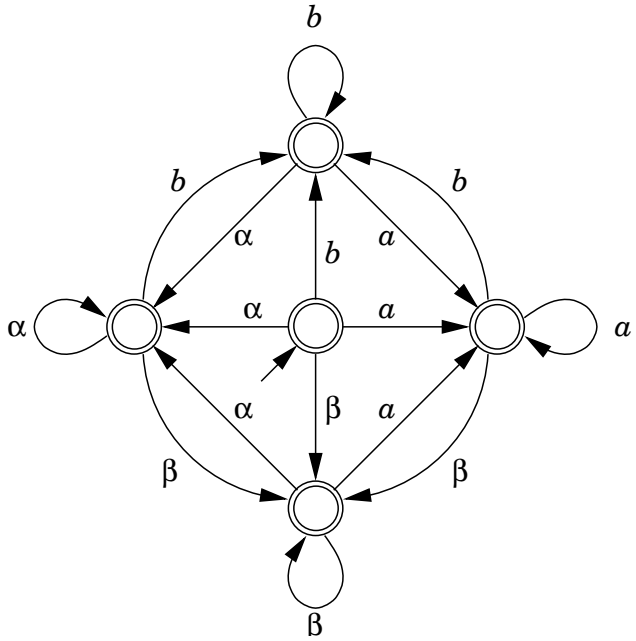

FIG. 2.1 – AFD acceptant le langage des mots réduits sur l'alphabet $\{a, b, \alpha, \beta\}$.

Vu sa définition, la relation R_ε est clairement synchrone. Montrons que la relation

$$R_a = \{(u, ua) \in \Sigma^* \times \Sigma^* \mid u \in L\} \cup \{(u\alpha, u) \in \Sigma^* \times \Sigma^* \mid u \in L\}$$

est synchrone. Pour cela, transformons l'automate représenté à la figure 2.1 afin qu'il accepte la relation R_a.

Nous commençons par remplacer chaque label a, b, α et β respectivement par

$$\begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix}, \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \text{ et } \begin{pmatrix} \beta \\ \beta \end{pmatrix}.$$

Le langage accepté par cet automate est alors

$$\{(u, u) \in \Sigma^* \times \Sigma^* \mid u \in L\}.$$

Nous introduisons ensuite un nouvel état f qui devient l'unique état final de l'automate et nous définissons de nouvelles transitions comme suit.

Pour tout état q de l'automate, nous définissons les transitions $(q, (\$, a), f)$ et $(q, (\alpha, \$), f)$. Au final, nous obtenons l'automate représenté à la figure 2.2 et celui-ci accepte clairement le langage à deux variables $R_a^\$$.

Nous pouvons construire des automates analogues qui acceptent les langages à deux variables $R_b^\$$, $R_\alpha^\$$ et $R_\beta^\$$, d'où la conclusion.

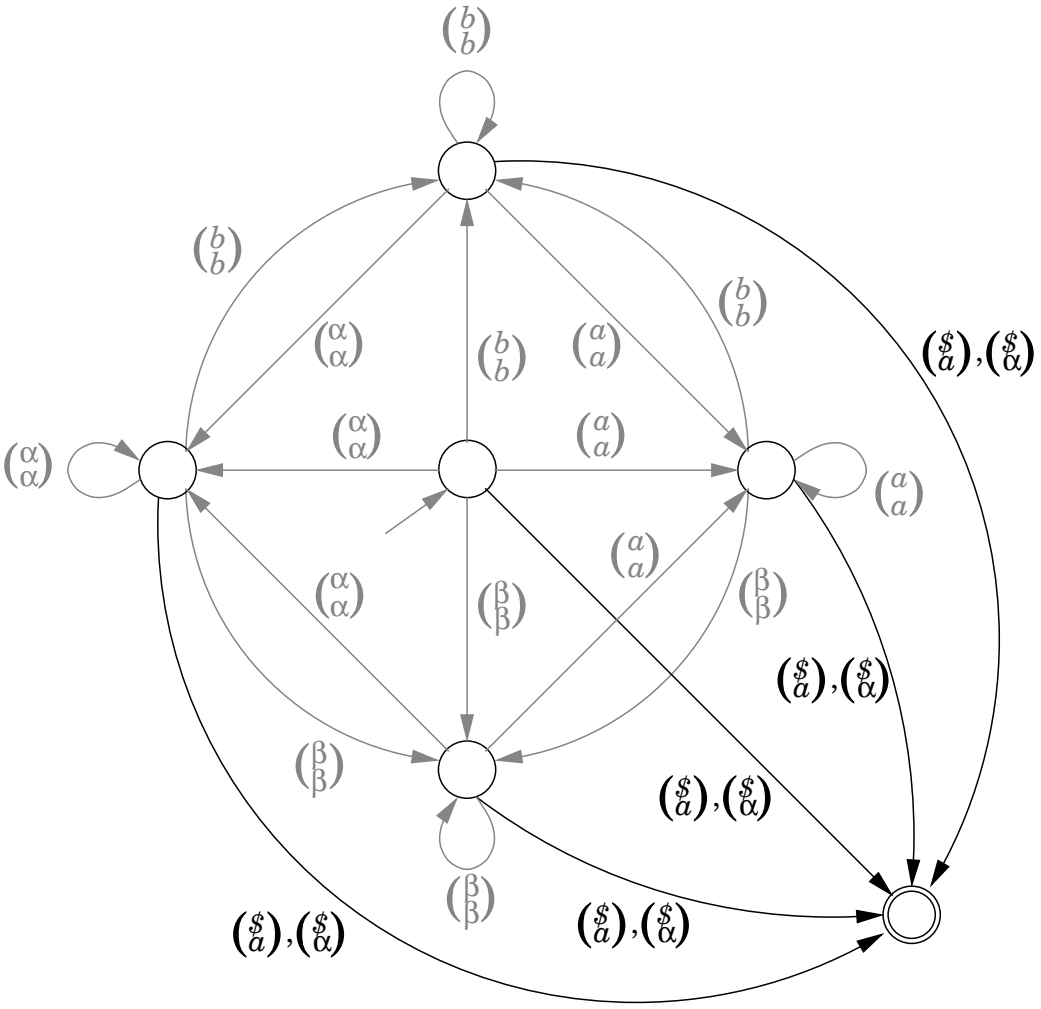

FIG. 2.2 – Automate acceptant la relation multiplicateur R_a.

2.3 Propriétés géométriques

Dans cette section, nous introduisons la notion de *graphe de Cayley* d'un groupe qui est la généralisation des tables de Cayley des groupes finis aux groupes infinis discrets[3]. Nous introduisons ensuite la *propriété du compagnon de voyage* qui nous permet de définir géométriquement les groupes automatiques.

2.3.1 Graphes de Cayley d'un groupe

Soit G un groupe généré par l'alphabet Σ. Son *graphe de Cayley* $\Gamma(G)$ est le graphe connexe orienté dont les sommets sont les éléments[4] de G et dont les labels des arcs sont les éléments de $\Sigma \cup \Sigma^{-1}$, Σ^{-1} étant un ensemble de même cardinal que Σ dont les éléments sont envoyés sur les inverses des images des éléments de Σ. Remarquons simplement que, vu sa définition, le graphe de Cayley d'un groupe dépend de l'ensemble Σ de générateurs du groupe.

Illustrons cette définition sur quelques exemples. La figure 2.3 représente le graphe de Cayley du groupe $S(3)$ des permutations à trois éléments. Les générateurs de ce groupe sont les permutations $x = (1\ 2)$ et $y = (2\ 3)$.

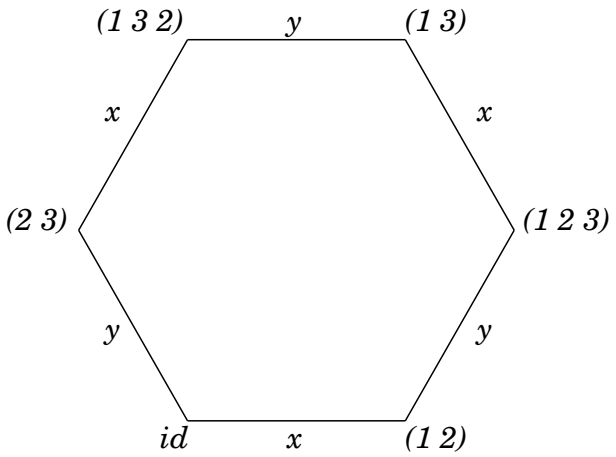

FIG. 2.3 – Graphe de Cayley du groupe $S(3)$.

De même, la figure 2.4 représente le graphe de Cayley du groupe $S(4)$; les générateurs sont les permutations $x = (1\ 2)$, $y = (2\ 3)$ et $z = (3\ 4)$.

[3] Ils sont néanmoins également valables pour les groupes finis.

[4] Il existe une définition plus générale des graphes de Cayley. Dans cette dernière, les sommets du graphe ne sont plus les éléments du groupe G, mais bien les classes latérales Hx, où H est un sous-groupe de G. Cependant, la définition introduite ici nous suffit et il n'est dès lors pas nécessaire de s'intéresser au cas général. Remarquons simplement que la définition que nous utilisons est équivalente à celle plus générale dans le cas où H est le sous-groupe trivial de G.

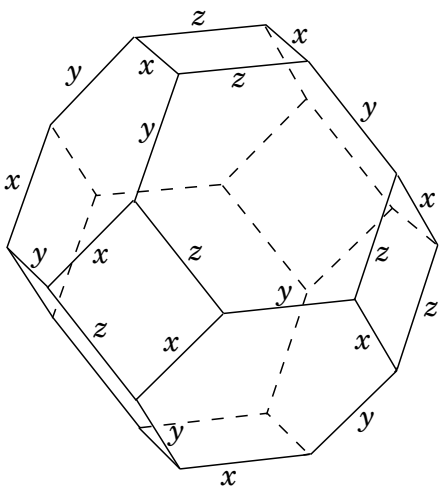

FIG. 2.4 – Graphe de Cayley du groupe $S(3)$.

Enfin, la figure 2.5 représente le graphe de Cayley du groupe libre sur $\{a, b\}$ dont les inverses de a et de b sont respectivement α et β. Comme le groupe libre est infini, il est impossible de représenter son graphe dans son entièreté. Il est facile de se convaincre que le graphe se prolonge à l'infini.

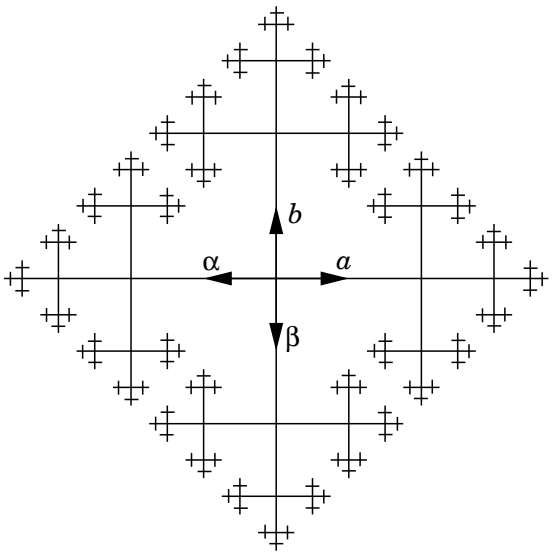

FIG. 2.5 – Graphe de Cayley du groupe libre $F(a, b)$.

À tout mot w écrit sur l'ensemble Σ des générateurs d'un groupe G, nous associons un *chemin*[5] $\hat{w} : [0, \infty[\to \Gamma(G)$ dans le graphe de Cayley de G comme suit. Si t est un entier, alors $\hat{w}(t) = \overline{w(t)}$ est l'image dans G du préfixe de w de longueur t ($w(t) = w$

[5]À ne pas confondre avec le chemin que le mot décrit dans un automate.

pour $t \geq |w|$). Nous étendons $\hat{w}$ aux valeurs non entières de t en se déplaçant sur les arcs du graphe avec une vitesse unitaire, i.e. le chemin traverse un arc en une unité de temps. La fiure 2.6 illustre le chemin associé à un mot $abb\alpha$; le chemin avance avec une vitesse unitaire pour $t \in [0, 4]$ puis s'arrête.

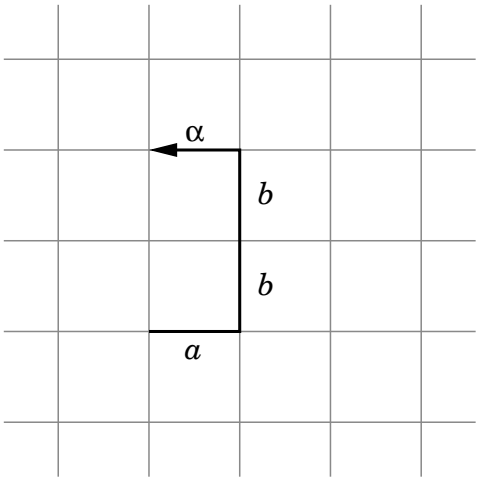

FIG. 2.6 – Représentation du chemin associé au mot $abb\alpha$.

Définissons une distance sur G. Étant donnés x et y deux éléments de G, la *distance* $d(x, y)$ est la plus petite longueur, en terme de nombre d'arcs, d'un chemin débutant en x et aboutissant en y dans $\Gamma(G)$. Il est facile de voir que cela définit bien une distance.

Pour illustrer cette définition, revenons au groupe $S(3)$. Dans celui-ci, il est facile de voir que les deux produits de permutations (1 2)(2 3)(1 2) et (2 3)(1 2)(2 3) représentent le même élément (1 3) ; nous avons $d(xyx, yxy) = 0$. Par contre, les produits (1 2)(2 3) et (2 3)(1 2)(2 3) ne coïncident pas. Cependant, il suffit d'appliquer la permutation (1 2) à l'un de ces deux éléments pour retrouver l'autre. Nous avons donc $d(xy, yxy) = 1$.

Pour le groupe libre, considérons deux mots réduits u et v sur $\{a, b, \alpha, \beta\}$. Soit $i \leq \min\{|u|, |v|\}$ tel que $u(i) = v(i)$, où $u(i)$ et $v(i)$ représentent les préfixes respectifs de u et de v de longueur i. Il est alors clair que nous avons $d(\bar{u}, \bar{v}) = |u| + |v| - 2i$.

Les graphes de Cayley sont très utiles pour les groupes admettant une structure rationnelle. En effet, si nous disposons de deux mots u et v appartenant au langage des représentants L, il nous est très facile de voir si ces deux mots sont envoyés sur le même élément, s'ils ne diffèrent que d'un générateur, ou s'ils sont très différents[6] l'un de l'autre. Nous pouvons donc voir facilement si le couple (u, v) appartient à la relation égalité ou à une des relations multiplicateur de la structure.

Nous notons par $|x|$ la longueur d'un élément x du groupe, définie par le nombre minimum de générateurs dans Σ nécessaires pour le représenter.

[6]Par "très différents", nous entendons que plusieurs générateurs sont nécessaire pour retrouver un élément à partir de l'autre.

Remarquons que dans le cas du groupe libre, nous avons toujours $|\bar{w}| = |w|$ pour tout mot réduit w.

Nous disposons des deux propriétés suivantes.

$$d(zx, zy) = d(x, y) = d(1, y^{-1}x) = d(1, x^{-1}y)$$
$$d(x, y) = |y^{-1}x| = |x^{-1}y|$$

En effet, si n est la distance entre x et y, alors le chemin le plus court entre ces deux éléments passe par n arcs dans le graphe de Cayley du groupe. Il existe donc $a_1, \ldots, a_n$ (resp. $b_1, \ldots, b_n$) dans $\Sigma \cup \Sigma^{-1}$ tels que $y = xa_1 \cdots a_n$ (resp. $x = yb_1 \cdots b_n$) et nous obtenons alors $x^{-1}y = 1a_1 \cdots a_n$ (resp. $y^{-1}x = 1b_1 \cdots b_n$) et il n'existe clairement pas de chemin plus court entre $x^{-1}y$ (resp. $y^{-1}x$) est 1 car dans ce cas, il existerait un chemin plus court entre x et y. Ceci montre que $d(x, y) = d(1, y^{-1}x) = d(1, x^{-1}y)$, ce qui conclut la première égalité car

$$d(zx, zy) = d(1, (zx)^{-1}zy) = d(1, x^{-1}z^{-1}zy) = d(x, y).$$

La deuxième égalité découle directement de la première car, vu sa définition, la distance $d(1, x^{-1}y)$ est exactement le nombre minimum de générateurs dans Σ pour représenter $x^{-1}y$.

Pour revenir à nouveau au groupe $S(3)$, il est facile de remarquer que $d(xy, yxy) = d(id, (xy)^{-1}yxy) = d(id, yxyxy) = 1$. De plus, le produit $(2\ 3)(1\ 2)(2\ 3)(1\ 2)(2\ 3)$ est en fait la permutation $(1\ 2)$, il suffit donc d'un générateur x pour représenter cet élément et nous avons donc bien $d(xy, yxy) = d(id, yxyxy) = |yxyxy| = 1$.

Si le contexte n'est pas clair, nous utilisons la notation $d_G(x, y)$ pour la distance relative au groupe G.

2.3.2 Lien avec les groupes automatiques

Les graphes de Cayley nous permettent de définir les groupes automatiques en termes géométriques grâce à la notion de *propriété du compagnon de voyage*[7] expliquée ci-après. Considérons deux cyclistes se déplaçant à la même vitesse dans le graphe de Cayley d'un groupe et démarant tous deux dans le sommet représentant l'identité du groupe. Ils peuvent éventuellement s'arrêter tous les deux. La propriété exprime le fait qu'à tout moment, la distance qui sépare les deux cyclistes, en termes de nombre minimum d'arcs qui les séparent, est majorée par une constante.

Formellement, pour un groupe G muni d'une structure rationnelle (Σ, L), nous considérons un couple (u, v) de mots sur Σ et pour tout entier $0 < i \leq |u|$ (resp. $0 < i \leq |v|$), nous notons par u_i (resp. par v_i) le préfixe de u (resp. de v) de longueur i et nous posons $u_i = u$ (resp. $v_i = v$) pour tout $i \geq |u|$ (resp. pour tout $i \geq |v|$). Nous disons alors que le couple (u, v) satisfait la *k-propriété du compagnon de voyage* si nous

[7] Dans la littérature anglaise, nous rencontrons le terme *"fellow traveller property"*.

avons $d(\bar{u}_i, \bar{v}_i) \leq k$ pour tout entier $i > 0$, où $d(\bar{u}_i, \bar{v}_i)$ est la distance entre $\bar{u}_i$ et $\bar{v}_i$ dans le graphe $\Gamma(G)$.

Plus généralement, nous disons qu'un sous-ensemble R de $\Sigma^* \times \Sigma^*$ satisfait la k-propriété du compagnon de voyage si tous les éléments (u, v) de R la satisfont.

Si nous reprenons l'exemple du groupe libre à deux générateurs, il est clair que toutes les relations multiplicateurs satisfont la 1-propriété du compagnon de voyage. En effet, puisqu'il n'existe aucune relation non triviale dans ce groupe, un couple (u, v) de mots sur $\{a, b, \alpha, \beta\}$ appartient, par exemple, à la relation[8] R_a si $ua = v$. Pour les mêmes raisons, il est clair que la relation égalité satisfait la 0-propriété du compagnon de voyage.

En ce qui concerne le groupe $S(4)$, si nous considérons l'alphabet $\Sigma = \{x, y, z\}$, alors (Σ, Σ^*) est une structure rationnelle sur $S(4)$ et nous pouvons remarquer que toutes les relations multiplicateurs, ainsi que la relation égalité, satisfont la 6-propriété du compagnon de voyage. En effet, vu son graphe de Cayley, deux éléments du groupe sont à une distance d'au plus 6.

Le théorème suivant est une clef indispensable de bon nombre de démonstrations de la suite de ce travail. Il stipule qu'une structure rationnelle est automatique si et seulement si les relations multiplicateurs satisfont la propriété du compagnon de voyage.

Théorème 2.3.1 (Caractérisation des structures automatiques)**.** *Une structure rationnelle (Σ, L) sur un groupe G est automatique si et seulement s'il existe un entier $k > 0$ tel que pour tout σ appartenant à $\Sigma \cup \{\varepsilon\}$, la relation multiplicateur R_σ satisfait la k-propriété du compagnon de voyage.*

Démonstration. Montrons que la condition est nécessaire. Soit σ appartenant à $\Sigma \cup \{\varepsilon\}$. La structure rationnelle (Σ, L) étant automatique, la relation R_σ est synchrone et nous supposons que l'automate multiplicateur $\mathcal{M}_\sigma$ correspondant est déterministe. Soit K_σ le nombre d'états de cet automate et soit (u, v) un couple de mots appartenant à R_σ. Il existe un chemin dans $\mathcal{M}_\sigma$ débutant en q_0, d'exécution $(u, v)^\$$, et aboutissant en un état final f. Considérons un préfixe synchrone (u_1, v_1) de (u, v). Le chemin débutant en q_0 d'exécution $(u_1, v_1)^\$$ aboutit dans un état q de l'automate et il existe alors un chemin de longueur inférieure à $K_\sigma - 1$ débutant en q, aboutissant en f et d'exécution $(u_2, v_2)^\$$, où u_2 et v_2 sont des mots sur Σ. Nous obtenons alors $(u_1 u_2, v_1 v_2) \in R_\sigma$ et donc $\overline{u_1 u_2 \sigma} = \overline{v_1 v_2}$. Il s'ensuit que

$$d(\bar{u}_1, \bar{v}_1) = d(1, \bar{u}_1^{-1} \bar{v}_1) = d(1, \bar{u}_2 \bar{\sigma} \bar{v}_2^{-1}) \leq |u_2 \sigma v_2^{-1}| \leq 2K_\sigma - 1,$$

où v_2^{-1} est le mot de Σ^* envoyé sur l'inverse[9] de $\bar{v}_2$; la relation R_σ satisfait donc la $(K_\sigma - 1)$-propriété du compagnon de voyage.

[8] Remarquons qu'il s'agit ici d'une égalité dans Σ^* et pas dans G.

[9] La majoration par $|v_2 a u_2^{-1}|$ vient du fait que rien n'assure que le mot $v_2 a u_2^{-1}$ soit réduit. Le nombre d'éléments de Σ nécessaires pour générer $\bar{v}_2 \bar{a} \bar{u}_2^{-1}$ est donc inférieur ou égal à la longueur de ce mot.

Pour la réciproque, nous considérons la boule fermée[10] B de centre 1 et de rayon k dans $\Gamma(G)$. Soit alors un AFD $\mathcal{A} = (Q, q_0, F, \Delta)$ sur Σ acceptant L. Nous ajoutons un nouvel état f à l'ensemble d'états qui devient l'unique état final de l'automate. Nous ajoutons alors les transitions $(q, \$, f)$ pour tout état q appartenant à F. L'automate ainsi défini accepte clairement le langage $L\$$ et il n'existe aucune transition dans Δ de la forme (f, a, q), où a est une lettre de Σ et où q est un état de l'automate. Nous notons cet automate par $\mathcal{A}' = (Q', q_0, \{f\}, \Delta')$.

Pour tout σ dans $\Sigma \cup \{\varepsilon\}$, nous définissons l'automate multiplicateur $\mathcal{M}_\sigma = (Q' \times Q' \times B, (q_0, q_0, 1), (f, f, \bar{\sigma}), \Delta_\sigma)$ sur $\Sigma^\$ \times \Sigma^\$$, où Δ_σ est l'ensemble des transitions obtenues comme suit.

Étant données (q, b, q') et (p, c, p') deux transitions dans Δ' et x un élément de G, le quadruple $((q, p, x), b, c, (q', p', \bar{b}^{-1}x\bar{c}))$ est une transition de l'automate $\mathcal{M}_\sigma$ si x et $\bar{b}^{-1}x\bar{c}$ appartiennent à B. Nous ajoutons également dans Δ les transitions du type $((q, p, x), b, \$, (q', p, \bar{b}^{-1}x))$ et $((q, p, x), \$, c, (q, p', x\bar{c}))$, où les triplets (q, b, q') et (p, c, p') sont des transitions de l'automate $\mathcal{A}$. Remarquons dès à présent que, vu la définition des transitions, l'automate $\mathcal{M}_\sigma$ est déterministe pour tout σ dans $\Sigma \cup \{\varepsilon\}$.

Nous concluons en montrant par récurrence sur $n = max\{|u|, |v|\}$ qu'un couple de mots (u, v) décrit un chemin dans $\mathcal{M}_\sigma$ débutant dans $(q_0, q_0, 1)$ et aboutissant dans (p, q, x) pour des états p et q si et seulement si $\bar{u}^{-1}\bar{v} = x$.

Les cas $n = 0$ et $n = 1$ sont triviaux. Supposons maintenant le résultat acquis pour $max\{|u|, |v|\} \leq n - 1$ et démontrons-le pour $max\{|u|, |v|\} = n$.

Considérons le cas où $|u| = |v| = n$. Nous posons $u = u'u_n$ et $v = v'v_n$, où u', v' appartiennent à Σ^* et u_n, v_n sont des lettres de Σ. Les mots u' et v' sont de longueur $n-1$ et par l'hypothèse de récurrence, il existe des états p et q tels que le couple (u', v') décrit un chemin dans $\mathcal{M}_\sigma$ débutant en $(q_0, q_0, 1)$ et aboutissant en $(q, p, \bar{u}'^{-1}\bar{v}')$, avec $\bar{u}'^{-1}\bar{v}'$ appartenant à B. L'automate $\mathcal{A}$ étant déterministe, les triplets (q, u_n, q') et (p, v_n, p') sont des transitions de celui-ci et le couple (u, v) jouissant de la k-propriété du compagnon de voyage, la distance $d(\bar{u}, \bar{v})$ est inférieure à k, ce qui implique que le sommet $\bar{u}^{-1}\bar{v}$ appartient à B. Il s'ensuit que le quadruple $((q, p, \bar{u}'^{-1}\bar{v}'), u_n, v_n, (q', p', \bar{u}^{-1}\bar{v}))$ est une transition de $\mathcal{M}_\sigma$.

Le cas où l'un des deux mots u et v a une longueur strictement inférieure à n s'obtient de manière analogue, en décomposant le couple (u, v) en $(u, v')(\$^{n-|v'|}, v'')$ si $|u| < n$ (resp. en $(u', v)(u'', \$^{n-|u'|})$ si $|v| < n$), avec v' et v'' (resp. u' et u'') appartenant à Σ^*. Il suffit alors de remarquer que $\bar{\$} = \bar{\$}^{-1} = 1$.

□

Une conséquence intéressante de ce théorème est que tout groupe fini est automatique. En effet, il suffit de considérer comme ensemble de générateurs Σ tous les éléments du groupe et comme langage des représentants L tous les mots sur Σ. Il est alors clair que les relations multiplicateurs, ainsi que la relation égalité, satisfont la

[10] Il s'agit en fait de l'ensemble des sommets x de $\Gamma(G)$ tels que $|x| \leq k$.

propriété du compagnon de voyage, car, le groupe étant fini, deux éléments du groupe ne peuvent différer que par au plus un nombre fini de générateurs.

Corollaire 2.3.2. *Soient (Σ, L) une structure automatique pour un groupe G et w un mot sur Σ. Il existe un entier K tel que la relation multiplicateur R_w satisfait la K-propriété du compagnon de voyage.*

Démonstration. Vu le théorème précédent, il existe un entier k strictement positif tel que pour tout élément σ de $\Sigma \cup \{\varepsilon\}$, la relation R_σ satisfait la k-propriété du compagnon de voyage.

Notons $w = a_1 \cdots a_p$, où les a_i sont des lettres de Σ. Nous avons $R_w = R_{a_1} \circ \cdots \circ R_{a_p}$. Montrons que R_w a la K-propriété du compagnon de voyage pour $K = kp$.

Procédons par récurrence sur p.

Les cas $p = 0$ et $p = 1$ sont triviaux. Supposons le résultat acquis pour les mots de longueur inférieure à $p - 1$ et démontrons-le pour ceux de longueur p. Nous pouvons écrire $w = w' a_p$; nous avons alors $R_w = R_{w'} \circ R_{a_p}$. Par l'hypothèse de récurrence, les relations $R_{w'}$ et R_{a_p} satisfont respectivement la $(k(p-1))$-propriété du compagnon de voyage et la k-propriété du compagnon de voyage. Considérons maintenant un couple (u, v) de mots appartenant à R_w et choisissons un de ses préfixes synchrones (u_i, v_i). Il existe un mot r sur Σ tel que les couples (u, r) et (r, v) de mots appartiennent respectivement à $R_{w'}$ et à R_{a_p}. Nous avons alors $d(u_i, r_i) \leq k(p-1)$ (resp. $d(r_i, v_i) \leq k$) et nous obtenons

$$d(u_i, v_i) \leq d(u_i, r_i) + d(r_i, v_i) \leq kp,$$

d'où la conclusion.

□

2.4 Caractère intrinsèque

Le résultat principal de cette section est que le caractère automatique d'un groupe est propre à ce groupe ; il ne dépend pas de son ensemble de générateurs. Ainsi, si nous savons qu'un groupe est automatique pour un ensemble de générateurs, il l'est aussi pour tout autre ensemble. Nous disposons du résultat suivant.

Théorème 2.4.1. *Soient G un groupe et Σ et Δ deux ensembles de générateurs (de semigroupes) pour G. Le groupe G est automatique sur Σ si et seulement s'il l'est sur Δ.*

Démonstration. Nous démontrons ce résultat en supposant l'existence d'une involution (définie en 2.1.1) pour chacun des deux ensembles de générateurs. Les deux sens de la démonstration étant complétement analogues, nous supposons que le groupe G admet une structure automatique (Σ, L) et nous montrons qu'il est automatique par rapport à Δ. Nous notons par R_a^Σ (resp. R_b^Δ) les relations multiplicateurs relatives à Σ (resp. à Δ).

Considérons le cas où Δ contient un élément e qui est envoyé sur l'élément neutre de G. Nous pouvons alors directement construire une structure automatique (Δ, L') sur G. L'ensemble Δ étant un ensemble de générateurs de G, pour toute lettre a de Σ, il existe un mot u sur Δ qui est envoyé sur le même élément que a. Notons u_a ce mot. L'élément e de Δ étant envoyé sur le neutre du groupe, nous pouvons supposer que les mots u_a ont la même longueur l pour tout a appartenant à Σ. Il suffit pour cela d'ajouter, si nécessaire, des occurences de e à la fin de ces mots.

Soit le morphisme θ défini sur Σ^* et à valeurs dans Δ^* défini par $\theta(a) = u_a$. Nous considérons alors le langage $L' = \theta(L)$. Remarquons que tous les mots de L' ont une longueur égale à un multiple de l. Notons par $d_\Sigma(x, y)$ (resp. $d_\Delta(x, y)$) la distance entre deux éléments x et y de G dans le graphe de Cayley de G quand il est généré par Σ (resp. Δ). Il vient

$$d_\Delta(x, y) \leq l d_\Sigma(x, y).$$

Montrons maintenant que le couple (Δ, L') est une structure automatique pour G. Par le théorème 1.1.1, le langage L' est régulier. Il suffit donc de montrer que les relations R_b^Δ satisfont la propriété du compagnon de voyage.

L'ensemble Δ étant fini, il existe un entier strictement positif λ tel que pour toute lettre b de Δ, il existe un mot u sur Σ de longueur inférieure à λ qui est envoyé sur le même élément que b. Par le théorème 2.3.1, il existe également un entier strictement positif k tel que la relation R_σ satisfait la k-propriété du compagnon de voyage pour tout symbole σ de $\Sigma \cup \{\varepsilon\}$. Pour tout mot w sur Σ, nous trouvons alors, par le corollaire 2.3.2, un entier K_w tel que la relation R_w a la K_w-propriété du compagnon de voyage. Nous considérons l'entier $K = sup\{K_w | w \in \Sigma^*,\ |w| \leq \lambda\}$.

Soient maintenant $(\theta(u), \theta(v))$ appartenant[11] à R_b^Δ et (z, t) un de ses préfixes synchrones. Vu la définition du morphisme θ, il existe deux mots u' et v' sur Σ de même

[11] Un élément de R_b^Δ est bien de cette forme puisque les couples appartiennent en particulier à l'ensemble $L' \times L'$.

longueur et deux mots z_1 et t_1 sur Δ de longueurs majorées par l tels que $\theta(u')z_1 = z$ et $\theta(v')t_1 = t$. Les deux alphabets Σ et Δ étant des ensembles de générateurs de G, il existe un mot w sur Σ qui a la même image que b dans G. En particulier, le mot w est de longueur inférieure à λ et la relation R_w satisfait donc la K-propriété du compagnon de voyage. De plus, il est clair que le couple (u, v) appartient à R_w. Il s'ensuit que

$$d_\Delta(\bar{z}, \bar{t}) = d_\Delta(\overline{\theta(u')}\bar{z}_1, \overline{\theta(v')}\bar{t}_1) \leq l d_\Sigma(\bar{u}', \bar{v}') + |z_1| + |t_1| \leq (K+2)l$$

et R_b^Δ satisfait donc la $(K+2)l$-propriété du compagnon de voyage.

Pour le cas où Δ ne contient pas d'élément e envoyé sur l'identité du groupe, il suffit de montrer que si G est automatique sur $\Sigma' = \Delta \cup \{e\}$, alors il est automatique sur Δ. En effet, étant donnée une structure automatique (Σ, L) sur G, nous avons montré au cas précédent qu'il était possible de trouver un langage L' sur Σ' tel que (Σ', L') soit une structure automatique sur G.

Considérons donc une structure automatique (Σ', L) sur G. Nous ne pouvons pas simplement supprimer toutes les occurences de e dans les mots de L car cela pourrait violer la propriété du compagnon de voyage. Soit K un entier tel que les relations $R_a^{\Sigma'},\ a \in \Sigma'$, ont la K-propriété du compagnon de voyage. L'ensemble Δ étant un ensemble de générateurs de G qui ne contient pas d'élément envoyé sur l'identité, il existe un mot w sur Δ de longueur m qui est envoyé sur l'identité du groupe. En particulier, ce mot appartient à Σ'^*. Pour tout mot w sur Σ', nous numérotons les occurences de e dans ce mot à partir de 1. Nous considérons alors l'application θ défini sur Σ'^*, à valeurs dans Δ^* et qui à un mot associe ce même mot dans lequel toutes les occurences de e ont été effacées et chaque occurence de e numérotée d'un multiple non nul de m a été remplacée par le mot w.

Remarquons que pour tout préfixe v d'un mot u sur Σ', nous avons $0 \leq |v| - |\theta(v)| < m$

Considérons alors le langage $L' = \theta(L)$ et montrons qu'il est régulier. Soit $\mathcal{A} = (Q, q_0, F, \Delta)$ un AFD acceptant le langage L. Nous construisons un AFND $\mathcal{A}'$ qui accepte L' comme suit.

L'ensemble d'états est

$$\{(q, i) \in Q \times \mathbb{N} \mid 0 \leq i < m\},$$

où la composante i désigne la différence de longueur entre le mot d'entrée $\theta(u)$ et le mot de départ u. L'état initial est le couple $(q_0, 0)$ et l'ensemble des états finals est

$$\{(f, i) \in Q \times \mathbb{N} \mid f \in F,\ 0 \leq i < m\}.$$

Chaque transition (q, a, p) de Δ donne lieu à plusieurs transitions dans $\mathcal{A}'$ comme suit : si $a \neq e$, les transitions en question sont les triplets $((q, i), a, (p, i))$ pour $0 \leq i < m$; si $a = e$, alors les transitions sont le triplet $((q, o), w, (p, m-1))$ et les triplets $((q, i), \varepsilon, (p, i-1))$ pour $1 \leq i < m$.

Montrons maintenant que (Δ, L') est une structure automatique sur G. Soient a une lettre de Δ, (u', v') un couple de R_a^{Δ} et (u'_1, v'_1) un préfixe synchrone de (u', v'). Nous pouvons trouver un couple (u, v) de $R_a^{\Sigma'}$ vérifiant $\theta(u) = u'$ et $\theta(v) = v'$. Nous obtenons alors

$$d(\bar{u}'_1, \bar{v}'_1) \leq d(\bar{u}'_1, \bar{u}_1) + d(\bar{u}_1, \bar{v}_1) + d(\bar{v}_1, \bar{v}'_1) \leq d(\bar{u}_1, \bar{v}_1) + 2m,$$

ce qui montre que R_a^{Δ} satisfait la propriété du compagnon de voyage.

□

2.5 Simplification des structures automatiques

La définition des structures automatiques est purement existentielle. Lorsque nous travaillons avec un groupe dont nous savons qu'il est automatique, nous devons trouver une structure automatique efficace pour celui-ci.

La notion de structure comme nous l'avons définie n'impose pas que le langage L soit minimal dans le sens qu'il contient un nombre minimum de mots. Le résultat principal de cette section est que nous pouvons toujours supposer qu'il ne contient qu'un unique représentant pour chaque élément du groupe. Le second résultat est que le langage des représentants peut être pris *préfixiel*. Cependant, la réalisation simultanée de ces deux condition est instable.

Une méthode pour restreindre le langage des représentant est de considérer un ordre sur l'alphabet[12] et de choisir le langage des représentants en fonction de cet ordre.

Commençons par démontrer deux résultats concernant les ordres sur un alphabet.

Si L est un langage régulier, nous définissons le langage $B(L)$ comme étant l'union des mots de L qui sont les plus grands parmi les mots de leur longueur pour l'ordre de l'alphabet, i.e.

$$B(L) = \bigcup_{n\geq 1} \{u \in L \cap \Sigma^n \mid \forall v \in L \cap \Sigma^n,\ u \geq v\},$$

où la relation $\geq$ désigne l'ordre de l'alphabet.

De la même façon, nous définissons le langage $S(L)$ comme étant l'union des mots de L qui sont les plus petits parmi les mots de leur longueur, i.e.

$$S(L) = \bigcup_{n\geq 1} \{u \in L \cap \Sigma^n \mid \forall v \in L \cap \Sigma^n,\ u \leq v\}.$$

Proposition 2.5.1. *Si L est un langage régulier sur un alphabet Σ, alors $B(L)$ est également un langage régulier sur ce même alphabet.*

Démonstration. Il suffit de montrer que le langage $L\backslash B(L)$ est régulier, car dans ce cas nous obtenons

$$B(L) = (\Sigma^*\backslash(L\backslash B(L))) \cap L,$$

ce qui suffit vu les propriétés de stabilité des langages réguliers pour l'intersection et le passage au complémentaire.

Considérons un AFD $\mathcal{A} = (Q, q_0, F, \delta)$ sur Σ acceptant le langage L. Nous construisons un AFND $\mathcal{A}_0$ sur Σ qui accepte exactement $(L\backslash B(L)$.

Nous utilisons pour cela deux bandes pour simuler le comportement de l'automate $\mathcal{A}$ sur un mot w. La première bande imite exactement $\mathcal{A}$, alors que la seconde, non déterministe, simule le comportement de $\mathcal{A}$ sur tous les mots possibles de longueur $|w|$, essayant de trouver un mot de L plus grand que w pour l'ordre de l'alphabet. Si un tel

[12] Il s'agit en général de l'ordre du dictionnaire.

mot existe et si w est accepté par l'automate $\mathcal{A}$, alors w est un mot de L qui n'est pas le plus grand de sa longueur ; l'automate $\mathcal{A}_0$ accepte donc w.

Plus formellement, nous considérons l'automate $\mathcal{A}_0 = (Q_0, p_0, F_0, \Delta_0)$, avec $Q_0 = Q \times Q \times \{g, e, l\}$; la troisième composante d'un état s'appelle le *curseur*.

Le curseur g (resp. e, l) signifie que, dans l'état courant, le mot lu sur la deuxième bande est *plus grand* (resp. *égal*, *plus petit*) que le préfixe de w lu sur la première bande.

Nous trouvons un mot sur Σ de longueur $|w|$ de la façon suivante : chaque fois que nous lisons une lettre a_i de w sur la première bande de $\mathcal{A}_0$, nous sélectionnons une lettre σ_i de Σ sur la deuxième bande. Si le curseur, est en e et si $\sigma_i > a_i$ (resp. $\sigma_i < a_i$), alors le curseur se place en g (resp. en l). Sinon, le curseur ne change pas.

Lorsque la lecture du mot w est terminée, nous avons construit un mot v de longueur $|w|$ sur Σ et le curseur indique sa position par rapport à w pour l'ordre de l'alphabet.

Nous prenons comme état initial l'état $p_0 = (q_0, q_0, e)$ et comme ensemble d'états finals l'ensemble

$$F_0 = \{(q, p, g) \mid q, p \in F\}.$$

Nous définissons l'ensemble des transitions $\Delta_0 \subset Q_0 \times \Sigma \times Q_0$ de $\mathcal{A}_0$ comme suit. Si a est une lettre de Σ, alors

- $((q, p, g), a, \delta(q, a), \delta(q, \sigma), g) \in \Delta_0 \quad \forall \sigma \in \Sigma$;
- $((q, p, e), a, \delta(q, a), \delta(q, \sigma), x) \in \Delta_0 \quad \forall \sigma \in \Sigma$,
  où $x = g$ (resp. e, l) si $a < \sigma$ (resp. $=$, $>$) ;
- $((q, p, l), a, \delta(q, a), \delta(q, \sigma), l) \in \Delta_0 \quad \forall \sigma \in \Sigma$.

Notons que Δ_0 est bien fini, car Σ et Q_0 sont finis. La figure 2.7 illustre l'ensemble des transitions de $\mathcal{A}_0$ pour le cas $\Sigma = \{a, \sigma\}$, avec $a < \sigma$.

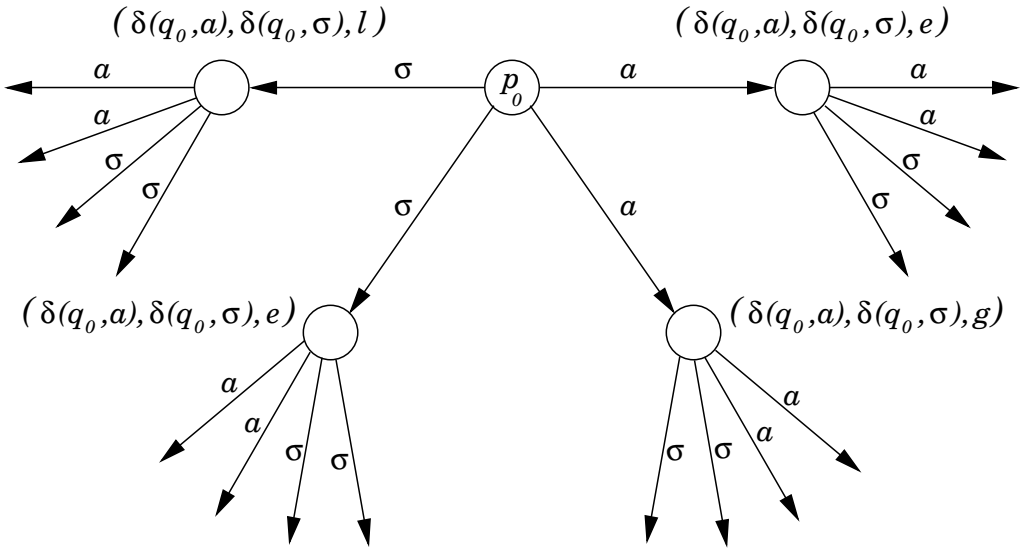

FIG. 2.7 – Illustration des transition de l'automate $\mathcal{A}_0$.

Si $p_0.u$ désigne l'ensemble des états accessibles à partir de p_0 en lisant u, nous obtenons

$$\begin{aligned}
w \in L(\mathcal{A}_0) &\Leftrightarrow w \in \{v \in \Sigma^* \mid p_0.v \cap F \neq \phi\} \\
&\Leftrightarrow p_0.w \cap F_0 \neq \phi \\
&\Leftrightarrow (q_0, q_0, e).w \cap F_0 \neq \phi \\
&\Leftrightarrow \{(q_0.w, q_0.v, x) \mid v \in \Sigma^*,\ |v| = |w|\} \cap \{(q, p, g) \mid q, p \in F\} \neq \phi \\
&\Leftrightarrow \exists v \in \Sigma^* :\ q_0.w \in F, q_0.v \in F,\ |v| = |w| \ et\ x = g \\
&\Leftrightarrow \exists v \in \Sigma^* : w \in L,\ v \in L,\ |v| = |w| \ et\ u > w \\
&\Leftrightarrow w \in L \backslash B(L)
\end{aligned}$$

Ainsi, le langage $L\backslash B(L)$ est accepté par l'automate $\mathcal{A}_0$ et est donc régulier. □

Proposition 2.5.2. *Si L est un langage régulier sur un alphabet Σ, alors $S(L)$ est également un langage régulier sur ce même alphabet.*

Démonstration. La démonstration de ce résultat est identique à celle du précédent en considérant l'ensemble des états finals de $\mathcal{A}_0$ par

$$F = \{(q, p, l) \in Q_0 \mid q, p \in F\}.$$

□

Nous ne pouvons cependant pas considérer les langage $B(L)$ où $S(L)$ comme langage des représentants car rien n'assure qu'il est envoyé surjectivement sur G. Nous considérons alors l'ordre lexicographique (nous classons les élément par longueur et dans une même classe de longueur, nous considérons l'ordre de l'alphabet). Pour tout élément du groupe, nous choisissons ensuite le plus petit mot de L pour cet ordre qui est envoyé sur cet élément.

Plus formellement, si $<$ est un ordre sur l'alphabet Σ, alors nous l'étendons en un ordre total[13] *ShortLex* sur Σ^* comme suit. Le couple (u, v) de mots sur Σ appartient à *ShortLex* si $|u| < |v|$ ou si $|u| = |v|$ et il existe des mots z, u', v' de Σ^* et des lettres a, b de Σ, $a < b$, vérifiant $u = zau'$ et $v = zbv'$. La relation *ShortLex* est synchrone. En effet, l'automate représenté à la figure 2.8 l'accepte.

Le théorème suivant stipule que si pour tout élément du groupe, nous ne conservons, dans le langage des représentants, que le représentant le plus petit pour l'ordre lexicographique, nous obtenons un langage régulier L' tel que la structure (Σ, L') soit automatique.

[13]Rappelons qu'un ordre total est un ordre pour lequel deux éléments peuvent toujours être comparés.

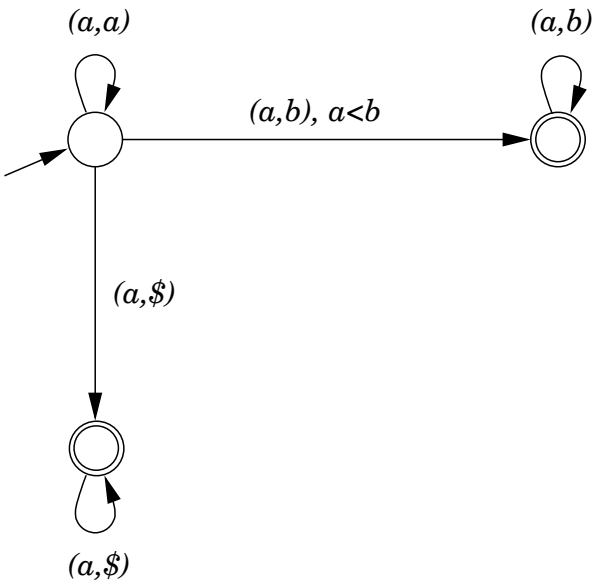

FIG. 2.8 – Automate normalisé acceptant la relation *ShortLex*.

Théorème 2.5.3 (Propriété d'unicité)**.** *Si* (Σ, L) *est une structure automatique sur* G*, alors le langage* $L' \subset L$ *défini ci-dessus est tel que* (Σ, L') *soit une structure automatique sur* G*.*

Démonstration. Le langage L' est en fait

$$L' = L \backslash \pi_2(R_\varepsilon \cap ShortLex).$$

puisque nous devons éliminer, dans chaque classe d'équivalence de L, tous les mots qui ne sont pas minimum pour l'ordre lexicographique. De plus, le langage L' est clairement régulier[14] et le couple (Σ, L') est donc une structure rationnelle sur G.

Montrons que (Σ^*, L') est également une structure automatique sur G. Il suffit de montrer que les relations R'_σ sont synchrones pour tout σ appartenant $\Sigma \cup \{\varepsilon\}$. Tout d'abord, il est facile de voir que la relation égalité

$$R_\varepsilon = \{(u, u) \in \Sigma^* \times \Sigma^* | u \in L'\}$$

est synchrone. Ensuite, il suffit de remarquer que, pour toute lettre a de Σ, nous avons

$$R'_a = R_a \cap (L' \times L').$$

La relation $L' \times L'$ étant synchrone vu le théorème 1.3.1, la relation R'_a est synchrone. □

Une autre simplification naturelle est de montrer que le langage des représentants peut être pris préfixiel.

[14] En effet, nous pouvons l'écrire $L' = L \cap (\Sigma^* \backslash (R_\varepsilon \cap ShortLex))$.

Théorème 2.5.4 (Proprété préfixielle)**.** *Si* (Σ, L) *est une structure automatique sur le groupe* G*, alors il existe une structure automatique* (Σ, L') *sur* G *telle que le langage* L' *est préfixiel.*

Démonstration. Il suffit de prendre comme langage des représentants le langage $L' = \mathrm{Pref}(L)$ qui est régulier. Considérons un automate $\mathcal{A}'$ acceptant le langage L'.

Montrons que (Σ, L') est une structure automatique sur G. Soient σ un élément de $\Sigma \cup \{\varepsilon\}$ et (u, v) un couple de R'_σ. Si n est le nombre d'états de l'automate $\mathcal{A}'$, alors il existe des mots z et t sur Σ de longueurs inférieures à n tels que les uz et vt soient des mots de L. Par le corollaire 2.3.2, pour tout mot w sur Σ, il existe un entier K_w tel que la relation R_w satisfait la K_w-propriété du compagnon de voyage.

Posons $K = sup\{K_w \mid |w| \leq 2n+1\}$. Le couple (uz, vt) appartient à la relation $R_{z^{-1}at}$ et $|z^{-1}at|$ est inférieur à $2n+1$. Un préfixe synchrone (u', v') de (u, v) est également un préfixe synchrone de (uz, vt). Il en découle que $d_G(\bar{u}', \bar{v}') \leq K$, d'où la conclusion.

□

2.6 Le problème du mot

Nous montrons dans cette section que, pour les groupes automatiques, le problème du mot est décidable en un temps quadratique en la longueur de l'instance.

Commençons par quelques rappels.

Un problème de décision est un problème pour lequel la réponse est *oui* ou *non*. Les données du problème sont appelées les *instances*. Une instance pour laquelle la réponse est oui (resp. non) est dite *positive* (resp. *négative*). Un problème est décidable s'il existe un algorithme qui permet, au moyen de codages adéquats, de décider si une instance donnée est positive ou négative. Dans le cas contraire, le problème est dit *indécidable*.

Avant de formaliser le problème du mot, nous devons introduire une nouvelle façon de définir un groupe, celle par *générateurs et relations*.

2.6.1 Introduction informelle

Si un groupe G est généré par un ensemble de générateurs de semigoupe Σ, il est possible d'écrire tout élément de G comme un produit

$$a_1^{\varepsilon_1} \dots a_n^{\varepsilon_n}$$

où les a_i sont des éléments de Σ et les ε_i des entiers relatifs.

Si G n'est pas un groupe libre, cette écriture n'est pas nécessairement unique. Pour arriver à retrouver le groupe G, il faut préciser quels produits sont égaux à l'élément neutre du groupe ; ces produits sont les *relations* du groupe[15]. Il est alors intuitivement clair que l'on peut retrouver le groupe, du moins à isomorphisme près. En fait, il n'est en général pas nécessaire de préciser toutes les relations possibles, puisqu'à partir de certaines relations de base, nous pouvons en déduire des relations qui en sont les conséquences. Par exemple, si a et b sont deux éléments de Σ et si nous savons que $abab = 1$, alors nous pouvons en déduire que $(ab)^4 = 1$ et ainsi de suite.

Nous arrivons ainsi à la notion intuitive de définition d'un groupe par générateurs et relations, c'est-à-dire par une présentation : il s'agit de spécifier un ensemble de générateurs, le Σ ci-dessus, et un ensemble R de relations, qui s'expriment comme des produits d'éléments de Σ. Nous disons alors que G est le groupe généré par Σ, dont les générateurs vérifient uniquement les relations spécifiées par R, ainsi que leurs conséquences ; nous notons la présentation par $\langle \Sigma; R \rangle$.

Avant même de donner une définition plus précise, nous pouvons donner quelques exemples évidents. Le groupe cyclique $\mathbb{Z}/n\mathbb{Z}$ est généré par un élément, à savoir la classe de 1. Si nous notons cet élément a, la seule relation que nous imposons est $a^n = 1$.

Vu la définition des groupes libres, il est clair que la présentation du groupe libre sur $\{a, b\}$ est $\langle \{a, b\}, \phi \rangle$. En effet, par définition, il n'existe aucune relation non triviale dans un groupe libre.

[15] Nous avons déjà rencontré ce concept lors de l'étude des groupes libres.

2.6.2 Définition formelle

Soient Σ un alphabet fini et R une partie du groupe libre $F(\Sigma)$, i.e. un ensemble de mots réduits sur $\Sigma \cup \Sigma^{-1}$. Considérons le plus petit sous-groupe normal[16] N de $F(\Sigma)$ contenant R. Ce sous-groupe est en fait l'ensemble des mots de la forme

$$\prod_{i=1}^{n} w_i r_i^{\pm 1} w_i^{-1},$$

où w_i appartient à $F(\Sigma)$ et r_i appartient à R pour tout i.

Si G est un groupe qui admet Σ comme ensemble de générateurs de semigroupe, nous étendons l'application p définie en 2.1 à l'ensemble $\Sigma \cup \Sigma^{-1}$ en considérant une involution sur Σ et nous obtenons donc un homomorphisme de groupes $\pi : F(\Sigma) \to G$. Si le noyau de cet homomorphisme, noté $ker(\pi)$, est le plus petit sous-groupe normal de $F(\Sigma)$ contenant R, nous disons que R est un ensemble de *relations* sur G et que Σ et R forment une *présentation de groupe* pour G ; nous notons $G = \langle \Sigma; R \rangle$.

L'ensemble R est en fait l'ensemble des relations que nous voulons imposer lorsque nous générons un groupe G. Pour cela, nous devons quotienter le groupe libre $F(\Sigma)$ par R ou plutôt par le plus petit sous-groupe normal de $F(\Sigma)$ contenant R afin que l'ensemble quotienté obtenu jouisse d'une structure de groupe. La définition précédente convient donc puisque le premier théorème d'isomorphie[17] stipule que le groupe G est isomorphe au groupe quotient $F(\Sigma)/N$.

Pour faire le lien avec l'introduction informelle ci-dessus, nous pouvons remarquer que les éléments de N sont en fait les "conséquences" des relations R. Nous pouvons également remarquer que tout groupe admet une présentation, à savoir la présentation $\langle F(G); G \rangle$. Par contre, une présentation n'est pas nécessairement unique.

Un groupe est *finiment généré* s'il est généré par un ensemble fini Σ. Il est *finiment présenté* s'il admet une présentation de la forme $\langle \Sigma; R \rangle$, où les ensemble Σ et R sont finis. Tout groupe finiment présenté est donc finiment généré, mais la réciproque n'est pas nécessairement vraie.

Les groupes automatiques étant finiment générés, nous ne considérons dans la suite que des ensembles finis de générateurs.

---

[16] Rappelons qu'un sous-groupe H de G est *normal* si

$$\forall x \in G,\ xHx^{-1} = H.$$

Nous notons $H \triangleleft G$.

[17] Si φ est un homomorphisme défini sur un groupe G et à valeurs dans un groupe H, alors

1. l'image par φ de G est un sous-groupe de H,
2. le noyau de φ est un sous-groupe normal de G et
3. le groupe H est isomorphe au groupe quotient $G/ker(\varphi)$.

2.6.3 Le problème du mot

Pour un groupe G de présentation $\langle \Sigma; R \rangle$, il est difficile de savoir si deux mots donnés sur Σ représentent ou non le même élément dans G ; c'est le *problème du mot*. Dans notre cas, nous avons supposé l'existence d'une involution ; nous pouvons formuler ce problème autrement : un mot donné sur Σ est-il envoyé sur l'identité du groupe ? En effet, si nous voulons savoir si les mots w_1 et w_2 sont envoyés sur le même élément, il nous suffit de vérifier que le mot $w_1 w_2^{-1}$ est envoyé sur l'identité.

Plus formellement , une instance du problème est un mot w sur Σ. La question qui est alors posée est de savoir s'il existe un entier m positif, des relations $r_1, r_2, \ldots, r_m$ dans R et des éléments $y_1, y_2, \ldots, y_m$ de $F(\Sigma)$ tels que

$$w \sim (y_1^{-1} r_1 y_1)(y_2^{-1} r_2 y_2) \cdots (y_m^{-1} r_m y_m) \tag{2.3}$$

et tels que le mot réduit obtenu à partir de cette décomposition est le mot de départ w. Sans cette dernière condition, il suffirait d'écrire $w \sim r$ pour un mot r appartenant à R et la réponse serait à chaque fois *oui*.

Il est clair qu'un mot décomposé de cette façon est envoyé sur l'identité et le problème du mot peut donc être formulé de la sorte. La question qui est posée est en fait "un mot w donné peut-il s'écrire comme un produit d'éléments du sous-groupe normal N de $F(\Sigma)$ engendré par R ?".

Revenons quelques instants aux graphes de Cayley. Si un groupe G admet une présentation $\langle \Sigma; R \rangle$, une relation w de R décrit un chemin fermé dans le graphe de Cayley associé au groupe G. La décomposition obtenue en (2.3) est en fait une subdivision de ce chemin en un nombre m de chemins fermés d'exécutions respectives r_i, pour $i = 1, \ldots, m$.

Illustrons ceci sur un exemple. Considérons le groupe[18] G admettant la présentation $\langle \{x, y\}; \{r_1, r_2\} \rangle$, où $r_1 = x^3$ et $r_2 = xyx^{-1}y^{-1}$. Considérons le mot réduit

$$w = y^2 x^{-1} y^{-1} x^{-1} y^{-1} x^{-1}.$$

Ce mot peut être décomposé en le produit

$$(x^{-1} r_2 x)(x^{-1} y r_2 y^{-1} x)(x^{-2} r_2 x^2)(r_1^{-1})$$

et si nous remplaçons les relations r_1 et r_2 par les mots auxquels elles correspondent et que nous réduisons le mot obtenu, nous retrouvons bien le mot de départ w. En effet,

$$\begin{aligned} &(x^{-1}xyx^{-1}y^{-1}x)(x^{-1}yxyx^{-1}y^{-1}y^{-1}x)(x^{-2}xyx^{-1}y^{-1}x^2)(x^{-3}) \\ &= x^{-1}xyx^{-1}y^{-1}xx^{-1}yxyx^{-1}y^{-1}y^{-1}xx^{-2}xyx^{-1}y^{-1}x^2x^{-3} \\ &= y^2x^{-1}y^{-1}x^{-1}y^{-1}x^{-1}. \end{aligned}$$

La figure 2.9 représente une partie suffisante du graphe de Cayley de G pour illustrer la décomposition faite ci-dessus.

[18] Nous pourrions montrer que ce groupe est en fait isomorphe au groupe $\mathbb{Z} \times \mathbb{Z}_3$.

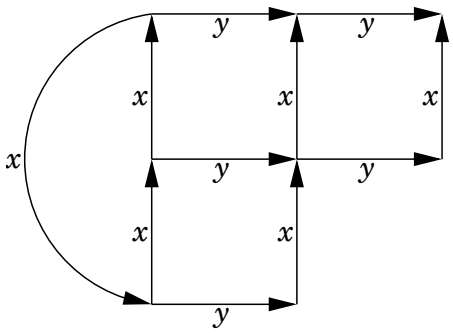

FIG. 2.9 – Partie du graphe de Cayley du groupe G admettant la présentation $\langle\{x,y\};\{x^3, xyx^{-1}y^{-1}\}\rangle$.

Nous introduisons maintenant la notion de *fonction isopérimétrique* d'une présentation. Il existe[19] un théorème qui stipule que le problème du mot est décidable dans un groupe si et seulement si celui-ci admet une fonction isopérimétrique récursive[20]. Ce théorème n'est pas démontré dans ce travail ; nous allons cependant montrer que dans le cas où le groupe présenté est automatique, le problème du mot est décidable en un temps quadratique en la longueur de l'instance w.

A tout mot w sur $F(\Sigma)$ envoyé sur l'identité du groupe, nous associons une *aire* définie comme étant le plus petit entier m vérifiant la décomposition (2.3) ; nous la notons m_w. La *fonction isopérimétrique* φ d'une présentation est la fonction qui à tout entier n, associe la plus grande aire m_w associée à un mot réduit de longueur inférieure à n qui est envoyé sur l'identité du groupe, i.e.

$$\varphi(n) = sup\{m_w \mid w \in F(\Sigma),\ |w| \leq n \ et\ \bar{w} = 1\}.$$

Il est évident que cette fonction dépend de la présentation choisie. Cependant, nous pouvons montrer que si nous disposons de deux présentations différentes pour un même groupe G, leurs fonctions isopérimétriques respectives se comportent de la même façon en ce sens qu'elles sont du même ordre de grandeur.

Plus précisément, considérons deux présentations $\langle\Sigma_1; R_1\rangle$ et $\langle\Sigma_2; R_2\rangle$ d'un même groupe G et notons φ_1 et φ_2 leurs fonctions isopérimétriques respectives. Puisque Σ_1 et Σ_2 sont deux ensembles de générateurs de semigroupe pour G, il existe une fonction θ_1 définie sur $(\Sigma_1 \cup \Sigma_1^{-1})^*$, à valeurs dans $(\Sigma_2 \cup \Sigma_2^{-1})^*$ et telle que $\theta_1(u)$ est le plus petit mot sur $\Sigma_2 \cup \Sigma_2^{-1}$ qui a la même image que u dans G. De même, il existe une fonction θ_2 définie sur $(\Sigma_2 \cup \Sigma_2^{-1})^*$, à valeurs dans $(\Sigma_1 \cup \Sigma_1^{-1})^*$ et telle que $\theta_2(v)$ est le plus petit mot sur $\Sigma_1 \cup \Sigma_1^{-1}$ qui a la même image que v dans G.

Posons

$$N_1 = sup\{aire(\theta_1(r)) \mid r \in R_1\}$$

[19] Voir [2] pour la démonstration de ce théorème.

[20] Rappelons qu'une fonction récursive est une fonction calculable par alorithme.

et

$$N_2 = sup\{|\theta_2(a)| \mid a \in \Sigma_2 \cup \Sigma_2^{-1}\}.$$

Nous obtenons

$$\varphi_2(i) \leq N_1\varphi_1(N_2 i),\ \forall i \in \mathbb{N}.$$

Par conséquent, si la fonction φ_1 est majorée par une fonction polynomiale, expoentielle ou récursive, il en est de même pour la fonction φ_2. Dans ce cas, nous disons que le groupe G satisfait une inégalité isopérimétrique polynomiale, exponentielle ou récursive. Nous verrons dans la suite que les groupes automatiques satisfont une inégalité isopérimétrique *quadratique*.

Lemme 2.6.1. *Soient G un groupe automatique et (Σ, L) une structure automatique sur G. Il existe un entier N tel que pour tout mot w dans L et pour tout élément x de G vérifiant $d(x, \bar{w}) \leq 1$,*

1. *l'élément x de G admet un représentant v dans L de longueur inférieure à $|w|+N$,*
2. *si un représentant de x dans L a une longueur strictement supérieure à cette borne, alors x admet une infinité de représentants dans L.*

Il découle directement du point 1 que tout mot w sur Σ est équivalent à un mot de L de longueur inférieure ou égale à $N|w| + n_0$, où n_0 est la longueur d'un représentant dans L de l'identité du groupe.

Démonstration. Pour tout élément σ de $\Sigma \cup \{\varepsilon\}$, nous considérons l'automate multiplicateur $\mathcal{M}_\sigma$, que nous supposons déterministe, associé à la relation R_σ et nous notons par N_σ le nombre d'états de cet automate.

Posons

$$N = sup\{N_\sigma \mid \sigma \in \Sigma \cup \{\varepsilon\}\}.$$

Soit w' un représentant de x dans L. Comme $\bar{w}$ et x ne sont distants que d'au plus un générateur dans le graphe de Cayley de G, le couple (w, w') appartient à la relation R_σ pour un élément σ de $\Sigma \cup \{\varepsilon\}$.

Si la longueur de w' est strictement supérieure à $|w| + N$, alors après avoir lu $|w|$ couples de lettres, l'automate $\mathcal{M}_\sigma$ exécute encore plus de N transitions et le chemin d'exécution $(w, w')^\$$ passe donc au moins deux fois par le même état ; il dessine par conséquent un chemin fermé dans le graphe. Nous pouvons donc raccourcir le mot w' en éliminant ce chemin jusqu'à arriver à un mot de la longueur voulue. Le mot obtenu est toujours un mot de L qui représente x, ce qui prouve le premier point.

Le deuxième point est alors immédiat. Il suffit de prolonger le mot w' en répétant un nombre arbitraire de fois le chemin fermé.

Pour la dernière assertion, considérons un mot $w = a_1 \cdots a_n$ sur Σ. Vu le premier point, il existe des mots $u_0, \ldots, u_n$ dans L tels que

$$\bar{u}_0 = 1,\ \bar{u}_1 = a_1, \ldots,\ \bar{u}_i = \bar{u}_{i-1}\bar{a}_i, \ldots,\ \bar{u}_n = \bar{u}_{n-1}\bar{a}_n = \bar{w},$$

avec $|u_i| \leq |u_{i-1}| + N$ pour tout i, d'où la conclusion. □

Théorème 2.6.2. *Soit G un groupe automatique et (Σ, L) une structure automatique sur G. Pour tout mot w sur Σ, nous pouvons trouver un représentant v dans L de $\bar{w}$ en un temps quadratique en la longueur de w.*

Démonstration. Soit w un mot sur Σ. Nous pouvons érire $\bar{w} = \bar{u}\bar{a}$ pour un mot u dans L et un générateur a dans Σ.

Nous commençons par montrer que, pour un mot u dans L, nous pouvons trouver un représentant de $\bar{u}\bar{a}$ en un temps $O(|u|)$[21] et ce représentant a une longueur majorée par $|u| + N$.

Nous devons trouver un mot v dans L tel que $\bar{u}\bar{a} = \bar{v}$. Considérons l'automate multiplicateur $\mathcal{M}_a = (Q, q_0, F, \Delta)$ associé à la relation R_a ; nous le supposons déterministe. L'idée est que $\mathcal{M}_a$ accepte un couple (u', v') de mots sur $\Sigma^\$$ où u' est le mot u éventuellement suivi d'occurences de \$ et v' est un représentant de $\bar{u}\bar{a}$, lui aussi éventuellement suivi d'occurences de \$.

Nous construisons l'automate $\mathcal{N}_a$ à partir de $\mathcal{M}_a$ dans lequel nous ignorons la deuxième composante. Rien n'assure que l'automate $\mathcal{N}_a$ soit encore déterministe.

Le mot u' est accepté par l'automate $\mathcal{N}_a$; il décrit donc un chemin accepteur dans le graphe associé à l'automate. Or, les graphes associés respectivement aux automates $\mathcal{M}_a$ et $\mathcal{N}_a$ sont les mêmes. Pour trouver le mot v', il nous suffit donc de lire la deuxième composante de l'exécution de ce chemin dans le graphe associé à $\mathcal{M}_a$.

Plus formellement, si u' est le mot $a_1 \cdots a_m$ avec a_i appartenant à Σ pour $i \leq |u|$ et $a_i = \$$ pour $|u| < i \leq m$, nous construisons les suites $Q_0, \ldots, Q_m$ et $T_0, \ldots, T_{m-1}$ comme suit.

Nous posons $Q_0 = \{q_0\}$. Pour tout $i > 0$, nous définissons T_i comme l'ensemble des arcs, dans le graphe associé à $\mathcal{M}_a$, dont la source est dans Q_{i-1} et dont le label est le couple (a_i, y_i), avec $y_i \in \Sigma^\$$. Nous définissons aussi Q_i comme l'ensemble des états atteints par les arcs de T_i.

Soit n le plus petit entier vérifiant $n \geq |u|$ et tel que Q_n contient un état final. Il existe un chemin de longueur n, dans le graphe associé à $\mathcal{M}_a$, joignant q_0 à un état final. Nous parcourons ce chemin à l'envers et nous lisons les labels $v_n, \ldots, v_1$. Nous formons alors le mot $v' = v_1 \cdots v_n$ sur $\Sigma^\$$. Enfin, nous obtenons un représentant v dans L de $\bar{u}\bar{a}$ en éliminant de v' les occurences de \$.

L'automate $\mathcal{M}_a$ étant fini, chaque étape de la construction ci-dessus se fait en un temps borné. Le temps total requis est donc linéaire en n. De plus, par le lemme précédent, l'entier n ne peut excéder $|u|$ que d'une constante N, car cela contredirait le caractère minimal[22] du mot v. Nous avons donc trouvé, en un temps $O(|u|)$, un représentant v de $\bar{u}\bar{a}$ dans L de longueur majorée par $|u| + N$.

[21] Une fonction f appartient à $O(g)$ s'il existe un entier N et une constante $C > 0$ tels que

$$\forall n \geq N,\ f(n) \leq Cg(n).$$

.

[22] Nous avons choisi le plus petit entier n pour que ça marche.

Nous pouvons maintenant passer à la démonstration même du théorème.

Supposons que le mot w s'écrive $a_1 \cdots a_n$, avec $a_i \in \Sigma$ pour tout i. Si u_0 est un représentant de longueur l dans L de l'identité du groupe, nous construisons succesivement les mots $u_1, \ldots, u_n$ de L tels que

$$\bar{u}_0 = 1,\ \bar{u}_1 = \bar{a}_1, \ldots,\ \bar{u}_i = \bar{u}_{i-1}\bar{a}_i, \ldots,\ \bar{u}_n = \bar{u}_{n-1}\bar{a}_n = \bar{w},$$

avec $|u_i| \leq |u_{i-1}| + N$ pour tout i.

En particulier, nous avons trouvé un représentant u_n dans L de $\bar{w}$ et ce représentant a une longueur majorée par $l + |w|N$.

Si u_0 est connu, le temps total pour calculer ces mots est majoré par

$$\sum_{i=1}^{n}(l + iN) \in O(nl) + O(n^2).$$

Il suffit donc de montrer qu'il est possible de trouver un représentant dans L de l'identité du groupe en un temps borné.

Il est possible d'identifier un mot $v = b_1 b_2 \cdots b_r$ de L en un temps constant en considérant un chemin accepteur arbitraire dans le graphe associé à l'automate acceptant le langage L. Nous utilisons ensuite les constructions faites ci-dessus pour définir la suite de mots v_i de L pour $0 \leq i \leq r$ en posant $v_0 = v$ et

$$\bar{v}_1 = \bar{v}_0\bar{b}_r^{-1},\ \bar{v}_2 = \bar{v}_1\bar{b}_{r-1}^{-1}, \ldots,\ \bar{v}_r = \bar{v}_{r-1}\bar{b}_1^{-1} = 1.$$

Cette opération se fait en un temps $O(r^2)$ mais peut être réalisée une fois pour toutes[23]. Nous obtenons donc un représentant dans L de l'identité en un temps borné, d'où la conclusion.

□

Théorème 2.6.3. *Soit G un groupe automatique. Nous disposons des propriétés suivantes.*

1. *Le groupe G est finiment présenté,*
2. *le groupe G admet une fonction isopérimétrique quadratique,*
3. *le problème du mot est solvable en un temps quadratique.*

Avant de passer à la démonstration de ce théorème, il paraît nécessaire de rappeler quelques notions concernant les actions de groupe.

Une *action* d'un groupe G sur un ensemble S est un homomorphisme $j : x \mapsto j_x$ de G dans le groupe $B(S)$ des bijections de S dans lui-même. Nous disons alors que G opère sur S et dans ce cas, nous définissons l'orbite d'un élément s de S par

$$Orb(s) = \{j_x(s) |\ x \in G\}$$

[23]Ici, le r ne varie pas, nous utilisons un mot de L pour trouver un représentant de l'identité et nous utilisons ce représentant à chaque fois.

et le stabilisateur de s par

$$Stab(s) : \{x \in G |\ j_x(s) = s\}.$$

Nous vérifions aisément que $Stab(s)$ est un sous-groupe de G.

Définissons maintenant les actions par conjugaison. Si G est un groupe et si x est un élément de G, nous notons σ_x l'automorphisme de G défini par $\sigma_x(y) = xyx^{-1}$. Il est clair que le groupe G opère par conjugaison sur lui-même par l'action $x \mapsto \sigma_x$. En effet, nous avons

$$\sigma_e = id$$

et

$$\sigma_{xy}(z) = xyz(xy)^{-1} = xyzy^{-1}x^{-1} = \sigma_x(\sigma_y(z)).$$

Passons maintenant à la démonstration du théorème.

Démonstration. 1. L'alphabet d'une structure automatique étant fini, il est clair que le groupe G est finiment généré. Vu le théorème 2.5.3, nous pouvons supposer que le langage L satisfait la propriété d'unicité.

Pour tous mots u et v sur Σ, nous notons $u \sim v$ si $\bar{u} = \bar{v}$. Il est clair que la relation $\sim$ est une relation d'équivalence.

Considérons un mot $w = a_1 \cdots a_n$ sur Σ envoyé sur l'identité du groupe. Nous notons par u_i l'unique mot de L envoyé sur $\overline{w(i)}$, où $w(i)$ est le préfixe de w de longueur i, avec $i = 0, 1, \ldots, n$. Nous obtenons

$$w \sim (u_0 a_1 u_1^{-1})(u_1 a_2 u_2^{-1}) \cdots (u_{n-1} a_n u_n^{-1}).$$

Montrons maintenant que chacun des facteurs de la forme $u_i a_{i+1} u_{i+1}^{-1}$ peut être factorisé en un produit de conjugués de mots[24] de longueur bornée.

Nous pouvons supposer sans perte de généralité que $p = |u_i|$ est inférieur à $m = |u_{i+1}|$. Posons

$$u_i = b_1 \cdots b_m,$$

avec $b_i \in \Sigma$ pour $i \leq p$ et $b_i = \varepsilon$ pour $p + 1 \leq i \leq m$, et

$$u_{i+1} = c_1 \cdots c_m,$$

où les c_i appartiennent à Σ pour tout i.

Pour tout entier t de $\{0, 1, \ldots, m\}$, nous posons $u_i(t) = b_1 \cdots b_t$ et nous notons par $u_{i+1}(t)$ le préfixe de u_{i+1} de longueur t. Vu la définition des u_i, il est clair que le couple (u_i, u_{i+1}) appartient à la relation synchrone $R_{a_{i+1}}$ et la distance $d(\bar{u}_i(t), \bar{u}_{i+1}(t))$ est donc majorée pour tout t par une constante k provenant de la propriété du compagnon

[24] Nous appelons *conjugué de mot* un mot w tel qu'il existe des mots u et v vérifiant $w = vuv^{-1}$.

de voyage. Nous obtenons donc, pour tout t, $u_i(t) \sim u_{i+1}(t)v(t)$, où $v(t)$ est un mot de longueur inférieure à k. Il est alors facile de vérifier que

$$u_i a_{i+1} u_{i+1}^{-1} \sim u_i(1)v(1)u_{i+1}(1)^{-1} \prod_{1 \leq t \leq m-1} u_{i+1}(t)v(t)^{-1}b_{t+1}v(t+1)u_{i+1}(t+1)^{-1}. \quad (2.4)$$

De plus, nous avons $u_{i+1}(t+1) = u_{i+1}(t)c_{t+1}$. Le t-ième facteur du produit ci-dessus est donc le conjugué par $u_{i+1}(t)$ de l'élément

$$v(t)^{-1}b_{t+1}v(t+1)c_{t+1}^{-1},$$

qui a une longueur inférieure à $2k+2$.

L'ensemble des relations de la présentation peut donc être pris comme un sous-ensemble de mots de longueurs inférieures à $2k+2$. Il n'existe donc qu'un nombre fini de relations et le groupe est bien finiment présenté.

2. Cela découle directement des constructions faites ci-dessus. En effet, vu le lemme 2.6.1, chaque mot u_i a une longueur linéairement bornée par $|w|$. Le nombre de termes dans le deuxième membre de l'équivalence (2.4) est donc également linéairement borné par $|w|$. Comme nous avons exactement $|w|$ facteurs de la forme $u_i a_{i+1} u_{i+1}^{-1}$, la fonction isopérimétrique est bien majorée par une fonction quadratique.

3. Soit $w = a_1 \cdots a_n$ un mot sur Σ. Nous procédons exactement comme dans le théorème 2.6.2 et nous trouvons un représentant u_n dans L de $\bar{w}$ en un temps $O(|w|^2)$. Nous considérons également le mot v_r de L envoyé sur l'identité comme nous l'avons défini dans ce même théorème 2.6.2.

Pour vérifier que le mot w est envoyé sur l'identité, il suffit donc de vérifier que le couple (u_n, v_r) de mots appartient à la relation R_ε, ce qui peut être vérifié en un temps proportionnel à $|u_n|$[25].

□

[25]Nous supposons que v_r est un mot de longueur inférieur à $|u_n|$.

2.7 Propriétés de fermeture

Le nombre de propriétés de fermeture satisfaites par la classe des groupes automatiques montre la robustesse du concept. Nous démontrons, dans cette section, deux résultats principaux concernant le produit direct de deux groupes et le passage aux sous-groupes finis.

Théorème 2.7.1. *Si G_1 et G_2 sont deux groupes automatiques, alors le produit direct $G = G_1 \times G_2$ l'est aussi.*

En particulier, tout produit fini de groupes automatiques est encore un groupe automatique.

Rappelons que le produit direct de deux groupes $(G_1, \circ)$ et $(G_2, \bullet)$ est le groupe $G = (G_1 \times G_2, \star)$, défini par $G_1 \times G_2 = \{(x_1, x_2) \in G \mid x_1 \in G_1,\ x_2 \in G_2\}$ et où l'opération $\star$ est définie composante à composante par

$$(x_1, x_2) \star (y_1, y_2) = (x_1 \circ y_1, x_2 \bullet y_2).$$

Si e_1 et e_2 sont les neutres respectifs de G_1 et de G_2, alors le couple (e_1, e_2) est le neutre de G. Enfin, l'inverse de l'élément (x_1, x_2) de G est l'élément (x_1^{-1}, x_2^{-1}).

Passons maintenant à la démonstration du théorème.

Démonstration. Soient (Σ_1, L_1) et (Σ_2, L_2) des structures automatiques sur G_1 et G_2 respectivement. Nous pouvons supposer que les deux alphabets Σ_1 et Σ_2 sont disjoints l'un de l'autre[26] et que les langages des représentants L_1 et L_2 satisfont la propriété d'unicité.

Posons $\Sigma = \Sigma_1 \cup \Sigma_2$. Les deux langages L_1 et L_2 étant réguliers respectivement sur Σ_1 et Σ_2, ils le sont aussi sur Σ, ainsi que leur concaténation $L = L_1 L_2$.

Nous considérons l'homomorhisme π défini sur Σ^* qui étend naturellement les homomorphismes respectifs relatifs aux structures (Σ_1, L_1) et (Σ_2, L_2). À tout mot $w = a_1 \cdots a_n$ sur Σ, π associe l'élément $(\bar{w}_1, \bar{w}_2)$ de G, où w_1 et w_2 sont des mots sur Σ_1 et Σ_2 respectivement construits de manière récursive comme suit. Les deux mots sont initialement vides et l'entier i est initialisé à 1. Si a_i appartient à Σ_1 (resp. à Σ_2), alors w_1 (resp. w_2) devient $w_1 a_i$ (resp.$w_2 a_i$) et w_2 (resp. w_1) reste inchangé. L'entier i augmente ensuite d'une unité. Nous répétons cette opération jusqu'à obtenir $i = n+1$.

En fait, le morphisme π ne fait qu'associer les images dans G_1 et G_2 des mots w_1 et w_2 formés des lettres de w appartenant à Σ_1 et Σ_2 respectivement. En particulier, à un mot $u_1 u_2$ de L, π associe l'élément $(\bar{u}_1, \bar{u}_2)$ de G.

Nous disposons donc maintenant d'une structure rationnelle (Σ, L) sur G. Il suffit de vérifier que toutes les relations R_σ, $\sigma \in \Sigma \cup \{\varepsilon\}$, de cette structure satisfont la propriété du compagnon de voyage.

Les groupes G_1 et G_2 étant automatiques, nous disposons de deux constantes k_1 et k_2 découlant de la propriété du compagnon de voyage pour ces deux groupes.

[26] Quite à considérer l'alphabet $\bar{\Sigma}_2 = \{\bar{a} |\ a \in \Sigma_2\}$ à la place de Σ_2 si ce n'est pas le cas.

Les deux groupes G_1 et G_2 étant automatiques, le lemme 2.6.1 assure l'existence de deux constantes N_1 et N_2 telles que pour tout couple (u, v) de mots sur Σ_1 (resp. sur Σ_2) appartenant à une relation R_σ pour un élément σ de Σ_1 (resp. de Σ_2), nous avons l'inégalité

$$||u| - |v|| \leq N_1 \text{ (resp. } N_2).$$

Pour tout couple (u, v) de mots sur Σ appartenant à une relation R_σ, $\sigma \in \Sigma \cup \{\varepsilon\}$, nous avons donc l'inégalité

$$||u| - |v|| \leq N = N_1 + N_2.$$

Posons $k = max\{k_1, k_2\} + N$ et vérifions que la relation R_σ satisfait la k-propriété du compagnon de voyage pour tout σ appartenant à $\Sigma \cup \{\varepsilon\}$.

Nous devons distinguer trois cas.

1. Si $\sigma = \varepsilon$. Il suffit de remarquer que

$$R_\varepsilon = \{(u, u) \in \Sigma^* \times \Sigma^* \mid u \in L\}.$$

En effet, considérons deux mots $u = u_1u_2$ et $v = v_1v_2$ dans L, avec u_1, v_1 des mots de L_1 et u_2, v_2 des mots de L_2. Ces deux mots sont envoyés sur le même élément de G si et seulement si

$$\begin{cases} \bar{u}_1 = \bar{v}_1 \\ \bar{u}_2 = \bar{v}_2 \end{cases}$$

Or, les langages L_1 et L_2 satisfont la propriété d'unicité et cette condition est donc équivalente à

$$\begin{cases} u_1 = v_1 \\ u_2 = v_2 \end{cases}$$

ou encore à $u = v$.

Vu sa forme, il est alors clair que R_ε satisfait la propriété du compagnon de voyage.

2. Si $\sigma \in \Sigma_2$.

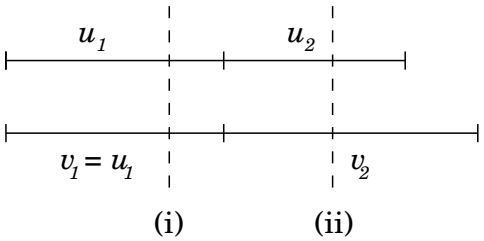

FIG. 2.10 – Choix des préfixes aynchrones : cas $\sigma \in \Sigma_2$.

Soit un couple (u_1u_2, v_1v_2) de mots appartenant à R_σ. Nous avons

$$\overline{u_1u_2\sigma} = \overline{v_1v_2},$$

c'est-à-dire

$$(\bar{u}_1, \bar{u}_2\bar{\sigma}) = (\bar{v}_1, \bar{v_2}).$$

Vu la propriété d'unicité, nous en déduisons que $u_1 = v_1$. Nous devons ensuite distinguer deux sous-cas.

(i) Soit (u_3, v_3) un préfixe synchrone de (u_1, v_1). Nous avons $u_3 = v_3$ et donc $d(\bar{u}_3, \bar{v}_3) = 0 \leq k$.

(ii) Soit maintenant (u_1u_3, v_1v_3) un préfixe synchrone de (u_1u_2, v_1v_2). Nous avons

$$d_{G_1\times G_2}((\bar{u}_1, \bar{u}_3), (\bar{v}_1, \bar{v}_3)) = d_{G_2}(\bar{u}_3, \bar{v}_3) \leq k_2 \leq k,$$

ce qui suffit.

3. Si $\sigma \in \Sigma_1$.

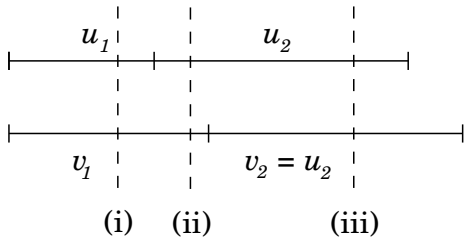

FIG. 2.11 – Choix des préfixes synchrones : cas $\sigma \in \Sigma_1$.

Soit (u_1u_2, v_1v_2) un couple de mots appartenant à R_σ. Nous pouvons supposer sans perte de généralité que $|u_1|$ est inférieur à $|v_1|$. Nous devons à nouveau distinguer trois sous-cas.

(i) Si le couple (u_3, v_3) de mots sur Σ est un préfixe synchrone de (u_1, v_1), alors

$$d_{G_1\times G_2}((\bar{u}_3, 1), (\bar{v}_3, 1)) = d_{G_1}(\bar{u}_3, \bar{v}_3) \leq k_1 \leq k.$$

(ii) Si le couple (u_1u_3, v_3) de mots sur Σ est un préfixe synchrone de (u_1u_2, v_1), alors

$$d_{G_1\times G_2}((\bar{u}_1, \bar{u}_3), (\bar{v}_3, 1)) = d_{G_1}(\bar{u}_1, \bar{v}_3) + d_{G_2}(\bar{u}_3, 1) \leq d_{G_1}(\bar{u}_1, \bar{v}_3) + |u_3| \leq k_1 + N \leq k,$$

où la majoration de $|u_3|$ par N découle de l'inégalité

$$|u_3| = |u_3| - |\varepsilon| \leq |v_1| - |u_1| \leq N_1.$$

(iii) Si le couple (u_1u_3, v_1v_3) est un préfixe synchrone de (u_1u_2, v_1v_2), le mot v_3 est un préfixe de u_3 et nous avons

$$|u_3| - |v_3| = |v_1| - |u_1| \leq N.$$

Puisque $(\overline{u_1u_3}, \overline{v_1v_3}) = (\bar{u}_1, \bar{u_3})(\bar{v}_1, \bar{v}_3)$, nous avons

$$d_{G_1\times G_2}(\overline{u_1u_3}, \overline{v_1v_3}) = d_{G_1}(\bar{u}_1, \bar{v}_1) + d_{G_2}(\bar{u}_3, \bar{v}_3) \leq k_1 + N \leq k,$$

d'où la conclusion.

Le cas général se démontre très facilement par réccurence sur le nombre de groupes dans le produit.

□

En conséquence immédiate, nous disposons du théorème suivant.

Théorème 2.7.2. *Tout groupe commutatif finiment généré est automatique.*

Démonstration. Soient $a_1, a_2, \ldots, a_k$ les générateurs d'un groupe commutatif G. Par commutativité, chaque élément du groupe peut s'écrire comme un produit

$$a_1^{n_1} a_2^{n_2} \cdots a_k^{n_k},$$

où les n_i sont des entiers relatifs.

Le groupe G est donc isomorphe au groupe $\langle a_1 \rangle \times \langle a_2 \rangle \times \cdots \times \langle a_k \rangle$. Or, il est évident qu'un groupe généré par un unique élément est automatique. Le produit des groupes $\langle a_i \rangle$ est donc lui aussi un groupe automatique, d'où la conclusion.

□

Le théorème suivant affirme qu'un groupe est automatique si est seulement si tous ses sous-groupes d'index fini le sont.

Théorème 2.7.3. *Soient G un groupe et H un de ses sous-groupes d'index fini. Le groupe G est automatique si et seulement si le sous-groupe H l'est.*

Démonstration. Comme H est un sous-groupe d'index fini de G, il existe des éléments $x_0, x_1, \ldots, x_n$ de G, où x_0 est l'identité du groupe, tels que

$$G = \bigcup_{0 \leq i \leq n} H x_i.$$

Posons $X = \{x_0, x_1, \ldots, x_n\}$.

Supposons pour commencer que le groupe G admet le couple (Σ, L) comme structure automatique où le langage L satisfait la propriété d'unicité. A chaque couple (x, a) de $X \times \Sigma$, nous associons l'unique couple (w, x') de $\Sigma^* \times X$ défini comme suit.

Vu la manière dont nous avons décomposé le groupe G en l'union de ses classes latérales, il existe un unique x' dans X tel que l'élément $x\bar{a}$ de G soit dans la classe Hx'. Nous choisissons alors l'unique mot w de L tel que $x\bar{a} = \bar{w}x'$.

Remarquons simplement que les ensembles X et Σ étant finis, le nombre de mots w définis de la sorte est fini. Notons W l'ensemble de ces mots.

La construction faite ci-dessus nous donne une application bien définie

$$X \times \Sigma \to \Sigma^* \times X.$$

Montrons que les mots de W génèrent le sous-groupe H. Soit un élément $z = a_0 a_1 \cdots a_m$ de G, où les a_i sont des lettres de Σ pour tout i. Nous avons

$$\overline{a_0 a_1 \cdots a_m} = x_0 \overline{a_0 a_1 \cdots a_m}$$

et donc

$$\begin{aligned} \overline{a_0 a_1 \cdots a_m} &= \bar{w}_0 x_{i_1} \overline{a_1 a_2 \cdots a_m} \\ &= \overline{w_0 w_1} x_{i_2} \overline{a_2 a_3 \cdots a_m} \\ &\vdots \\ &= \overline{w_0 w_1 \cdots w_m} x_{i_{m+1}}. \end{aligned} \tag{2.5}$$

L'élément z appartient donc au sous-groupe H si et seulement si $x_{i_{m+1}} = x_0 = 1$, ce qui suffit.

Construisons maintenant une structure automatique (Δ, L') sur H. Nous choisissons un alphabet Δ disjoint de Σ et de même cardinal que W. Nous définissons alors une bijection entre Δ et W et nous obtenons une application

$$X \times \Sigma \to \Delta \times X \tag{2.6}$$

qui s'étend à $X \times \Sigma^*$ grâce à l'égalité (2.5) et qui associe à tout couple (x, u) de $X \times \Sigma^*$ un couple (v, x') de $\Delta^* \times X$. Il est facile de voir que les ensembles $X \times \Sigma^*$ et $\Delta^* \times X$ sont en fait en bijection.

Nous pouvons voir l'application (2.6) comme un automate $\mathcal{A}$ sur l'alphabet $\Sigma \times \Delta$; l'ensemble des états est X et si le couple (x, a) de $X \times \Sigma$ est envoyé sur le couple (b, x') de $\Delta \times X$ par l'application (2.6), alors le triplet $(x, (a, b), x')$ est une transition de $\mathcal{A}$. L'état initial est x_0 et correspond également à l'unique état final[27].

Nous définissons alors le langage L' à partir de cet automate comme l'ensemble des mots v sur Δ pour lesquels il existe un mot w dans L tel que le couple (w, v) soit accepté par $\mathcal{A}$. Le langage L' est en fait l'image par l'application (2.6) des mots dans L qui sont aussi des relations reconnues par l'automate[28].

Le langage L' ainsi défini est régulier et tel que $\bar{L}' = H$.

Montrons que le couple (Δ, L') est une structure automatique sur H. Soit b une lettre de Δ et soit (v_1, v_2) un couple de la relation R'_b. Les mots v_1 et v_2 sont dans le langage L et sont donc respectivement l'image par l'application (2.6) de deux mots u_1 et u_2 de L.

[27] Si nous ignorons la deuxième composante des labels, l'automate $\mathcal{A}$ est en fait le graphe de Cayley du groupe G quotienté par le sous-groupe H.

[28] Ces mots ne sont pas nécessairement des relations pour le groupe G. Cependant, si nous multiplions un élément x d'un classe latérale Hx_i par l'image d'un tel mot, nous restons dans la même classe Hx_i. Dans le cas où le sous-groupe H est normal, ces mots sont des relations pour le groupe quotient G/H.

Soit (z_1, z_2) un préfixe synchrone de (v_1, v_2). Par la même application, il existe un préfixe synchrone (t_1, t_2) de (u_1, u_2) et des éléments x_i et x_j de X tels que $x_0\bar{t}_1 = \bar{z}_1 x_i$ et $x_0\bar{t}_2 = \bar{z}_2 x_j$. Nous avons donc

$$\begin{aligned} d_H(\bar{z}_1, \bar{z}_2) &= d_H(x_0\bar{t}_1 x_i^{-1}, x_0\bar{t}_2 x_j^{-1}) \\ &\leq \lambda d_G(\bar{t}_1, \bar{t}_2) + |x_i| + |x_j|, \end{aligned}$$

avec

$$\lambda = max\{|w| \ | w \in W\}.$$

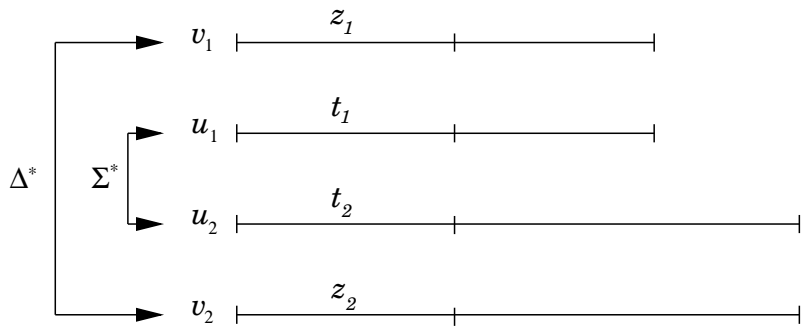

FIG. 2.12 – Correspondance entre les longueurs des préfixes z_i et t_i pour $t = 1, 2$.

Soit w le mot de L correspondant à la lettre b de Δ. Le couple (u_1, u_2) appartient à la relation R_w.

Comme le groupe G est automatique, il satisfait la k-propriété du compagnon de voyage pour un certain entier k. Posons

$$K = max\{|x| \ | \ x \in X\}.$$

Nous obtenons

$$d_H(\bar{z}_1, \bar{z}_2) \leq \lambda k + 2K$$

et le couple (Δ, L') est donc bien une structure automatique sur H.

Réciproquement, soit (Δ, L') une structure automatique sur H. Considérons un alphabet fini Σ disjoint de Δ et de même cardinal que X et définissons une bijection entre Σ et X.

Montrons que le couple $(\Sigma \cup \Delta, L'\Sigma)$ forme une structure automatique sur G. Tout d'abord, il est facile de voir que le langage $L'\Sigma$ est régulier et que nous avons $\overline{L'\Sigma} = G$.

Considérons un couple (ua, vb) dans la relation R_σ, $\sigma \in \Sigma \cup \Delta \cup \{\varepsilon\}$. Par définition, nous avons $\overline{ua\sigma} = \overline{vb}$, ce qui est équivalent à

$$\overline{ua\sigma}\bar{b}^{-1} = \bar{v}$$

et le couple (u, v) appartient donc à la relation $R_{a\sigma b^{-1}}$.

Or, les deux mots u et v appartiennent à L et il est possible de trouver un mot w sur Δ tel que le couple (u, v) appartienne à la relation R_w qui satisfait la K_w-propriété du compagnon de voyage.

De plus, il n'existe qu'un nombre fini de mots de la forme $a\sigma b^{-1}$ avec $a, b \in \Sigma$. Il est donc possible de trouver un entier K tel que la relation R_σ satisfait la K-propriété du compagnon de voyage, d'où la conclusion.

□

Chapitre 3

Groupes automatiques asynchrones

Dans ce chapitre, nous introduisons une classe de groupes plus large que celle des groupes automatiques synchrones, à savoir celle des *groupes automatiques asynchrones*. L'idée est sensiblement la même que celle rencontrée pour les groupes automatiques synchrones en ce sens que nous définissons également une structure sur le groupe composée de l'ensemble des générateurs Σ, d'un langage des représentants L ainsi que des automates multiplicateurs sur l'alphabet Σ. Cependant, les automates multiplicateurs sont différents du cas synchrone de par le fait qu'ils ne lisent pas simultanément les deux mots qui leurs sont proposés. Il s'agit d'*automates asynchrones*.

3.1 Automates asynchrones

Nous définissons ici les automates asynchrones et nous expliquons leur fonctionnement qui diffère légèrement de celui des automates que nous avons l'habitude de rencontrer. Nous étudions ensuite quelques résultats concernant les langages acceptés par ces automates.

Intuitivement, un *automate asynchrone* est une machine qui lit sur deux bandes, en l'occurence la *bande gauche* et la *bande droite*, chacune des deux contenant un mot. À tout instant, la machine sait sur quelle bande elle doit lire ; cette information est contenue dans l'état dans lequel elle se trouve. Nous utilisons également le symbole de fin de mot \$, mais celui-ci sert à indiquer la fin d'un mot et plus à égaler la longueur des deux mots. Plus précisement, dès qu'un symbole \$ est lu sur une bande, seule l'autre bande peut encore être parcourue. Lorsqu'un deuxième symbole \$ est lu sur cette autre bande, la machine s'arrête.

Passons maintenant à la définition formelle ; pour cela, nous utilisons le *shuffle*. Un *automate asynchrone à deux bandes* ou simplement un *automate asynchrone* sur un alphabet Σ est un automate fini déterministe partiel sur $\Sigma^{\$}$ pour lequel l'ensemble des états est partitionné en cinq sous-ensembles notés respectivement Q_L, Q_R, $Q_L^{\$}$, $Q_R^{\$}$ et

$Q^\$$. L'ensemble $Q^\$$ contient un seul état qui est l'unique état final $q^\$$ de l'automate ; il n'existe aucune transition partant de cet état. L'état initial se trouve dans $Q_L \cup Q_R$.

La partition de l'ensemble des états sert à définir sur quelle bande le symbole suivant doit être lu : les indices L et R des sous-ensembles d'états indiquent sur quelle bande la machine doit lire le symbole suivant[1]. Ainsi, si l'automate se trouve dans un état de $Q_L \cup Q_L^\$$, le symbole suivant doit être lu sur la bande gauche.

L'exposant \$ indique si un symbole de fin de mot a déja été lu sur l'une des deux bandes ou non.

Définissons maintenant les transitions de l'automate. Une transition partant d'un état de $Q_L \cup Q_R$ et dont le label est une lettre de Σ aboutit dans un état de $Q_L \cup Q_R$, de même qu'une transition partant d'un état de $Q_L^\$ \cup Q_R^\$$ et dont le label est une lettre de Σ aboutit dans un état de $Q_L^\$ \cup Q_R^\$$. D'autre part, une \$-transition partant d'un état de Q_L ou de Q_R aboutit respectivement dans un état de $Q_R^\$$ ou de $Q_L^\$$. Enfin, une \$-transition partant d'un état de $Q_L^\$ \cup Q_R^\$$ aboutit dans l'état final $q^\$$.

Explicitons quelque peu cette définition.

Initialement, l'automate se trouve dans un état de $Q_L \cup Q_R$, mais dès qu'un symbole de fin de mot est lu, par exemple sur la bande droite, la machine se trouve dans un état de $Q_L^\$$. Après cela, l'automate ne lit plus que sur la bande gauche jusqu'à lire un second symbole de fin de mot, ce qui le mène à l'état $q^\$$. La situation est bien entendu complètement analogue lorsque le premier symbole de fin de mot rencontré est sur la bande gauche.

Ainsi, tout mot accepté par l'automate contient exactement deux symboles de fin de mot \$, un pour chaque bande. Lorsque nous travaillerons avec plusieurs automates asynchrones, si le contexte n'est pas clair, nous noterons par exemple par $Q_L(\mathcal{A})$ l'ensemble des états de l'automate $\mathcal{A}$ qui se trouvent dans Q_L.

Au vu de cette définition, nous disons qu'un couple (w_L, w_R) de mots sur Σ est *accepté* par l'automate asynchrone $\mathcal{A}$ s'il existe un mot w appartenant au langage $w_L\$ \sqcup\!\sqcup\, w_R\$$ qui est accepté par l'automate. Remarquons simplement que s'il existe un tel mot, il est unique. En effet, l'automate étant déterministe, les transitions à effectuer pour lire un couple (et donc les passages d'une bande à l'autre) sont complétement déterminées.

Illustrons ceci par un exemple en considérant l'automate asynchrone sur l'alphabet $\Sigma = \{a, b\}$ représenté à la figure 3.1 ; cet automate accepte le langage $L = \{(a^{3n}, b^{2n}) \in \Sigma^* \times \Sigma^* \mid n \geq 1\}$. Le couple $(aaaaaa, bbbb)$ est accepté par cet automate. Il suffit alors de regarder la suite de transitions à effectuer dans l'automate pour trouver quel mot w du shuffle $aaaaa\$ \sqcup\!\sqcup\, bbbb\$$ est lu par l'automate.

Nous commençons donc dans l'état initial 1 et nous lisons a sur la bande gauche pour arriver dans l'état 2 ; nous obtenons donc $w(1) = a$. Nous nous trouvons donc sur la bande droite et nous lisons la lettre b pour arriver dans l'état 3 ; nous obtenons

[1] Le symbole L correspond à la bande gauche (left) et le symbole R à la bande droite (right).

$w(2) = ab$. Nous lisons ensuite deux fois la lettre a sur la bande gauche et nous arrivons dans l'état 5 ; nous obtenons $w(4) = abaa$. La tête de lecture se trouve à nouveau sur la bande droite et lit un b pour revenir dans l'état 1, avec $w(5) = abaab$. Mis à part les deux symboles de fin de mot \$, il ne reste plus qu'à lire le couple (aaa, bb) qui est exactement le couple que nous venons de lire. Nous parcourons donc à nouveau la boucle d'états $1-2-3-4-5$ pour obtenir $w(10) = abaababaab$. Nous terminons la lecture par les deux symboles de fin de mot \$. Le mot qui est lu par l'automate est donc le mot $w = abaababaab\$\$$.

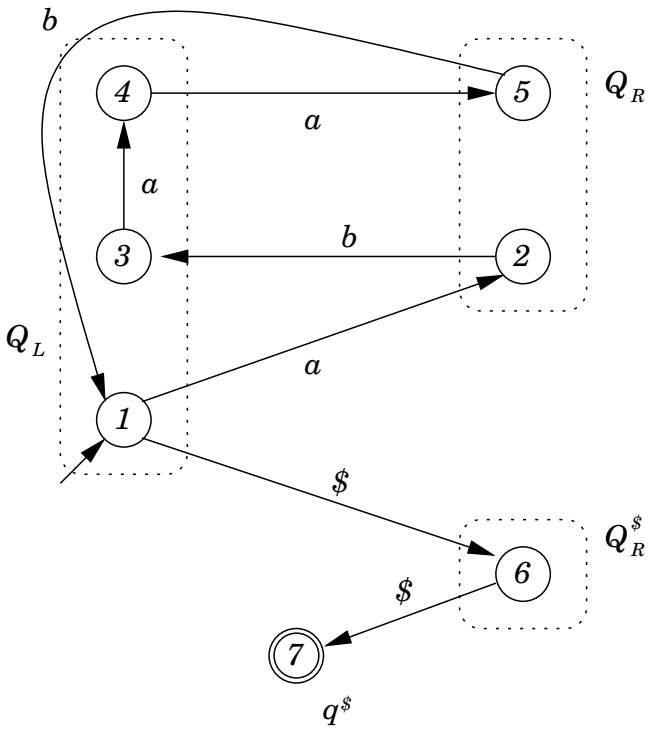

FIG. 3.1 – Automate asynchrone acceptant les couples de mots de la forme (a^{3n}, b^{2n}), $n \geq 1$.

Le lemme suivant décrit comment un automate asynchrone peut être vu comme un automate "ordinaire" sur un couple d'alphabets.

Lemme 3.1.1. *Soient Σ un alphabet et L un langage à deux variables sur (Σ, Σ). Une condition nécessaire et suffisante pour que le langage soit régulier (au sens d'un langage régulier à deux variables) est qu'il soit accepté par un automate asynchrone pour lequel toute transition partant d'un état de Q_L et dont le label est une lettre de Σ aboutit dans un état de Q_R et vice-versa.*

En d'autres termes, les deux bandes sont lues alternativement jusqu'à ce qu'un des deux mots soit entièrement lu.

Démonstration. C'est évident : un langage L à deux variables est régulier s'il existe un automate à deux variables[2] acceptant le langage $L^{\$}$. Il est alors très facile de transformer cet automate en automate asynchrone lisant les deux bandes alternativement. □

Les langages acceptés par automates asynchrones[3] ne jouissent pas de toutes les propriétés des langages réguliers à une variable.

Par exemple, l'ensemble des mots de la forme (a^n, a^{2n}) pour $n \geq 0$ est un langage à deux variables accepté par l'automate asynchrone représenté à la figure 3.2. De même, un automate analogue à ce dernier accepte le langage composé des mots de la forme (a^{2n}, a^n) pour $n \geq 0$. Par contre l'union de ces deux langages n'est pas acceptée par un automate asynchrone. En effet, si c'était le cas, l'automate en question devrait choisir quelle bande il lit le plus vite et serait donc non déterministe, ce qui contredit la définition d'un automate asynchrone.

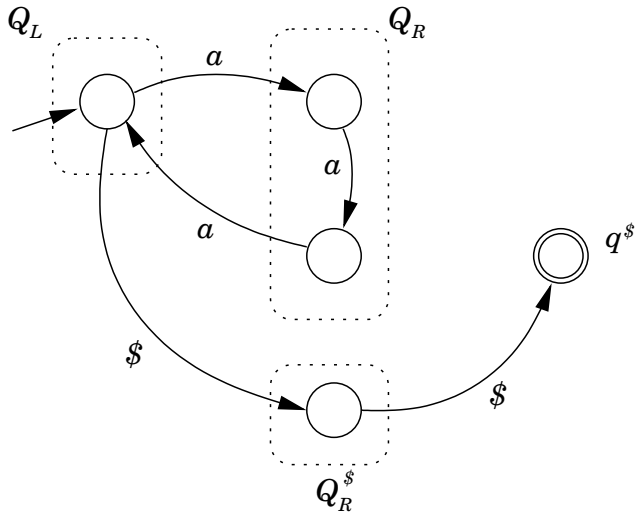

FIG. 3.2 – Automate asynchrone acceptant les couples de mots de la forme (a^n, a^{2n}).

Bien que toutes les propriétés des langages réguliers ne soient pas transposables à la classe des langages acceptés par automate asynchrone, le lemme suivant stipule que la stabilité pour le passage au complémentaire est toujours valable.

Lemme 3.1.2. *La classe des langages acceptables par automate asynchrone est stable pour le passage au complémentaire.*

Démonstration. Soit $\mathcal{A}$ un automate asynchrone qui accepte un langage $L \subset \Sigma^* \times \Sigma^*$. Construisons un automate asynchrone $\mathcal{B}$ qui accepte exactement le langage $(\Sigma^* \times \Sigma^*) \backslash L$.

Nous commençons par supprimer tous les arcs de transition aboutissant dans l'état $q^{\$}$ et nous ajoutons trois nouveaux états, à savoir $q_L \in Q_L$, $q_R^{\$} \in Q_R^{\$}$ et $q_L^{\$} \in Q_L^{\$}$.

[2] C'est-à-dire un automate lisant des couples de lettres à chaque transition.

[3] À ne pas confondre avec les langages réguliers à deux variables car les automate asynchrones qui les acceptent sont particuliers (cf. lemme 3.1.1).

Nous définissons maintenant les transitions de l'automate. Tout d'abord, chaque transition partant d'un des trois nouveaux états et dont le label est une lettre de Σ aboutit dans ce même état.

Nous rajoutons ensuite trois \$-transitions, l'une partant de q_L et aboutissant en $q_R^{\$}$, la deuxième partant de $q_R^{\$}$ et aboutissant en $q^{\$}$ et la dernière partant de $q_L^{\$}$ et aboutissant en $q^{\$}$.

Enfin, pour chaque état q non final, s'il existe un symbole σ de $\Sigma \cup \{\$\}$ qui n'est le label d'aucune transition partant de q dans $\mathcal{A}$, nous introduisons une nouvelle transition partant de q, de label σ et aboutissant dans un des états q_L, $q_R^{\$}$ ou $q_L^{\$}$. Le choix de l'état d'arrivée d'une telle transition est univoquement déterminé par la définition d'automate asynchrone et est illustré à la figure 3.3. Par exemple, si l'état q appartient à $Q_L^{\$}(\mathcal{A})$ et s'il n'existe pas de transition de label \$ partant de q, nous définissons une transition dans $\mathcal{B}$ partant de q, de label \$ et aboutissant en $q^{\$}$.

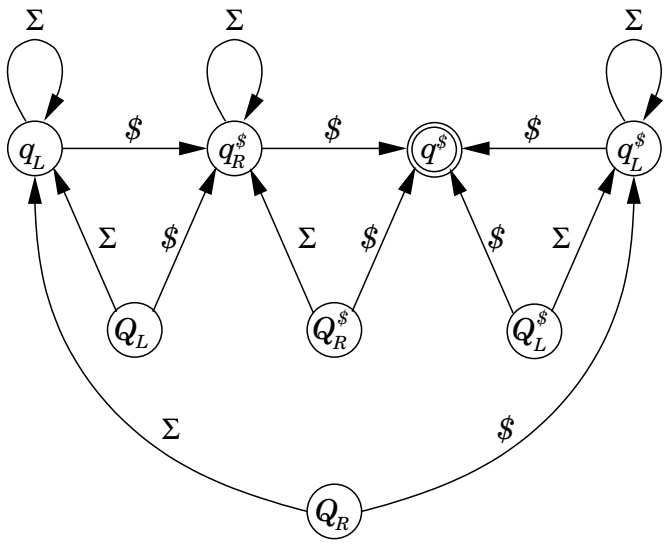

FIG. 3.3 – Automate acceptant le langage $(\Sigma^* \times \Sigma^*) \backslash L$.

Nous terminons la construction de l'automate $\mathcal{B}$ en éliminant tous les états inaccessibles et tous les états à partir desquels il est impossible d'atteindre l'état final $q^{\$}$, ainsi que toutes les transitions se rapportant à ces états.

Il est maintenant facile de voir qu'un couple (w_L, w_R) de mots sur Σ est accepté par $\mathcal{A}$ si et seulement s'il ne l'est pas par $\mathcal{B}$. □

Le lemme suivant fournit un moyen de passer d'un langage accepté par automate asynchrone à un langage régulier à une variable.

Lemme 3.1.3. *Les quantificateurs $\forall$ et $\exists$ convertissent un langage accepté par automate asynchrone en un langage régulier à une variable.*

Démonstration. Soit L un langage sur $\Sigma^* \times \Sigma^*$ accepté par un automate asynchrone $\mathcal{A}$. Vu le lemme précédent, il suffit de prouver le résultat pour l'opérateur $\exists$.

Pour cela, il suffit de remplacer, dans l'automate $\mathcal{A}$, chaque transition partant de $Q_L \cup Q_L^{\$}$ ainsi que chaque \$-transition par des ε-transitions. Nous obtenons alors un automate fini non déteministe qui accepte exactement le langage $\exists L$. □

Proposition 3.1.4. *Soit $\mathcal{A}$ un automate asynchrone sur l'alphabet Σ. Si w_L est un mot sur Σ, alors le langage*

$$\{w_R \in \Sigma^* \mid (w_L, w_R) \in L(\mathcal{A})\}$$

est régulier.

Le résultat est vrai aussi si L et R sont interchangés.

Démonstration. Nous construisons l'automate partiel $\mathcal{A}'$ sur $\Sigma^{\$}$ qui accepte le langage

$$\{w_L \mid (w_L, w_R) \in L\}.$$

Nous considérons pour cela l'ensemble Y des suffixes de $w_R\$$. L'idée est que l'automate $\mathcal{A}'$, en parcourant un mot w_L, ait constamment en mémoire quel élément de Y l'automate $\mathcal{A}$ devrait encore lire s'il lisait le couple (w_L, w_R).

Nous définissons donc l'ensemble des états de $\mathcal{A}'$ par l'ensemble $Q \times Y$, où

$$Q = Q_L(\mathcal{A}) \cup Q_L^{\$}(\mathcal{A}) \cup Q^{\$}(\mathcal{A}).$$

Nous définissons alors les transitions de l'automate $\mathcal{A}'$ de telle sorte que chaque état de $\mathcal{A}'$ corresponde à une situation dans laquelle $\mathcal{A}$ aurait encore à lire le suffixe y de $w_R\$$. Plus précisément, soit (q, y) un état de $\mathcal{A}'$. S'il existe une transition dans $\mathcal{A}$ partant de q, de label $\sigma \in \Sigma^{\$}$ et aboutissant dans un état q', alors nous définissons une transition dans $\mathcal{A}'$ partant de (q, y), de label σ et aboutissant dans l'état $(q.r, y')$, où r est le plus petit préfixe non trivial de $\sigma y = ry'$ tel que $q.r$ appartienne à Q. Si un tel préfixe r n'existe pas, alors la transition n'est pas définie.

L'unique état final de l'automate $\mathcal{A}'$ est le couple $(q^{\$}, \varepsilon)$.

Si q_0 est l'état initial de l'automate $\mathcal{A}$ et si r est le plus petit préfixe (éventuellement trivial) de $w_R\$ = ry'$ tel que $q_0.r$ appartienne à Q, alors l'état initial de $\mathcal{A}'$ est le couple $(q_0.r, y')$. S'il n'existe pas de tel préfixe r, alors l'automate partiel est vide et le langage correspondant est l'ensemble vide.

Nous terminons les constructions en éliminant de l'automate tous les états inaccessibles ainsi que tous ceux à partir desquels il est impossible d'atteindre un état final. □

3.2 Groupes automatiques asynchrones

Dans cette section, nous introduisons la notion de *groupe automatique asynchrone* à travers la définition de *structure automatique asynchrone*. Nous définissons également une certaine *standardisation* parmis les automates asynchrones qui nous permet d'établir certaines propriétés intéressantes.

Une *structure automatique asynchrone* sur un groupe G consiste en un ensemble Σ de générateurs de semigroupe pour G, un automate fini $\mathcal{A}$ sur Σ et des automates asynchrones $\mathcal{M}_\sigma$ pour tout symbole σ de $\Sigma \cup \{\varepsilon\}$ satisfaisant les propriétés suivantes.

1. L'application $\pi : L(\mathcal{A}) \to G$ est surjective, et
2. pour tout symbole σ de $\Sigma \cup \{\varepsilon\}$, un couple (w_L, w_R) de mots sur Σ est accepté par l'automate asynchrone $\mathcal{M}_\sigma$ si et seulement si chacun des deux mots est accepté par $\mathcal{A}$ et si nous avons $\overline{w_L\sigma} = \bar{w}_R$.

Nous disons que l'automate $\mathcal{A}$ est l'automate *accepteur* de la structure et que le langage qu'il accepte est le *langage des représentants*. L'automate $\mathcal{M}_\varepsilon$ est l'automate *égalité* et les automates $\mathcal{M}_a$, où a est une lettre de Σ, sont les automates *multiplicateurs*. Ces termes ont bien sûr la même signification que dans le cas synchrone et nous notons également la structure par (Σ, L).

Nous montrons maintenant que l'existence d'une structure automatique asynchrone permet de définir certains automates multiplicateurs dits *standards*, basés sur le langage des représentants L et sur un voisinage de l'identité du groupe G.

Supposons que $\mathcal{A}$ est un automate asynchrone sur l'alphabet Σ et que le couple (w_L, w_R) de mots sur Σ est accepté par $\mathcal{A}$. Nous imaginons que l'automate $\mathcal{A}$ commence à lire le couple de mots au temps 0 et qu'il lit une lettre par unité de temps. Nous définissons alors les fonctions temporelles t_L et t_R en notant par $w_L(t_L(t))$ (resp. par $w_R(t_R(t))$) le préfixe de w_L (resp. de w_R) lu après un temps t. En d'autres mots, au temps t, l'automate a déjà lu le préfixe de longueur $t_L(t)$ de w_L et le préfixe de longueur $t_R(t)$ de w_R.

Lemme 3.2.1 (Fonction de différence de mots)**.** *Soit Σ un ensemble de générateurs de semigroupe pour un groupe G. Supposons que g est un élément de G et que $\mathcal{M}_g$ est un automate asynchrone sur l'alphabet Σ tel que tout couple (w_L, w_R) de mots sur Σ accepté par cet automate satisfait $\bar{w}_L g = \bar{w}_R$. Alors il existe une fonction f définie sur l'ensemble Q des états de $\mathcal{M}_g$ et à valeurs dans G telle que pour tout couple (w_L, w_R) de mots accepté par $\mathcal{M}_g$,*

$$f(q) = \overline{w_L(t_L)}^{-1}\overline{w_R(t_R)},$$

où q est l'état de $\mathcal{M}_g$ atteint en ayant lu $(w_L(t_L), w_R(t_r))$.

En d'autres termes, les états de $\mathcal{M}_g$ enregistrent, entre autres choses, la *différence de mots* entre les images des préfixes des mots w_L et w_R, au fur et à mesure que ceux-ci sont lus. De plus, ce résultat implique que pour un automate multiplicateur donné,

il n'existe qu'un nombre fini de différences de mots entre les deux mots d'un couple accepté.

Démonstration. Soit q un état de $\mathcal{M}_g$. Puisque, par définition, les états inutiles[4] des automates asynchrones ont été supprimés, il existe un chemin de longueur minimale partant de l'état q et aboutissant dans l'état final. La longueur de ce chemin (en terme de nombre d'arcs) est évidemment majorée par le nombre d'états de l'automate.

Supposons que l'exécution de ce chemin soit un mot appartenant au shuffle de deux mots u_q et v_q sur Σ. Nous définissons sur Q la fonction f à valeurs dans G par

$$f(q) = \bar{u}_q g \bar{v}_q^{-1},\ q \in Q.$$

Supposons maintenant qu'au temps t, l'automate a déjà lu les préfixes respectifs $w_L(t_L(t))$ et $w_R(t_R(t))$ de w_L et de w_R et qu'il se trouve dans l'état q. Cela signifie que le couple $(w_L(t_L(t))u_q, w_R(t_R(t))v_q)$ de mots sur Σ est accepté par l'automate $\mathcal{M}_g$ et vérifie donc

$$\overline{w_L(t_L(t))u_q} g = \overline{w_R(t_R(t))v_q},$$

ou encore

$$\bar{u}_q g \bar{v}_q^{-1} = \overline{w_L(t_L(t))}^{-1} \overline{w_R(t_R(t))},$$

d'où la conclusion. □

Il est possible, dans une structure automatique asynchrone, que les automates multiplicateurs présentent un grand favoritisme entre les deux bandes en ce sens qu'ils lisent le mot d'une bande beaucoup plus vite que le mot de l'autre.

Par exemple, pour un groupe fini G, si l'ensemble des générateurs est G lui-même et si le langage des représentants choisi est simplement G^*, i.e. l'ensemble de tous les mots sur G, alors nous pouvons choisir les automates multiplicateurs pour qu'ils parcourent d'abord complétement la bande gauche pour ne passer qu'ensuite à la bande droite, tout en gardant en mémoire l'évolution de $\overline{w_L(t_L)}^{-1}\overline{w_R(t_R)}$. Ce choix est en effet possible car, le groupe étant fini, les deux mots d'un couple accepté n'admettent qu'un nombre fini de différences de mots.

Notre première étape pour trouver des automates multiplicateurs standards pour une structure automatique asynchrone donnée est de modifer les automates de la structure afin qu'un tel cas de figure ne puisse apparaître.

Ainsi, nous disons qu'un automate asynchrone est *asynchrone à lecture bornée* s'il existe un entier k tel que l'automate ne lit jamais plus de k symboles à la suite sur une même bande. L'entier k est appelé le *facteur asynchrone* de l'automate. Nous parlons également de *structure automatique asynchrone bornée* lorsque tous les automates multiplicateurs de la structure sont asynchrones à lecture bornée.

[4] Il s'agit des états inaccessibles, ainsi que des états à partir desquels il est impossible d'atteindre l'état final.

En particulier, un automate asynchrone à lecture bornée ne peut lire un mot sur une bande que k fois plus rapidement que le mot sur l'autre bande.

Le théorème suivant stipule qu'une structure automatique asynchrone sur un groupe peut toujours être supposée bornée.

Théorème 3.2.2 (Théorème d'asynchronisation bornée)**.** *Si G est un groupe admettant une structure automatique asynchrone (Σ, L), alors le groupe G admet une structure automatique asynchrone bornée (Σ, L') où L' est un sous-ensemble du langage L.*

Démonstration. Notons par c le nombre maximum d'états parmi les automates de la structure.

Nous commençons par modifier l'automate accepteur relatif à la structure (Σ, L) en un automate $\mathcal{A}'$ qui garde constamment en mémoire les c^2 dernières transitions qu'il a effectuées.

Pour cela, nous montrons comment, pour un automate fini déterministe $\mathcal{B}$ et un entier positif k, nous pouvons former un automate *k-historique*[5] $\mathcal{B}'$ sur le même alphabet et qui accepte le même langage.

Les états de l'automate $\mathcal{B}'$ sont des suites de transitions de la forme

$$(q_1, a_1, q_2, \ldots, q_n, a_n, q_{n+1}),$$

où $n \leq k$ et où, pour tout $1 \leq i \leq n$, le triplet (q_i, a_i, a_{i+1}) est une transition de l'automate $\mathcal{B}$. Nous exigeons que la longueur de la suite de transitions définissant un état soit maximale en ce sens que si au moins k transitions ont déjà été effectuées, alors la longueur n de la suite est k. Le cas $n < k$ ne peut donc apparaître que durant les $k-1$ premières transitions[6]. Les états finals de $\mathcal{B}'$ sont ceux pour lesquels q_{n+1} est un état final de $\mathcal{B}$. Enfin, l'état initial est la suite (q_1) de longueur 0, où q_1 est l'état initial de $\mathcal{B}$.

Définissons maintenant les transitions de l'automate $\mathcal{B}'$. Si l'automate $\mathcal{B}'$ se trouve dans l'état

$$(q_1, a_1, q_2, \ldots, q_n, a_n, q_{n+1}),$$

alors pour toute transition de $\mathcal{B}$ partant de l'état q_{n+1}, de label a et aboutissant dans un état q, il existe une transition dans $\mathcal{B}'$ de label a, partant de l'état

$$(q_1, a_1, q_2, \ldots, q_n, a_n, q_{n+1})$$

et aboutissant dans l'état

$$(q_1, a_1, q_2, \ldots, q_n, a_n, q_{n+1}, a, q)$$

si $n < k$, ou dans l'état

$$(q_2, a_2, q_3, \ldots, q_n, a_n, q_{n+1}, a, q)$$

[5] C'est-à-dire un automate qui garde en mémoire les k dernières transitions qu'il a effectuées.

[6] L'automate ne peut évidemment pas enregistrer des transitions qui n'ont pas encore été effectuées.

si $n = k$.

Illustrons tout ceci sur un exemple. Supposons que l'automate fini déterministe $\mathcal{B}$ a déjà effectué les transitions

$$(q_1, a_1, q_2), (q_2, a_2, q_3), \ldots, (q_{k+2}, a_{k+2}, q_{k+3})$$

sur une entrée w, où q_1 est l'état initial de $\mathcal{B}$. Alors, vu les constructions faites ci-dessus, l'automate $\mathcal{B}'$ passe successivement par les états

$$(q_1), (q_1, a_1, q_2), (q_1, a_1, q_2, a_2, q_3), \ldots, (q_1, a_1, q_2, \ldots, q_k, a_k, a_{k+1}),$$
$$(q_2, a_2, q_3, \ldots, q_{k+1}, a_{k+1}, q_{k+2}), (q_3, a_3, q_4, \ldots, q_{k+1}, a_{k+1}, q_{k+2}, a_{k+2}, q_{k+3}).$$

Vu la manière dont nous avons défini l'automate $\mathcal{B}'$, il est clair que ce dernier accepte exactement le même langage que $\mathcal{B}$.

Nous pouvons maintenant construire l'automate c^2-historique associé à l'automate $\mathcal{A}$. Dans cet automate, nous supprimons tous les états qui contiennent un chemin fermé trivial de l'automate $\mathcal{A}$, i.e. nous supprimons tous les états de la forme

$$(\ldots, q_i, a_i, \ldots, a_{j-1}, q_j = q_i, \ldots),$$

où le mot $a_i a_{i+1} \ldots a_{j-1}$ est un représentant de l'identité du groupe[7] G.

Nous notons ensuite par $\mathcal{A}'$ l'automate obtenu en ayant supprimé, non seulement tous ces états, mais aussi tous les états rendus alors inaccessibles, ainsi que toutes les transitions s'y rapportant. Nous construisons également l'automate $\mathcal{A}''$ sur l'alphabet $\Sigma^{\$}$ qui accepte exactement le langage $L(\mathcal{A}')\$$ et que nous supposons normalisé de telle sorte qu'il ne possède qu'un unique état final $q^{\$}$. Dans la définition ci-après, nous faisons référence aux automates $\mathcal{A}'$ et $\mathcal{A}''$ lorsque nous parlons des *automates auxiliaires associés à* $\mathcal{A}$.

Remarquons que l'automate $\mathcal{A}'$ n'accepte pas nécessairement tous les mots acceptés par $\mathcal{A}$. En effet, vu sa construction, les mots contenant un facteur de longueur inférieure à c^2 représentant un chemin fermé dans $\mathcal{A}$ et dont l'image dans G est l'identité ne sont plus acceptés. Cependant, le langage accepté par $\mathcal{A}'$ est toujours envoyé surjectivement sur le groupe G : considérons un élément g de G et notons par w un mot sur Σ de longueur minimale qui est accepté par l'automate $\mathcal{A}$ et qui est envoyé sur g. Si en lisant le mot w, l'automate c^2-historique associé à $\mathcal{A}$ passe par un état que nous avons supprimé pour construire $\mathcal{A}'$, alors le mot w contient un facteur de longueur inférieure à c^2 qui est envoyé sur l'identité du groupe et qui décrit un chemin fermé dans le graphe associé à l'automate $\mathcal{A}$. Ainsi, le mot w' obtenu en supprimant tous les facteurs de w de cette forme est accepté par $\mathcal{A}'$ et est également envoyé sur g.

[7] Pour vérifier que ce chemin satisfait cette condition, nous utilisons l'automate égalité en comparant un mot accepté par $\mathcal{A}$ contenant le facteur correspondant à l'exécution du chemin avec ce même mot auquel nous avons retiré ce facteur.

Nous construisons maintenant les nouveaux automates multiplicateurs à lecture bornée $\mathcal{M}'_\sigma$, $\sigma \in \Sigma \cup \{\varepsilon\}$, qui acceptent un couple (w_L, w_R) de mots sur Σ si et seulement si ce couple est accepté par l'automate multiplicateur $\mathcal{M}_\sigma$ de la structure (Σ, L) et si chacun des mots w_L et w_R est accepté par l'automate $\mathcal{A}'$.

Nous réalisons cela en considérant comme ensemble d'états de l'automate, l'ensemble $Q_\sigma \times Q'' \times Q''$, où Q_σ est l'ensemble des états de l'automate $\mathcal{M}_\sigma$ et où Q'' est l'ensemble des états de l'automate $\mathcal{A}''$. De cette façon, l'automate $\mathcal{M}'_\sigma$ peut suivre l'évolution des états dans l'automate $\mathcal{A}''$ au fur et à mesure que les mots $w_L\$$ et $w_R\$$ sont lus. La construction qui est faite est la même que celle décrite pour effectuer l'intersection de deux automates juste après le théorème 1.1.1.

Au vu de toutes les constructions qui précèdent, l'automate $\mathcal{A}'$ et les automates multiplicateurs $\mathcal{M}'_\sigma$, avec σ appartenant à $\Sigma \cup \{\varepsilon\}$, forment une structure automatique asynchrone. Montrons alors que cette structure est bornée de facteur asynchrone $c^2 - 1$.

Considérons un couple (w_L, w_R) de mots sur Σ accepté par $\mathcal{M}'_\sigma$ pour un élément σ de $\Sigma \cup \{\varepsilon\}$ et supposons que sur cette entrée, l'automate $\mathcal{M}'_\sigma$ lise consécutivement c^2 symboles[8] de w_L ; nous supposons qu'il commence à lire ces c^2 symboles au temps T. Comme les automates $\mathcal{A}$ et $\mathcal{M}_\sigma$ ont tous deux au plus c états, il existe $T \leq t < t+p \leq T + c^2$ tels que l'automate $\mathcal{A}$, ainsi que l'automate $\mathcal{M}_\sigma$, se trouvent dans le même état aux temps[9] t et $t+p$. Nous obtenons donc $t_R(t+p) = t_R(t)$ et $t_L(t+p) = t_L(t) + p$. Si nous appliquons le lemme 3.2.1 (Fonction de différence de mots) à l'automate $\mathcal{M}'_\sigma$, nous obtenons

$$\begin{aligned} \overline{w_L(t_L(t))}^{-1}\overline{w_R(t_R(t))} &= \overline{w_L(t_L(t+p))}^{-1}\overline{w_R(t_R(t+p))} \\ &= \overline{w_L(t_L(t)+p)}^{-1}\overline{w_R(t_R(t))}. \end{aligned}$$

Il s'ensuit que $\overline{w_L(t_L(t))} = \overline{w_L(t_L(t+p))} = \overline{w_L(t_L(t)+p)}$, et le mot w_L contient donc un facteur de longueur inférieure à c^2 envoyé sur l'identité du groupe et décrivant une boucle dans l'automate $\mathcal{A}$, ce qui contredit le fait que le mot w_L est accepté par l'automate $\mathcal{A}'$. □

Nous pouvons maintenant passer à la définition d'*automate standard*.

Soit Σ un ensemble de générateurs de semigroupe pour un groupe G et soit $\mathcal{A}$ un automate fini sur Σ tel que l'application $\pi : L(\mathcal{A}) \to G$ est surjective. Nous considérons un élément g de G ainsi que trois entiers positifs $R_1 < R_2 < R_3$ et nous définissons *l'automate multiplicateur standard* $\mathcal{M}_g$ basé sur le quadruplet $(\mathcal{A}, R_1, R_2, R_3)$; il s'agit d'un automate asynchrone sur l'alphabet Σ.

L'idée est que l'automate $\mathcal{M}_g$ doit accepter la multiplication par g des éléments de G représentés par des mots de $L(\mathcal{A})$, mais cela n'est possible que lorsque les entiers

[8] Nous obtiendrons le développement analogue si les c^2 symboles sont lus sur la bande droite.

[9] En effet, les deux automates ayant chacun au plus c états, leur intersection (comme décrite après le théorème 1.1.1) en a au plus c^2. Or comme nous parcourons exactement c^2 symboles, nous passons par $c^2 + 1$ états. Nous passons donc au moins deux fois par un même état.

R_1, R_2 et R_3 sont choisis de façon appropriée, comme stipulé dans le théorème 3.2.4 (Théorème des automates multiplicateurs standards).

Considérons les *automates auxiliaires* $\mathcal{A}'$ et $\mathcal{A}''$ associés à l'automate $\mathcal{A}$ comme introduits dans la démonstration du théorème précédent. Nous notons par Q'' l'ensemble des états de $\mathcal{A}''$ et par B la boule ouverte du graphe de Cayley de G centrée en l'identité et de rayon R_3. Comme nous l'avons déjà défini dans le cas synchrone, il s'agit de l'ensemble des éléments de G qui se trouvent à une distance de l'identité du groupe strictement inférieure à R_3 dans le graphe de Cayley, i.e.

$$B = \{g \in G \mid d_G(g, 1) < R_3\}.$$

Nous définissons l'ensemble des états de $\mathcal{M}_g$ par l'ensemble des quintuples de la forme

$$(h, q_L, q_R, d, T) \in \{L, R\} \times Q'' \times Q'' \times B \times \{A, B\}.$$

Tout d'abord, la composante h indique sur quelle bande le prochain symbole doit être lu. Ensuite, les deux composantes q_L et q_R nous informent dans quel état se trouve l'automate $\mathcal{A}''$ après avoir lu $w_L(t_L)$ et $w_R(t_R)$ respectivement. Elles nous indiquent également si, oui ou non, un symbole de fin de mot a été lu sur l'une des deux bandes. La quatrième composante d enregistre quant à elle la différence de mots $\overline{w_L(t_L)}^{-1}\overline{w_R(t_R)}$. Enfin, la dernière composante T indique si la bande parcourue est *devant* (A) ou *derrière*[10] (B) dans un sens que nous précisons dans la suite.

Nous partitionnons alors l'ensemble des états en les cinq sous-ensembles Q_L, Q_R, $Q_L^{\$}$, $Q_R^{\$}$ et $Q^{\$}$ en fonction des valeurs de h, de q_L et de q_R. Nous éliminons ensuite tous les états de la forme $(h, q^{\$}, q^{\$}, d, T)$, où $q^{\$}$ est l'état final de l'automate $\mathcal{A}''$, excepté les deux états $(L, q^{\$}, q^{\$}, g, A)$ et $(R, q^{\$}, q^{\$}, g, A)$ que nous amalgamons en un unique état que nous choisissons comme unique état final. Remarquons que cette "fusion" est légitime car, comme nous allons le voir, nous nous basons sur les transitions de l'automate $\mathcal{A}''$ pour définir celles de $\mathcal{M}_g$ et il n'existe aucune transition partant de $q^{\$}$ dans $\mathcal{A}''$. Pour finir, nous choisissons comme état initial le quintuple $(L, q_0, q_0, 1, A)$, où 1 est l'identité du groupe G et où q_0 est l'état initial de l'automate $\mathcal{A}''$. En particulier, nous pouvons remarquer que l'automate $\mathcal{M}_g$ commence à lire le couple de mots par celui de la bande gauche.

Définissons maintenant les transitions de l'automate $\mathcal{M}_g$. Comme nous l'avons annoncé précédemment, nous nous basons sur les transitions existantes dans l'automate $\mathcal{A}''$.

Considérons un symbole σ de $\Sigma^{\$}$.

Nous commençons par définir l'action des transitions sur les composantes q_L et q_R. Une transition de label σ et partant d'un état où la première composante est L (resp. R) laisse q_R (resp. q_L) inchangé et agit sur q_L (resp. sur q_R) comme le fait l'automate $\mathcal{A}''$. Ainsi, s'il n'existe pas de transition de label σ partant d'un état q_L de $\mathcal{A}''$, alors

[10] Dans la littérature anglaise, nous rencontrons les termes *ahead* (A) et *behind* (B).

il n'existe pas non plus de transition dans $\mathcal{M}_g$ de label σ et partant des états de la forme[11] (L, q_L, q_R, d, T).

L'action des transitions sur la composante d est également facile à définir : une transition de label σ et partant d'un état dont la première composante est L (resp. R) modifie d en $d' = \bar{\sigma}^{-1}d$ (resp. en $d' = d\bar{\sigma}$). Cette définition a bien un sens puisque nous obtenons

$$\begin{aligned} d' &= \bar{\sigma}^{-1}d \\ &= \bar{\sigma}^{-1}\overline{w_L(t_L(t))}^{-1}\overline{w_R(t_R(t))} \\ &= \overline{w_L(t_L(t+1))}^{-1}\overline{w_R(t_R(t+1))} \end{aligned}$$

si $h = L$ et

$$\begin{aligned} d' &= d\bar{\sigma} \\ &= \overline{w_L(t_L(t))}^{-1}\overline{w_R(t_R(t))}\bar{\sigma} \\ &= \overline{w_L(t_L(t+1))}^{-1}\overline{w_R(t_R(t+1))} \end{aligned}$$

si $h = R$.

Remarquons que pour $\sigma = \$$, nous avons $\bar{\sigma} = \bar{\sigma}^{-1} = 1$, où 1 est le neutre du groupe G ; une \$-transition laisse donc d inchangé. Nous notons par r la distance entre g et l'identité du groupe dans le graphe de Cayley. Si r devient supérieur ou égal à R_3, i.e. si une transition implique que l'élément d sort de la boule B, alors cette transition n'est pas définie dans $\mathcal{M}_g$.

En ce qui concerne les deux composantes h et T, elles restent inchangées quelle que soit la transition qui est exécutée, excepté dans les trois cas particuliers suivants :

(i) si, après avoir exécuté une transition, nous obtenons $r = R_1$, alors la composante h reste inchangée et la composante T devient A, quelle qu'était sa valeur avant la transition, i.e. le mot couramment lu passe *devant* ;

(ii) si, après la transition, nous obtenons $r = R_2$ et si $T = A$, alors T devient B et nous changeons également h, sauf si $q_L = q^{\$}$ ou $q_R = q^{\$}$, i.e. la tête de lecture change de bande et puisque le mot qui était lu était *devant*, celui que nous commençons à lire est *derrière* ;

(iii) si, après la transition, nous obtenons $q_L = q^{\$}$ (resp. $q_R = q^{\$}$), alors h devient R (resp. L).

La figure suivante est une partie du graphe de Cayley d'un groupe G et représente le comportement des automates multiplicateurs standards. Le point S est l'identité du groupe. Les petits cercles ont un rayon R_1 et les grands ont un rayon R_2. Nous

[11] Il peut par contre en exister partant d'états de la forme (R, q_L, q_R, d, T).

commençons par lire le mot de la bande gauche, avec le mot de celle de droite arrêté en B et le mot de gauche *devant*. Lorsque nous atteignons le point A, nous arrêtons de lire le mot de gauche et nous lisons le mot de droite. Puisque le mot de droite est *derrière*, rien ne change au point X bien que la distance entre A et X ait atteint R_2 à nouveau. Lorsque nous atteignons le point C, le mot de droite passe *devant* et nous continuons à lire sur celle-ci. Aux points D ou E, nous arrêtons de le lire et nous recommençons à lire celui de gauche qui est alors *derrière*. Dans la figure de droite, le mot de gauche passe *devant* au point G et nous continuons à le lire. Dans la figure de droite, la distance entre les points E et F atteint R_3 et le mot de gauche est *derrière*. Nous ne pouvons plus continuer et le processus échoue.

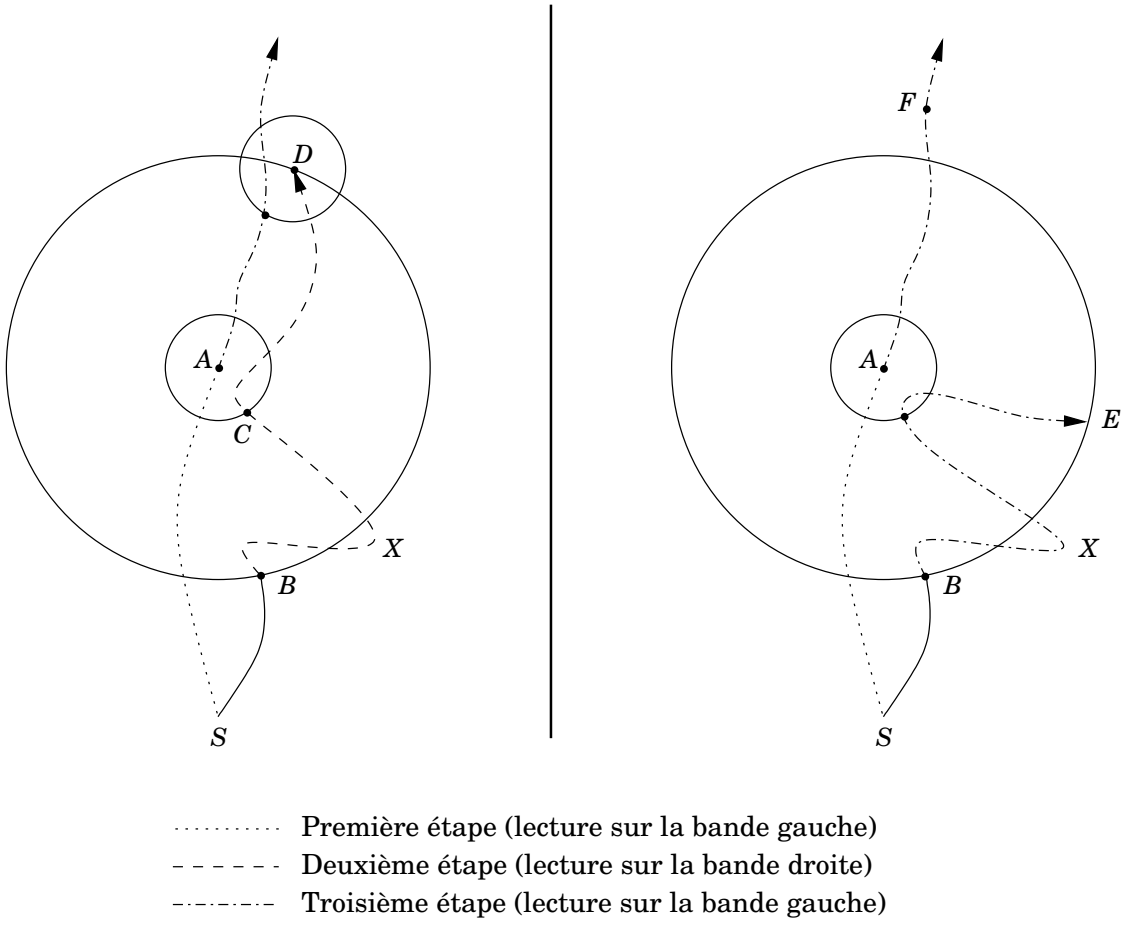

FIG. 3.4 – Comportement des automates multiplicateurs standards d'une structure automatique asynchrone.

Il découle de la définition que si un couple (w_L, w_R) de mots sur Σ est accepté par $\mathcal{M}_g$, alors les mots w_L et w_R sont acceptés par $\mathcal{A}'$ et $\overline{w_L g} = \overline{w_R}$. Remarquons que la réciproque n'est pas nécessairement vraie, mais l'unique cas pour lequel la réciproque est fausse est celui où la différence de mots g entre w_L et w_R sort de la boule B, i.e.

la distance r atteint R_3. Le théorème 3.2.4 (Théorème des automates multiplicateurs standards) exprime des conditions pour que cela ne puisse arriver.

Il est important de signaler que nous avons défini les automates multiplicateurs standards $\mathcal{M}_g$ pour tout élément g de G et non uniquement pour les éléments $a \in \Sigma$. Cela est dû au fait que, contrairement au cas synchrone, nous ne pouvons obtenir un automate multiplicateur $\mathcal{M}_{ab}$ à partir des automates $\mathcal{M}_a$ et $\mathcal{M}_b$, avec a et $b \in \Sigma$. En effet, l'ensemble des langages acceptés par automates asynchrones[12] n'est pas fermé pour la composition[13].

En effet, consiérons un alphabet à trois lettres $\Sigma = \{a, 2, 3\}$ et les deux langages L_1 et L_2 sur Σ définis par

$$L_1 = \{(a^n, 2a^{2n}) \mid n \geq 0\} \cup \{(a^n, 3a^{3n}) \mid n \geq 0\}$$

et

$$L_2 = \{(xa^j, a^j) \mid x = 2 \text{ ou } x = 3,\ j \geq 0\}.$$

Il est clair que ces deux langages sont acceptés par les automates représentés aux figures 3.5 et 3.6.

La composition des langages L_1 et L_2 est le langage

$$L_1 \circ L_2 = \{(u, v) \in \Sigma^* \times \Sigma^* \mid \exists w \in \Sigma^* : (u, w) \in L_1 \text{ et } (w, v) \in L_2\}.$$

Or, un couple (u, w) de mots sur Σ appartient à L_1 si et seulement s'il s'écrit sous la forme

$$(a^n, 2a^{2n}) \quad \text{ou} \quad (a^n, 3a^{3n}),$$

avec $n \geq 0$.

De plus, un couple (w, v) de mots sur Σ appartient à L_2 si et seulement s'il s'écrit sous la forme

$$(2a^j, a^j) \quad \text{ou} \quad (3a^j, a^j),$$

avec $j \geq 0$.

Il est alors clair que le langage $L_1 \circ L_2$ est le langage

$$L_1 \circ L_2 = \{(a^n, a^{2n} \mid n \geq 0\} \cup \{(a^n, a^{3n} \mid n \geq 0\},$$

qui n'est accepté par aucun automate asynchrone[14].

Le concept suivant nous aide à démontrer les deux résultats importants de cette section que sont le théorème 3.2.3 (Caractérisation des structures automatiques asynchrones bonées) et le théorème 3.2.4 (Théorème des automates multiplicateurs standards).

[12] Encore une fois, à ne pas confondre avec les langages réguliers à deux variables.

[13] Un langage accepté par automate asynchrone étant une partie de $\Sigma^* \times \Sigma^*$, la composition de deux langages est définie comme la composition de relations que nous avons introduit dans la section 1.3.

[14] L'automate asynchrone qui l'accepterait devrait choisir s'il lit la bande droite deux fois ou trois fois plus vite que la bande gauche et serait donc non déterministe, ce qui contredit la définition d'un automate asynchrone.

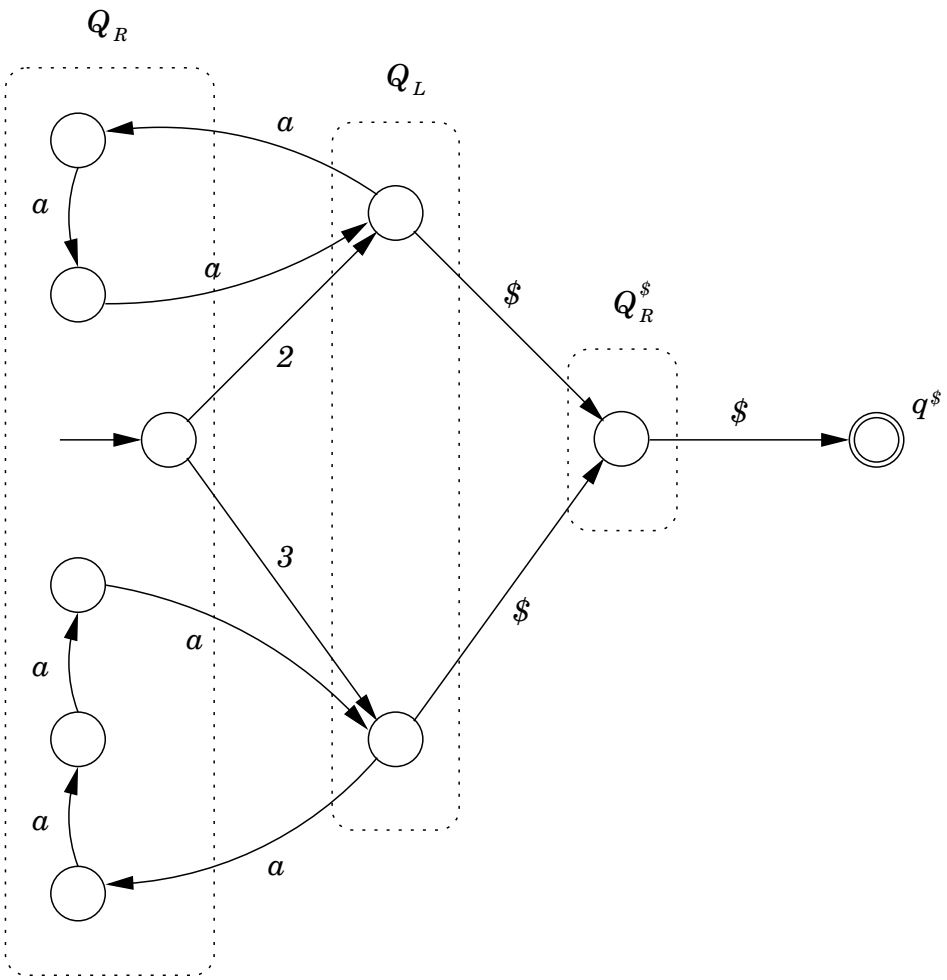

FIG. 3.5 – Automate asynchrone acceptant le langage L_1.

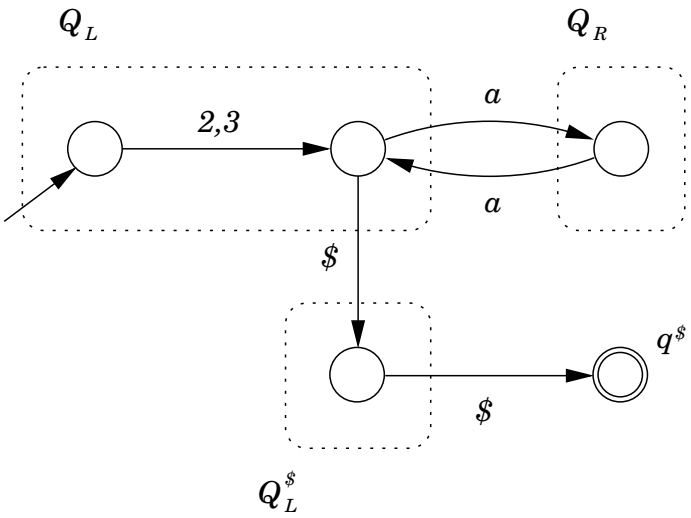

FIG. 3.6 – Automate asynchrone acceptant le langage L_2.

Soit G un groupe et soit Σ un ensemble de générateurs de semigroupe pour G. Notons Γ le graphe de Cayley de G relatif à Σ et L un langage régulier sur Σ envoyé de manière surjective sur G. Une *fonction de départ* pour (G, L), ou pour G si le contexte est clair, est une fonction $D : \mathbb{R} \to \mathbb{R}$ telle que si $w \in L$, $r, s \geq 0$, $t \geq D(r)$ et $s+t \leq |w|$, alors $d_G(\overline{w(s)}, \overline{w(s+t)}) > r$.

S'il existe une telle fonction pour un groupe G et un langage L, alors un mot de L peut éventuellement sortir de tout voisinage fini de l'identité dans G. Remarquons que nous aurions pu définir les fonctions de départ sur les entiers positifs uniquement.

Théorème 3.2.3 (Caractérisation des structures automatiques asynchrones bornées)**.** *Soient G un groupe, Σ un ensemble de générateurs de semigroupe de G et L un langage régulier sur Σ qui est envoyé de manière surjective sur G. Le couple (Σ, L) est une structure automatique asynchrone bornée sur G si et seulement si les deux conditions suivantes sont satisfaites :*

(i) il existe une fonction de départ D sur G ;

(ii) il existe $K > 0$ tel que pour tous w,$w' \in L$ tels que $d_G(\overline{w}, \overline{w}') \leq 1$, les chemins $\hat{w}$ et $\hat{w}'$ sont à une distance de Hausdorff d'au plus K l'un de l'autre.

Dire que les chemins $\hat{w}$ et $\hat{w}'$ sont à une distance de Hausdorff d'au plus K l'un et l'autre signifie que, pour tout s positif, il existe $s' \geq 0$ tel que la distance entre $\overline{w(s)}$ et $\overline{w'(s')}$ est au plus K, et qu'il en est de même si l'on intervertit w et w'. La constante K est appelée la *constante lipschitzienne asynchrone.*

Démonstration. Soit G un groupe muni d'une structure automatique asynchrone à lecture bornée (Σ, L) avec un facteur asynchrone k. Commençons par montrer l'existence de la constante lipschitzienne K.

Soient w_L, w_R deux mots de L satisfaisant $\overline{w_L a} = \overline{w_R}$ pour une lettre a de Σ. Le couple (w_L, w_R) de mots est donc accepté par l'automate multiplicateur $\mathcal{M}_a$. Vu le lemme 3.2.1 (Fonction de différence de mots), l'ensemble des différences de mots $\overline{w_L(t_L(t))}^{-1}\overline{w_R(t_R(t))}$ est fini[15]. De plus, l'alphabet Σ étant fini, il n'existe qu'un nombre fini[16] de différences de mots, pour tout couple (w_L, w_R) de mots qui sont à une distance au plus 1 l'un de l'autre. Si nous notons ces différences de mots par $g_1, \cdots, g_n$, avec $g_i \in G$ pour tout i, il suffit de prendre K supérieur à la plus grande distance entre l'un de ces g_i et l'identité, i.e.

$$K \geq sup\{d_G(g_i, 1) \mid 1 \leq i \leq n\},$$

et la condition (ii) est satisfaite.

Montrons maintenant qu'il existe une fonction de départ D pour G. Soit $\mathcal{A}$ l'automate accepteur de la structure et soit r un entier positif. Pour tout élément g du groupe tel que $d(1, g) \leq r$, et pour tout couple (q_1, q_2) d'états de $\mathcal{A}$, nous considérons un chemin (s'il existe) de longueur minimale partant de q_1 et aboutissant en q_2 et représentant g dans G. Nous notons également par f la fonction qui à r associe la longueur $f(r)$ du plus grand chemin ainsi défini. Montrons alors que la fonction

$$D : r \mapsto k(f(r) + 2c) + 1,$$

où c est le nombre d'états de l'automate $\mathcal{A}$, est bien une fonction de départ.

Nous procédons par contraposition.

Si le mot v est un facteur d'un mot accepté[17], il décrit un chemin dans $\mathcal{A}$ entre deux états q_1 et q_2. Soit u_1 un mot sur Σ de longueur minimale décrivant un chemin partant de q_0 et aboutissant en q_1 et soit u_2 un mot de longueur minimale décrivant un chemin partant de q_2 et aboutissant dans un état final. Les longueurs respectives de u_1 et de u_2 sont majorées par le nombre c d'états de $\mathcal{A}$. Soit enfin v' un mot de longueur minimale décrivant un chemin partant de q_1, aboutissant en q_2 et représentant le même élément que v.

Etablissons maintenant la correspondance entre les constructions qui viennent d'être faites et la définition. Nous prenons $s = |u_1|$ et $t = |v|$. Supposons que $d(u_1, u_1 v) \leq r$, i.e.

$$d(1, \overline{v}) \leq r.$$

Vu ce qui précède, nous avons $|v'| \leq f(r)$. De plus, comme le couple $(u_1 v u_2, u_1 v' u_2)$ est accepté par l'automate égalité $\mathcal{M}_\varepsilon$, nous obtenons

$$|u_1| + |v| + |u_2| \leq k(|u_1| + |v'| + |u_2|)$$

ce qui implique $|v| \leq k(f(r) + 2c)$, d'où la conclusion. □

[15] Il en existe exactement $\#Q$, où Q est l'ensemble des états de l'automate $\mathcal{M}_a$.

[16] Pour chaque lettre de Σ, il en existe exactement $\#Q$. Au total, il en existe donc au plus $\#\Sigma.\#Q$.

[17] Si dans la définition d'une fonction de départ, le mot w dont il est question est le mot $a_1 a_2 \cdots a_n$, où les a_i sont des lettres de Σ, alors le mot v choisi ici est le mot $a_{s+1} \cdots a_{s+t}$, avec $0 \leq s \leq s + t \leq n$.

La réciproque de ce théorème est prouvée par le théorème suivant.

Théorème 3.2.4 (Théorème des automates multiplicateurs standards)**.** *Supposons que les conditions (i) et (ii) du théorème 3.2.3 soient satisfaites. Considérons $g \in G$ tel que $d_G(1, g) = p$ et choisissons des entiers positifs*

$$R_0 > Kp + K + 1,$$
$$R_1 > \tfrac{1}{2}D(2R_0) + R_0,$$
$$R_2 > \tfrac{1}{2}D(2R_1) + R_1,$$
$$R_3 > \tfrac{1}{2}D(2R_2) + R_2.$$

Alors l'automate multiplicateur asynchrone standard $\mathcal{M}_g$ basé sur $(\mathcal{A}, R_1, R_2, R_3)$, où $\mathcal{A}$ est un automate fini acceptant L, est un vrai multiplicateur[18] *: il accepte un couple (w_L, w_R) de mots sur Σ si et seulement si w_L, $w_R \in L$ et $\overline{w_L}x = \overline{w_R}$. De plus, l'automate $\mathcal{M}_g$ est asynchrone à lecture bornée avec un facteur asynchrone $D(R_3)$.*

Démonstration. Vu la construction des automates multiplicateurs standards, la condition nécessaire est évidente. De plus, pour la condition suffisante, il suffit de prouver que la distance entre $\overline{w_L(t_L)}$ et $\overline{w_R(t_R)}$ dans le graphe de Cayley de G n'atteint jamais R_3.

Considérons les intervalles $I_L = [0, |w_L|]$ et $I_R = [0, |w_R|]$ de $\mathbb{R}$ et notons par ψ la fonction, définie sur $I_L \times I_R$ et à valeurs dans $\mathbb{R}$, qui associe à chaque couple (s, t) de $I_L \times I_R$ la distance entre les éléments $\overline{w_L(s)}$ et $\overline{w_R(t)}$ dans le graphe de Cayley. Posons également

$$S_i = \psi^{-1}([0, R_i]),\ \forall i \in \{0, 1, 2, 3\}.$$

La figure 3.7 représente la partie $I_L \times I_R$ de $\mathbb{R}^2$. Le trait gras représente le chemin α que nous introduisons ci-dessous. Le trait plus fin, continu ou discontinu illustre l'évolution de (t_L, t_R) au fur et à mesure que l'automate $\mathcal{M}_g$ lit l'entrée (w_L, w_R) : le trait est continu lorsque $T = A$ et discontinu lorsque $T = B$. Lire la bande droite, i.e. $h = R$, implique un déplacement vers le haut et lire la bande gauche, i.e. $h = L$, implique un déplacement vers la droite. La région en blanc désigne l'ensemble S_1, celle en gris clair dénote $S_2 \backslash S_1$ et la partie gris foncé représente $S_3 \backslash (S_2 \cup S_1)$. Pour ne pas compliquer l'illustration, l'ensemble S_0 n'est pas représenté.

Nous commençons par remarquer que si deux points de $I_L \times I_R$ ont la même image par φ, à savoir φ_0, alors la valeur de φ est au plus $\frac{1}{2}D(2\varphi_0) + \varphi_0$ pour tous les points d'une même ligne[19], verticale ou horizontale, entre deux points.

En effet, supposons que nous nous trouvions sur une ligne horizontale, i.e. supposons que nous lisions le mot $w_L \in L$ et supposons que la différence de mots quitte la boule ouverte de rayon ψ_0 centrée en un point fixé et sorte de la boule ouverte de rayon $\frac{1}{2}D(2\psi_0) + \psi_0$ centrée en ce même point pour ensuite revenir dans la boule de rayon

[18] Les problèmes liés à la différence de mots sont résolus.

[19] Il s'agit des lignes représentant l'évolution de (t_L, t_R) dans la figure 3.7.

ψ_0. Cette évolution nécessite plus de $D(2\psi_0)$ étapes, ce qui contredit la définition de la fonction de départ D car nous avons

$$d_G(\overline{w_L(t)}, \overline{w_L(t+2\psi_0)} \leq 2\psi_0$$

si nous avons commencé à lire w_L à partir de la t-ième lettre.

Nous appliquons ceci aux cas où ψ_0 vaut R_0, R_1 et R_2, ce qui explique le choix de R_1, R_2 et R_3.

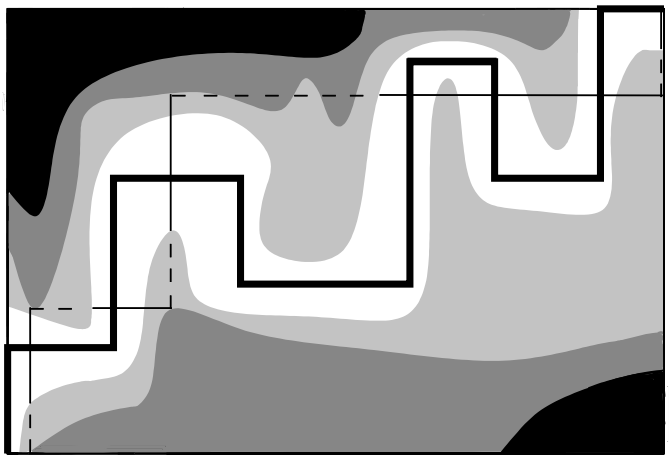

FIG. 3.7 – Représentation, dans $I_L \times I_R$, des ensembles S_1, S_2, S_3, de l'évolution de (t_L, t_R) et du chemin α.

Montrons maintenant que les éléments $(0,0)$ et $(|w_L|, |w_R|)$ appartiennent à la même composante connexe de S_1. Pour cela, nous montrons qu'il existe un chemin α dans S_1 joignant les deux points $(0,0)$ et $(|w_L|, |w_R|)$ de $I_L \times I_R$.

Commençons par choisir un point $(s, t(s))$ dans S_0 pour tout élément s de I_L. Ce choix est en effet possible car la distance de Hausdorff entre les deux chemins $\hat{w}_L$ et $\hat{w}_R$ est au plus[20] $Kp+K$ et puisque nous avons $R_0 > Kp+k+1$, nous avons $(s+r, t(s)) \in S_0$ pour $max(-s, -1) \leq r \leq min(1, |w_L| - s)$.

Il est facile de se convaincre que l'application $s \mapsto t(s)$ n'est en général ni continue, ni monotone. Nous choisissons $t(0) = 0$ et $t(|w_L|) = |w_R|$. Les segments $[s, s+1] \times \{t(s+1)\}$ et $\{s\} \times [t(s), t(s+1)]$ sont inclus dans S_0 et vu la définition de R_1, chacun de ces segments se trouve dans S_1. Il existe donc bien un chemin α dans S_1 joignant les deux points $(0,0)$ et $(|w_L|, |w_R|)$.

Étudions maintenant le comportement de l'automate $\mathcal{M}_g$ au fur et à mesure qu'il lit le couple (w_L, w_R). Nous commençons dans le coin $(0,0)$ du rectangle $I_L \times I_R$ et nous nous déplaçons vers la droite avec $T = A$ jusqu'à ce que nous arrivions à la frontière

[20] La distance Kp ne suffit pas car si $p = 0$, i.e. si l'élément g est l'identité, les deux chemins sont à une distance de Hausdorff K.

entre S_2 et S_3 ou au mur droit[21]. À cet instant, nous posons $T = B$ et nous nous déplaçons vers le haut. Nous remontons donc jusqu'à revenir dans S_1 où nous posons $T = A$. La situation est alors complètement analogue à la situation initiale avec L et R interchangés. Pendant tout ce temps, nous sommes toujours restés à l'intérieur de S_3 : nous nous sommes d'abord déplacés horizontalement d'un point avec $\psi \leq R_1$ à un point avec $\psi = R_2$, puis verticalement jusqu'à un autre point avec $\psi \leq R_1$. Par induction, nous ne sortirons jamais de S_3.

Il reste à prouver la dernière assertion. Supposons que $\mathcal{M}_g$ lise $k > D(2R_3)$ symboles d'affilée sur la même bande, par exemple sur la bande gauche. Vu la définition de la fonction de départ, nous obtenons $d_G(w_L(s), w_L(s+k)) > 2R_3$ si nous avons commencé à lire ces k symboles à partir du s-ième. Or, si l'automate avait déjà lu t symboles de w_R, nous avions

$$d_G(w_L(s), w_R(t)) < R_3.$$

Il est alors clair que $d_G(w_L(s+k), w_R(t)) > R_3$, d'où une contradiction. □

[21] Il s'agit du côté droit du rectangle $I_L \times I_R$.

3.3 Propriétés des groupes automatiques asynchrones

Nous étudions dans cette section les propriétés des groupes automatiques asynchrones. L'un des résultats les plus intéressants est, comme dans le cas synchrone, que le langage L des représentants satisfait la propriété d'unicité, i.e. L peut être choisi de telle sorte que, pour chaque élément g de G, il n'existe qu'un unique mot de L envoyé sur G. La démonstration de ce résultat est cependant plus subtile que pour le cas synchrone et nécessite le lemme suivant.

Lemme 3.3.1 (Préfixe commun)**.** *Soit $\mathcal{A}$ un automate asynchrone à lecture bornée sur un alphabet Σ. Il existe un entier K tel que si un couple (w_L, w_R) de mots sur Σ qui est accepté par $\mathcal{A}$ est tel que les préfixes $w_L(T)$ et $w_R(T)$ sont identiques pour un entier $T \geq 0$, alors $|t_L - t_R| < K$ si $t_L \leq T$ ou si $t_R \leq T$. En d'autres termes, les deux mots ne peuvent pas trop s'éloigner tant que $\mathcal{A}$ lit le préfixe qu'ils ont en commun.*

Démonstration. Supposons qu'à un instant donné, l'automate $\mathcal{A}$ a lu t_L symboles de w_L et t_R symboles de w_R. Supposons également que $w_L(t_L)$ est un préfixe de $w_R(t_R)$ (le même raisonnement s'applique si $w_R(t_R)$ est un préfixe de $w_L(t_L)$).

Nous notons par w le mot de $w_L(t_L) \sqcup\!\sqcup w_R(t_R)$ que l'automate a lu. Il existe alors un mot u sur $\Sigma^{\$}$ de longueur minimale tel que l'automate aboutit dans un état final en lisant wu. Si nous décomposons le mot u en les deux mots u_L et u_R qui correspondent aux mots qui sont lus sur chacune des bandes de l'automate en lisant[22] u, alors le couple $(w_L(t_L)u_L, w_R(t_R)u_R)$ de mots sur $\Sigma^{\$}$ est accepté par $\mathcal{A}$ et celui-ci se comporte exactement de la même façon, pour les $t_L + t_R$ premières étapes, sur l'entrée (w_L, w_R) et sur l'entrée $(w_L(t_L)u_L, w_R(t_R)u_R)$. Après ces étapes, l'automate doit encore lire les $|u_R|$ dernières lettres du mot de la bande de droite et les $t_R - t_L + |u_L|$ dernières lettres du mot de la bande gauche.

Si k est le facteur asynchrone de $\mathcal{A}$, cela signifie que

$$t_R - t_L + |u_L| \leq k|u_R|.$$

Or, nous avons également $|u_R| \leq k|u_L|$, et donc

$$t_R - t_L \leq (k^2 - 1)c,$$

où c est le nombre d'états de l'automate. Il suffit alors de prendre $K = (k^2 - 1)c$ pour conclure.

□

Théorème 3.3.2 (Propriété d'unicité : cas asynchrone)**.** *Si G est un groupe admettant la structure automatique asynchrone (Σ, L), alors G admet la structure automatique (Σ, L'), où le langage L' est un sous-langage de L qui ne contient qu'un unique représentant pour chaque élément de G.*

[22] i.e. le mot u est l'unique mot du shuffle de u_L et u_R qui est tel que le couple $(w_L(t_L)u_L, w_R(t_R)u_R)$ est accepté par l'automate $\mathcal{A}$.

Démonstration. Vu le théorème 3.2.2 (Théorème d'asynchronisation bornée), nous pouvons supposer que la structure automatique est bornée. Nous choisissons comme langage L' le langage qui contient, pour tout $g \in G$, le représentant de g dans L qui est le plus petit pour l'ordre du dictionnaire (et non l'ordre lexicographique[23]). Ce langage est bien défini car il n'existe qu'un nombre fini de représentations de g dans L. En effet, si $w \in L$ est un représentant de g, alors tout autre représentant de g dans L est de longueur majorée par $k|w|$, où k est le facteur asynchrone de l'automate égalité $\mathcal{M}_\varepsilon$ (ce choix de L' n'aurait donc pas été possible si la structure n'était pas bornée). Il suffit de montrer que le langage L' est régulier. Les automates multiplicateurs et l'automate égalité peuvent être facilement adaptés pour rejeter les mots n'appartenant pas à L'. Nous allons donc construire un AFND $\mathcal{A}'$ qui accepte exactement $L \backslash L'$. Dans ce cas, le langage L' est régulier, puisque $L' = (\Sigma^* \backslash (L \backslash L')) \cap L$.

L'idée est de considérer l'automate égalité $\mathcal{M}_\varepsilon$ et de le rendre capable de "deviner[24]" un de ses mots, en l'occurence celui de la bande gauche, en remplaçant toutes les transitions partant des états de $Q_L \cup {Q_L}^{\$}$ par des ε-transitions. Cependant, si nous ne changeons que cela, le langage accepté est simplement L. Nous devons donc transformer l'automate pour qu'il garde en mémoire la différence de mots entre le mot fantôme et le vrai mot jusqu'à ce qu'il puisse décider lequel des deux est le plus petit pour l'ordre du dictionnaire. Vu le lemme précédent, la mémoire nécessaire pour cela est bornée. Une fois que les deux mots divergent, si le mot fantôme est le plus petit, l'automate continue et essaye d'atteindre un état final en lisant ce qu'il reste du mot d'entrée ; s'il y parvient, le mot d'entrée n'est pas le plus petit et est donc accepté. Si par contre le mot fantôme est le plus grand, alors l'automate s'arrête et le mot d'entrée n'est pas accepté.

Pour formaliser ce qui précède, nous commençons par définir l'automate $\mathcal{A}'$ comme un automate asynchrone. Soit Q l'ensemble des états de l'automate $\mathcal{M}_\varepsilon$ et soit Σ_K l'ensemble des mots sur $\Sigma^{\$}$ de longueur au plus K, où K est la constante associée à l'automate $\mathcal{M}_\varepsilon$ fournie par le lemme précédent.

Nous notons par P le sous-ensemble de $Q \times \Sigma_K \times \Sigma_K$ formé des triplets (q, u, v), où u et v n'ont aucun préfixe en commun (excepté ε). Tout état de $\mathcal{A}'$ est un ensemble d'éléments de P défini comme suit. Pour tout état q de $\mathcal{M}_\varepsilon$, nous associons l'état $[q]$ défini par

$$[q] = \{\ (q, u, v) \in P \mid \varepsilon < u < v\ \},$$

où $<$ représente l'ordre du dictionnaire.

Nous défnissons également un puit consistant en l'ensemble des triplets de la forme (q, u, v), avec $q \in Q$ et $u > v > \varepsilon$, auxquels nous ajoutons le triplet $(q^{\$}, \varepsilon, \varepsilon)$, où $q^{\$}$ est l'état final de $\mathcal{M}_\varepsilon$. Signalons que dans l'ordre du dictionnaire, le symbole \$ précède

[23] Rappelons que l'ordre lexicographique tient compte de la longueur des mots ; il classe les mots par longueur et dans une classe de longueur, il ordonne grâce à l'ordre du dictionnaire. Ainsi, dans l'ordre lexicographique, le mot *vélo* vient avant le mot *avion*.

[24] En le rendant non déterministe.

tous les autres. Tout autre élément de P forme un état qui est le singleton le contenant. Remarquons que ces états sont soit de la forme $\{(q, \varepsilon, \varepsilon)\}$, avec $q \neq q^{\$}$, soit de la forme $\{(q, u, v)\}$ où uniquement un des deux mots u et v est le mot vide. L'état initial de $\mathcal{A}'$ est $\{(q_0, \varepsilon, \varepsilon)\}$, où q_0 est l'état initial de $\mathcal{M}_\varepsilon$, l'état final est l'état $[q^{\$}]$.

Définissons maintenant les transitions de l'automate. Pour toute transition (q, a, p) de l'automate $\mathcal{M}_\varepsilon$, nous définissons une transition $([q], a, [p])$ dans l'automate $\mathcal{A}'$ et il n'existe pas d'autre transition partant des états de la forme $[q]$.

En ce qui concerne les états singletons, nous définissons les transitions suivantes. Si σ est un élément de $\Sigma^{\$}$ et si $\{(q, u, v)\}$ est un état de $\mathcal{A}'$, alors la transition de label σ envoie le triplet (q, u, v) sur le triplet $(q.\sigma, u', v')$, où u' et v' sont obtenus en éliminant le plus grand préfixe commun de $u\sigma$ et v si $q \in Q_L \cup {Q_L}^{\$}$ ou de u et $v\sigma$ si $q \in Q_R \cup {Q_R}^{\$}$. S'il n'existe pas de transition dans $\mathcal{M}_\varepsilon$ partant de q et de label σ, ou si u' ou v' a une longueur supérieure à K, alors la transition de label σ dans $\mathcal{A}'$ aboutit dans le puit.

Explicitons quelque peu ces transitions lorsque la longueur des mots u' et v' ne dépasse pas K et lorsque la transition $q.\sigma$ est bien définie dans $\mathcal{M}_\varepsilon$ (on peut faire un raisonnement analogue pour $Q_R \cup {Q_R}^{\$}$).

Supposons pour commencer que $u = v = \varepsilon$. Dans ce cas, les mots $u\sigma$ et v n'ont pas de préfixe commun et la transition aboutit donc dans l'état $\{(q.\sigma, \sigma, \varepsilon)\}$ de $\mathcal{A}'$.

Supposons ensuite que $u \neq \varepsilon$ et que $v = \varepsilon$. Les mots ua et v n'ont encore une fois pas de facteur commun et la transition aboutit dès lors dans l'état $\{(q.\sigma, u\sigma, \varepsilon)\}$.

Enfin, supposons que $u = \varepsilon$ et $v = \sigma_1 \cdots \sigma_n \neq \varepsilon$, avec $a_i \in \Sigma^{\$}$ pour tout i. Trois cas sont possibles :

(i) si $\sigma = \sigma_1$, alors les mots $u\sigma$ et v ont le préfixe σ en commun et la transition aboutit donc dans l'état $\{(q.\sigma, \varepsilon, v_2 \cdots v_n)\}$;

(ii) si $\sigma > \sigma_1$, alors les deux mots n'ont aucun préfixe commun et puisque $u\sigma > v > \varepsilon$, la transition aboutit dans le puit ;

(iii) si $\sigma < \sigma_1$, les deux mots n'ont pas de préfixe commun et la transition aboutit dans l'état $[q.\sigma]$.

Vu toutes ces constructions, l'automate $\mathcal{A}'$ atteint un état de la forme $\{(q, u, v)\}$ si et seulement si les conditions suivantes sont satisfaites :

(i) au moins un des deux mots u et v est le mot vide,

(ii) $\mathcal{A}'$ a déjà lu l'entrée (wu, wv) pour un mot w,

(iii) après avoir lu la même entrée, l'automate $\mathcal{M}_\varepsilon$ se trouve dans l'état q,

(iv) $q \neq q^{\$}$.

De même, l'automate $\mathcal{A}'$ atteint un état de la forme $[q]$ si et seulement s'il a lu le couple $(w_L(t_L), w_R(t_R))$, où $w_L(t_L) < w_R(t_R)$ et si l'automate $\mathcal{M}_\varepsilon$ se trouve dans l'état q après avoir lu ce même couple.

Pour terminer notre construction, nous éliminons de $\mathcal{A}'$ tous les états inaccessibles et tous ceux à partir desquels nous ne pouvons atteindre l'état final. Nous éliminons

également toutes les transitions se rapportant à ces états. Nous remplaçons ensuite par des ε-transitions toutes les \$-transitions ainsi que toutes les transitions partant des états $[q]$ et $\{(q,u,v)\}$, où q est un état de $Q_L \cup {Q_L}^{\$}$. Ces changements ont pour effet que l'automate $\mathcal{A}'$ "devine" ses mots de la bande gauche ainsi que le symbole de fin de mot pour la bande droite (ce qui est nécessaire étant donné que le mot que $\mathcal{A}'$ lit ne se termine pas par un \$). Il découle de tout ce qui précède que l'automate non déterministe ainsi défini accepte exactement le langage $L\backslash L'$, d'où la conclusion. □

Le théorème suivant stipule que, comme dans le cas synchrone, le caractère automatique asynchrone d'un groupe est propre à ce groupe ; il ne dépend pas de l'ensemble des générateurs choisi.

Théorème 3.3.3 (Changement de générateurs)**.** *Si G est un groupe admettant une structure automatique asynchrone par rapport à un ensemble de générateurs de semi-groupe Σ, alors il admet une structure automatique asynchrone par rapport à tout autre ensemble de générateurs.*

Démonstration. Notons par (Σ, L) la structure automatique asynchrone sur G. Nous pouvons supposer que cette structure est bornée et que les conditions (i) et (ii) du théorème 3.2.3 sont satisfaites. Soit Σ' un autre ensemble de générateurs de semigroupe pour G, nous devons définir une structure automatique asynchrone sur G d'alphabet Σ'.

Pour tout $a \in \Sigma$, nous notons w_a le mot de longueur minimale sur Σ' représentant le même élément que a dans G. Réciproquement, pour tout $b \in \Sigma'$, nous notons v_b le mot de longueur minimale sur Σ représentant le même élément que b dans G. Puisque les alphabets Σ et Σ' sont finis, nous pouvons considérer la borne supérieure l sur la longueur des mots w_a et v_b, i.e.

$$l = max\{\ sup\{\ |w_a| \mid a \in \Sigma\ \},\ sup\{\ |v_b| \mid b \in \Sigma'\ \}\ \}.$$

Nous choisissons comme langage des représentants L' le langage obtenu à partir des mots de L où nous avons remplacé chaque lettre a de Σ par le mot w_a sur Σ' qui lui correspond.

Il suffit maintenant de montrer que la structure (Σ', L') satisfait les conditions (i) et (ii) du théorème 3.2.3.

Nous notons d_Σ et $d_{\Sigma'}$ la distance dans le graphe de Cayley du groupe G lorsqu'il est généré par Σ et par Σ' respectivement. Nous obtenons

$$l^{-1} d_{\Sigma'}(g,g') \leq d_\Sigma(g,g') \leq l d_{\Sigma'}(g,g'),\ \forall g,g' \in G.$$

Soit K une constante lipschitzienne pour le langage L ; considérons deux mots w et w' sur Σ' tels que $\overline{w} = \overline{w'b}$, avec $b \in \Sigma'$. À ces deux mots il correspond deux mots v et v' sur Σ qui sont tels que $\overline{v} = \overline{v'v_b}$ et la distance de Hausdorff entre les deux chemins correspondant dans le graphe de Cayley de G relatif à Σ est donc au plus Kl. Ainsi,

la distance de Hausdorff dans le graphe de Cayley relatif à Σ' est au plus Kl^2 et Kl^2 est une constante lipschitzienne pour L'.

Soit maintenant D une fonction de départ pour L ; montrons qu'il existe également une fonction de départ pour L'. Soit w un facteur d'un mot de L' tel que $|w| > 2l + D(rl + 2l^2)l$. Aux extrémités de w, nous trouvons des facteurs de mots w_a pour certaines[25] lettres a de Σ. En éliminant ces facteurs de w, nous obtenons un mot qui provient d'un facteur de longueur supérieure à $D(rl + 2l^2)$ d'un mot de L.

Vu la définition de la fonction de départ, ce facteur représente un élément g de G tel que

$$d_\Sigma(g, 1) > rl + 2l^2,$$

ou 1 est l'identité du groupe. Nous obtenons donc $d_{\Sigma'}(g, 1) > r + 2l$ et donc

$$d_{\Sigma'}(\overline{w}, 1) > r,$$

d'où la conclusion. □

Les deux théorèmes suivants sont des généralisations de théorèmes[26] que nous avons démontrés dans la cas synchrone. Ils se démontrent essentiellement de la même façon et nous nous contenterons donc de présenter les grandes étapes des démonstrations.

Théorème 3.3.4 (Inégalité isopérimétrique : cas asynchrone)**.** *Tout groupe automatique asynchrone est finiment présenté et satisfait une inégalité isopérimétrique exponentielle.*

Démonstration. Ce théorème se démontre comme le théorème 2.6.3 à la différence que nous nous servons du théorème 3.2.2 (Théorème d'asynchronisation bornée), à la place d'utiliser le lemme 2.6.1, afin de considérer une structure automatique asynchrone bornée.

Ainsi, si $w = a_1 \cdots a_n$ est un mot sur Σ, nous considérons à nouveau des représentants $u_0, \ldots u_n$ dans L des préfixes de w et nous disposons de l'équivalence

$$w \sim (u_0 a_1 u_1^{-1})(u_1 a_2 u_2^{-1}) \cdots (u_{n-1} a_n u_n^{-1}).$$

Nous décomposons alors comme nous l'avons fait dans le cas synchone chaque terme de la forme $(u_i a_{i+1} u_{i+1}^{-1})$ sous la forme

$$u_i(1)v(1)u_{i+1}(1)^{-1} \prod_{1 \leq t \leq m-1} u_{i+1}(t)v(t)^{-1} b_{t+1} v(t+1) u_{i+1}(t+1)^{-1}.$$

Comme pour tout $0 \leq i < n$, le couple (u_i, u_{i+1}) de mots est accepté par l'automate multiplicateur asynchrone à lecture bornée $\mathcal{M}_{a_{i+1}}$, la longueur du mot u_{i+1} est majorée par un multiple de $|u_i|$. La longueur du mot u_n est donc majorée par une exponentielle de $|w|$. Ainsi, le nombre de termes obtenus dans la décomposition ci-dessus est majoré par une exponentielle de $|w|$, d'où la conclusion. □

[25] Les premières lettres de w forment un suffixe d'un mot w_a pour une lettre a de Σ et les dernières lettres forment un préfixe d'un mot $w_{a'}$ pour une lettre a' de Σ.

[26] Il s'agit en fait des théorèmes 2.6.3, 2.7.1 et 2.7.3.

Vu la remarque faite dans la section 2.6.2, le problème du mot est donc décidable pour un groupe automatique asynchrone. Cependant, l'existence d'un algorithme permettant de le décider avec une complexité polynomiale n'a pas encore été démontrée.

Le théorème suivant est une généralisation de deux propriétés de fermeture que nous avons étudiées dans le cas synchrone.

Théorème 3.3.5. *1. Le produit direct de deux groupes automatiques asynchrones est un groupe automatique asynchrone.*

2. Soient G un groupe et H un sous-groupe d'index fini, G est automatique asynchrone si et seulement si H l'est.

Démonstration. 1. Comme dans le cas synchrone, si les couples (Σ_1, L_1) et (Σ_2, L_2) sont des structures asynchrones bornées respectivement sur les groupes G_1 et G_2, nous montrons que le couple $(\Sigma_1 \cup \Sigma_2, L_1 L_2)$ forme une structure asynchrone bornée sur le groupe $G_1 \times G_2$. Il est trivial que le langage $L_1 L_2$ est régulier et qu'il est envoyé surjectivement sur le groupe $G = G_1 \times G_2$ par l'homomorphisme π défini par $\pi(u_1 u_2) = (\bar{u}_1, \bar{u}_2)$. Il suffit donc de vérifier les conditions (i) et (ii) du théorème 3.2.3 (Caractérisation des structures automatiques asynchrones bornées).

Si D_1 et D_2 sont des fonctions de départ pour les groupes G_1 et G_2 respectivement, il est facile de vérifier que la fonction

$$D(r) = \max\{D_1(r), D_2(r)\}$$

est une fonction de départ pour le groupe[27] G.

Le point (ii) du théorème 3.2.3 (Caractérisation des structures automatiques asynchrones bornées) se démontre exactement de la même façon que le théorème 2.7.1 mais nous utilisons le caractère borné des structures automatiques asynchrones et la distance de Hausdorff à la place de la propriété du compagnon de voyage.

2. Comme dans le cas synchrone, si H est une sous-groupe d'index fini de G, nous décomposons G en l'union de ses classes latérales, i.e.

$$G = \bigcup_{0 \leq i \leq n} H x_i,$$

où x_0 est l'identité de G. Nous procédons exactement comme dans le cas synchrone pour trouver l'ensemble des générateurs Δ et le langage des représentants L' du sous-groupe H. Il suffit alors encore une fois de vérifier les conditions (i) et (ii) du théorème 3.2.3 (Caractérisation des structures automatiques asynchrones bornées).

Nous disposons d'une fonction de départ D pour G, il est donc évident que nous pouvons en trouver une pour le sous-groupe H. Nous démontrons alors le point (ii) de la même façon que le théorème 2.7.3 en utilisant le caractère borné des structures

[27] Il suffit de vérifier la définition d'une fonction de départ, avec $u_1 u_2 \in L$, pour les cas $0 \leq s \leq s + t \leq |u_1|$, $0 \leq s \leq |u_1| < s + t \leq |u_1 u_2|$ et $|u_1| < s \leq s + t \leq |u_1 u_2|$.

automatiques asynchrones et la distance de Hausdorff à la place de la propriété du compagnon de voyage.

La réciproque se démontre exactement comme dans le cas synchrone, en utilisant encore une fois le caractère borné des structures automatiques asynchrones et la distance de Hausdorff.

□

3.4 Exemple : les groupes de Baumslag-Solitar

Nous définissons dans cette section les groupes de Baumslag-Solitar qui sont des groupes automatiques asynchrones mais pas synchrones. Les méthodes nécessaires à la démonstration[28] n'ont pas été développées dans ce travail[29] ; nous nous contentons ici d'étudier son graphe de Cayley.

Commençons par définir ces groupes. Pour tout entier $p, q > 0$, le groupe $G_{p,q}$ de Baumslag-Solitar est le groupe de présentation

$$G_{p,q} = \langle \{a, b\} \ ; \ \{ba^p b^{-1} a^{-q}\} \rangle .$$

Ce groupe est en fait automatique asynchrone mais pas synchrone pour $p \neq q$ et est automatique synchrone pour $p = q$.

Nous pouvons supposer sans perte de généralité que $p \leq q$. En effet, si $ba^p b^{-1} a^{-q} = 1$, nous avons $b^{-1} a^q b a^{-p} = 1$.

Nous commençons par représenter l'unique relation $ba^p b^{-1} a^{-q}$ par un rectangle dont le label des côtés verticaux est b, dont le côté horizontal supérieur et divisé en p segments de même longueur et de label x. La figure 3.8 représente la relation du groupe $G_{1,2}$.

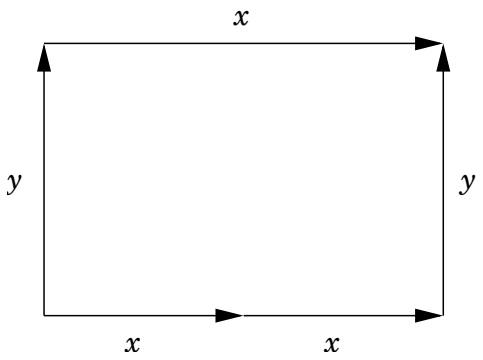

FIG. 3.8 – Représentation de l'unique relation du groupe de Baumslag-Solitar $G_{1,2}$ dans son graphe de Cayley.

Nous assemblons maintenant des copies de ce rectangle pour construire le graphe de Cayley de G. Pour le groupe $G_{1,2}$, nous obtenons le graphe représenté à la figure 3.9.

Nous commençons par assembler des copies du rectangle par les côtés verticaux et nous obtenons donc une bande horizontale infinie. Pour obtenir le graphe de Cayley complet, chaque sommet doit être l'extrémité de deux arcs de label y, un partant de ce sommet, l'autre y aboutissant. Par conséquent, sur le côté horizontal supérieur de notre bande, nous devons coller q nouvelles bandes par leur côté inférieur et décalées les unes par rapport aux autres. Toutes ces bandes ont donc un côté en commun.

[28] Il s'agit de l'ensemble des règles de Knuth-Bendix

[29] Voir [2] pour la démonstration.

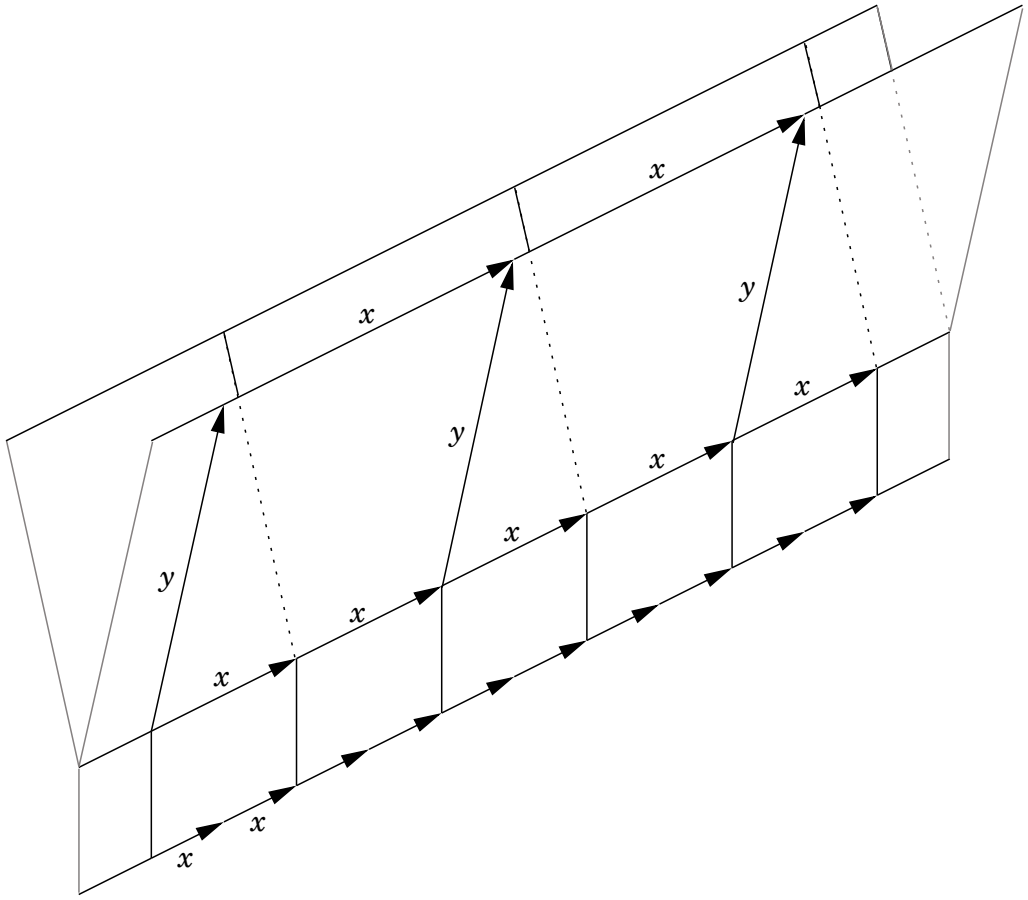

FIG. 3.9 – Partie du graphe de Cayley du groupe de Baumslag-Solitar $G_{1,2}$.

De même, il doit y avoir p bandes "pendues" au côté horizontal inférieur de la bande de départ. Remarquons que si $p = q = 1$, nous obtenons le quadrillage habituel du plan euclidien par des rectangles.

Si nous regardons le graphe obtenu de profil, nous obtenons un arbre infini T pour lequel chaque noeud a q arcs dirigés vers le haut et p arcs dirigés vers le bas. La figure 3.10 représente l'arbre obtenu pour $G_{1,2}$.

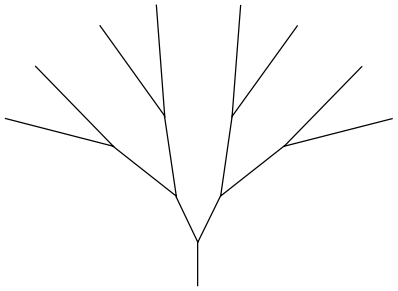

FIG. 3.10 – Arbre du graphe de Cayley du groupe de Baumslag-Solitar $G_{1,2}$.

Bibliographie

[1] C. CHOFFRUT, Semigroups, Algorithms, Automata and Languages, *A short introduction to automatic group theory*, G. M. S. Gomes, J-E Pin, P. Silva Ed., World Scientific, 133-154, 2002.

[2] David B. A. EPSTEIN, J. W. CANNON, D. F. HOLT, S. V. F. LEVY, M. S. PATERSON et W. P. THURSTON, *Word Processing in Groups*, 3-61 et 135-160, Jones and Bartlett Publishers, 1992.

[3] J. SHALLIT, Numerotation Systems, Linear Recurences, and Regular Sets, *Information and Computation*, Vol 113, 1994

[4] M. RIGO, *Théorie des automates et langages formels*, Notes de cours, ULG, 2005.

[5] G. HANSOUL, *Algèbre*, Notes de cours, ULG, 2002.

Table des matières