Table des matières

1 Notions fondamentales **19**

1 Notations et généralités 19
2 Graphes orientés 20
3 Structures algébriques 21
3.1 Monoïdes 21
3.2 Semi-anneaux 25
3.3 Polynômes et séries formelles 28
3.4 Ensembles et séries rationnels 30
3.5 Ensembles reconnaissables 31
4 Automates 31
4.1 Automates sur un semi-anneau 31
4.2 Automates sur un monoïde 33
4.3 Automates sur un alphabet 35
4.4 Automates à multiplicité 41

2 Automate universel **45**

1 Définitions et propriétés de l'automate universel 46
1.1 Automate universel dans un monoïde quelconque 46
1.2 Automate universel d'un ensemble reconnaissable 51
1.3 Automate universel et générateurs du monoïde 52
1.4 Automate universel d'un langage rationnel de A^* 54
2 Calcul effectif de l'automate universel 56
3 Écorché de l'automate universel 59
4 Développement d'un automate 64
4.1 Motivations et définitions 64
4.2 Propriétés de l'automate développé 68
4.3 Écorché du développé 70

3 Automates universels et langages réversibles **75**

1 Langages réversibles 76
2 Langages à groupe 80
3 Automate universel d'un langage à groupe 80
3.1 Structure générale de l'automate universel d'un langage à groupe . . 80
3.2 Structure des composantes fortement connexes de l'automate universel 82
4 Automate universel d'un langage réversible 86
4.1 Structure générale de l'automate universel d'un langage réversible . 86
4.2 Structure des pelotes de l'automate universel 88
5 Construction d'un automate réversible 90
5.1 Cordes 90
5.2 Automate quasi-réversible et automate universel 91

4 Hauteur d'étoile **97**

1 Hauteur d'étoile et degré d'enlacement 98
1.1 Hauteur d'étoile d'un langage rationnel 98
1.2 Enlacement d'un graphe orienté 99
1.3 Enlacement et hauteur d'étoile 104
1.4 Du calcul d'une expression au théorème d'Eggan 106
2 Hauteur d'étoile des langages à groupe 110
3 Hauteur d'étoile des langages réversibles 114
4 Automate universel et hauteur d'étoile 120

5 Déterminisation des automates (max,+) **123**

1 Le semi-anneau tropical et sa famille 124
2 Caractérisation des séries séquentielles 125
2.1 Séries translatées 125
2.2 Le problème d'une caractérisation topologique 128
3 Décidabilité de la séquentialité dans le cas des alphabets unaires 130
4 Algorithmes 135
4.1 Décidabilité 135
4.2 Déterminisation 136
5 Non-ambiguïté des séries rationnelles sur un alphabet à une lettre 140
6 Automates univoques 143
7 Le cas général 147
8 Problème de la généralisation à d'autres semi-anneaux 148

6 Dérivation d'expressions rationnelles avec multiplicité 149

1 Expressions rationnelles 150
2 Motivation de la dérivation 154
3 Dérivation et termes dérivés 156
4 L'automate des termes dérivés 165
5 Les termes dérivés fantômes 167
6 Variations 169
7 Le cas commutatif 172

— ∘ —

Chapitre 4

Hauteur d'étoile

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

J. Brel, *La quête*

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

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

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

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

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

1. « transition graph », à l'époque.

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

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

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

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

— ∘ —

1 Hauteur d'étoile et degré d'enlacement

1.1 Hauteur d'étoile d'un langage rationnel

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

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

i) 0, 1 *et tout élément de A est une expression rationnelle (atomique).*

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

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

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

façon dont elle a été construite.

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

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

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

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

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

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

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

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

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

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

— ∘ —

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

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

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

— ∘ —

1.2 Enlacement d'un graphe orienté

La pertinence de la hauteur d'étoile comme mesure de la complexité d'un langage est renforcée par le fait que cette grandeur ne mesure pas seulement la complexité des expressions qui mesurent le langage mais aussi celle des automates qui le reconnaissent, ou plus exactement, celle de leur graphe sous-jacent. Ce résultat, que nous développerons

plus loin, est dû à Eggan [22] qui posa le premier les questions relatives à la hauteur d'étoile.

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

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

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

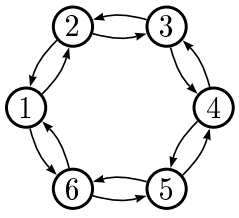
Étant donné un ordre total ω sur un ensemble, tout sous-ensemble est naturellement ordonné par ω. Dans ce qui suit, $\overline{\omega}$ désigne le maximum selon ω du sous-ensemble dans lequel on se trouve.

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

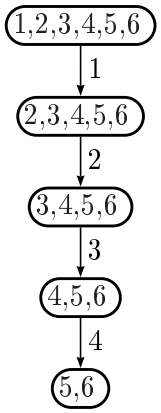
- $\mathsf{enl}_\omega(\mathcal{G}) = 0$ *si $\mathcal{G}$ est acyclique.*
- *Si $\mathcal{G}$ n'est pas une pelote,* $\mathsf{enl}_\omega(\mathcal{G}) = \mathsf{max}\{\mathsf{enl}_\omega(\mathcal{P}) \mid \mathcal{P}$ *pelote de* $\mathcal{G}\}$.
- *Si $\mathcal{G}$ est une pelote,* $\mathsf{enl}_\omega(\mathcal{G})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

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

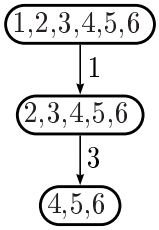
EXEMPLE 17 On veut calculer l'enlacement du graphe suivant :



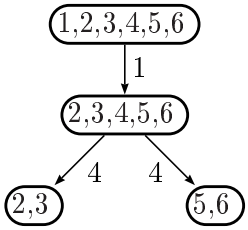
On peut fixer différents ordres sur les sommets. Soit $\omega_1 = (1,2,3,4,5,6)$, $\omega_2 = (1,3,5,2,4,6)$ et $\omega_3 = (1,4,2,5,3,6)$. On suppose que l'élément le plus grand est ici le premier de la liste. Le calcul de l'enlacement s'effectue comme indiqué sur la figure 1. Chaque nœud représente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



(a) Enlacement selon ω_1.



(b) Enlacement selon ω_2.



(c) Enlacement selon ω_3.

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

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

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

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

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

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

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

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

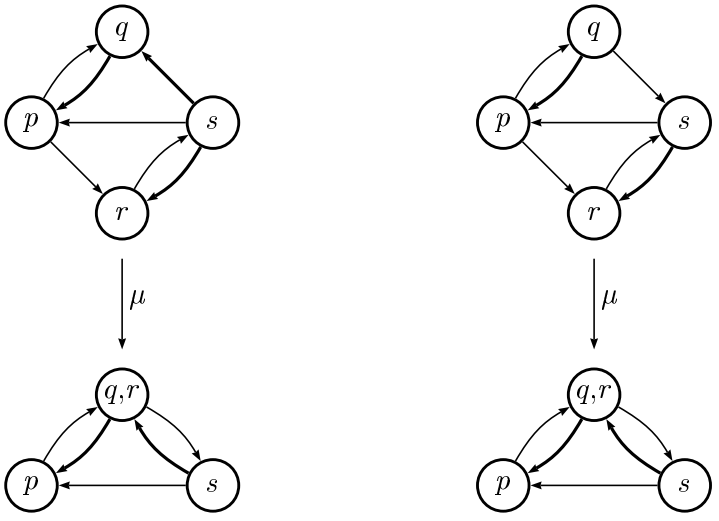
ce qui contredit la minimalité de $\mathcal{G}$. □

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

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

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

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



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

Fig. 2 – *Deux morphismes de graphes.*

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

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

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

Cette proposition est issue du lemme suivant :

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

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

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

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

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

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

Il existe une pelote $\mathcal{Q}$ de $\mathcal{H}$ de degré d, et d'après le lemme, il existe une pelote de $\mathcal{G}$ dont $\mathcal{Q}$ est l'image. Cette pelote $\mathcal{P}$ est au plus de degré c. Elle ne peut être de degré inférieur par minimalité de c. Il y a donc un état q de $\mathcal{P}$ tel que $\mathsf{enl}(\mathcal{P} \setminus \{q\}) = c - 1$. D'où $\mathsf{enl}(\mathcal{Q} \setminus \{q\mu\}) \geqslant d-1$ et $\mathsf{enl}(\mathcal{P} \setminus \{q\mu\mu^{-1}\}) \leqslant c-1$. On a donc construit un morphisme

3. plutôt que « conserve », puisque l'enlacement peut décroître.

conforme entre $\mathcal{P}\!\!\setminus\!\!\{q\mu\mu^{-1}\}$ et $\mathcal{Q}\!\!\setminus\!\!\{q\mu\}$ qui contredit la minimalité de c. Donc la proposition est vraie. □

— ∘ —

Maintenant que nous disposons d'un outil suffisament puissant pour manipuler l'enlacement de graphes, il faut établir le lien entre cette notion et la hauteur d'étoile d'un langage rationnel.

Définition 4.8 *L'**enlacement d'un automate** est égal à l'enlacement de son graphe sous-jacent. De même, un morphisme d'automate est conforme s'il est surjectif et que chaque chemin de l'image peut se relever en un chemin.*

La proposition 4.2 s'applique évidemment aux automates. Une illustration de l'utilisation de cette proposition est donnée par le résultat suivant :

Proposition 4.4 *Soit $\mathcal{A}$ un automate déterministe émondé reconnaissant le langage $\mathcal{L}$. Il existe un morphisme conforme de $\mathcal{A}$ sur l'automate minimal de $\mathcal{L}$. Par conséquent, l'automate minimal d'un langage est celui qui, parmi tous les automates déterministes reconnaissant ce langage, a un enlacement minimum.*

Démonstration. Soit $\mathcal{A}_\mathcal{L} = \langle\, Q,A,E,\{i\},T \,\rangle$ l'automate minimal de $\mathcal{L}$. L'application qui, à tout état p de $\mathcal{A}$, associe l'état $i \cdot \mathsf{Past}_\mathcal{A}(p)$ est un morphisme. Tout chemin $\mathcal{C}$ de $\mathcal{A}_\mathcal{L}$ peut se prolonger en un calcul étiqueté par un mot u. Ce mot u est accepté par l'automate $\mathcal{A}$; il existe donc un calcul de $\mathcal{A}$ étiqueté par u dont l'image dans $\mathcal{A}_\mathcal{L}$ ne peut être que $\mathcal{C}$ puisque $\mathcal{A}_\mathcal{L}$ est déterministe. Le morphisme est donc conforme. La seconde partie de la proposition découle de la proposition 4.2. □

— ∘ —

1.3 Enlacement et hauteur d'étoile

L'enlacement, on l'a vu, est relatif au graphe sous-jacent à un automate; la hauteur d'étoile, elle, correspond à une expression rationnelle. Le théorème d'Eggan [22] permet d'établir le lien entre ces deux grandeurs :

Théorème 4.1 *La hauteur d'étoile d'un langage rationnel est égale à l'enlacement minimal des automates qui reconnaissent le langage.*

Pour cela, on utilise des automates qui sont non plus étiquetés par des lettres, mais par des expressions rationnelles :

Définition 4.9 *Un quintuplet $\mathcal{A} = \langle\, Q,\mathsf{RExp}A,E,I,T \,\rangle$ est un **automate généralisé** si Q est un ensemble fini d'états, $\mathsf{RExp}A$ l'ensemble des expressions rationnelles sur l'alphabet A, E un sous-ensemble de $Q \times \mathsf{RExp}A \times Q$ et I et T des sous-ensembles de Q.*

Un automate $\mathcal{A}$ est donc un automate généralisé particulier, puisque les seules expressions qui étiquettent ses transitions sont des lettres.

Nous alons étendre la notion d'enlacement aux automates généralisés. On définit l'indice « Eggan »[4] i_{E} d'un automate généralisé. Cette méthode pour prouver le théorème d'Eggan [22] a été explicitée dans une communication faite en collaboration avec J. Sakarovitch [42].

Définition 4.10 *Soit $\mathcal{A} = \langle Q, A^*, E, I, T \rangle$ un automate généralisé et ω un ordre total sur Q. L'**indice** i_{E} **d'un automate généralisé** $\mathcal{A}$ selon ω, est défini récursivement par :*

– *Si $\mathcal{A}$ est acyclique :*

$$\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A}) = \mathsf{max}\{\mathsf{h}[|e|] \mid e \in E\}.$$

– *Si $\mathcal{A}$ n'est pas une pelote :*

$$\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A}) = \mathsf{max}(\{\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{P}) \mid \mathcal{P} \text{ pelote de } \mathcal{A}\} \cup \{\mathsf{h}[|e|] \mid e \text{ n'appartient pas à une pelote}\}).$$

– *Si $\mathcal{A}$ est une pelote,*

$$\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A}) = 1 + \mathsf{max}(\{\mathsf{i}_{\mathrm{E}}^{(\omega)}(\mathcal{A} \setminus \{\overline{\omega}\})\} \cup \{\mathsf{h}[|e|] \mid e \text{ adjacente à } \overline{\omega}\}).$$

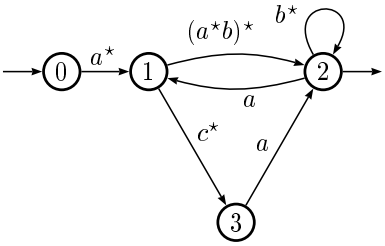
Les deux types extrêmes d'automates généralisés représentant un langage sont d'une part l'automate classique étiqueté par des lettres et d'autre part un automate réduit à un état initial et un état final dont l'unique transition (entre ces deux états) est étiquetée par une expression représentant le langage.

On voit que l'indice i_{E} d'un automate classique est exactement égal à l'enlacement selon le même ordre et que l'indice i_{E} d'un automate du second type est exactement la hauteur d'étoile de l'expression qui l'étiquette. On va montrer qu'on peut passer du premier type au second et que l'indice i_{E} reste invariant au cours de cette transformation.

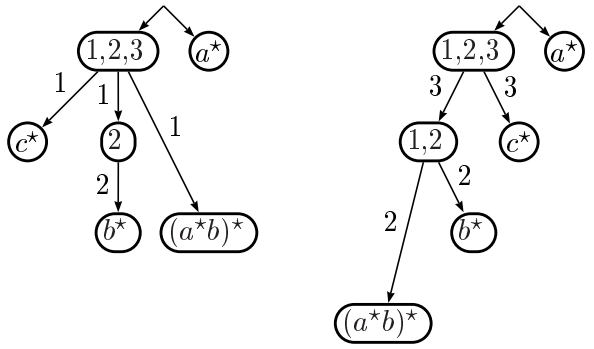
De même que le calcul de l'enlacement d'un graphe, le calcul de l'indice i_{E} peut se représenter par un arbre. La racine de cet arbre est l'automate lui-même, Cependant, dans ce cas, on a deux sortes de nœuds. D'une part, comme précédemment les nœuds qui correspondent à des pelotes, après suppressions successives d'états. D'autres part, des feuilles qui correspondent aux expressions rationnelles qui, soit étiquettent des transitions adjacentes à l'état qu'on supprime, soit étiquettent des transitions qui se retrouvent en dehors des pelotes. Les transitions dans l'arbre qui relient ces feuilles à leur père ont une longueur qui n'est pas obligatoirement égale à 1 mais à la hauteur d'étoile de l'expression qu'elles représentent (c'est la raison pour laquelle, on ne représente pas les expressions de hauteur d'étoile 0 et qu'on retrouve dans ce cas les arbres présentés pour l'enlacement).

Exemple 19 Considérons l'automate généralisé suivant :

4. Du nom du théorème qu'il permet de prouver.



Calculons l'indice $\mathrm{i_E}$ pour les ordres $\omega_1 = (0,1,2,3)$ et $\omega_2 = (3,2,1,0)$. Les arbres correspondants sont représentés figure 3. Dans le premier cas, l'indice $\mathrm{i_E}$ est égal à 3, dans le second, il est égal à 4.



(a) Indice $\mathrm{i_E}$ selon ω_1. (b) Indice $\mathrm{i_E}$ selon ω_2.

FIG. 3 – *Calcul de l'indice* $\mathrm{i_E}$.

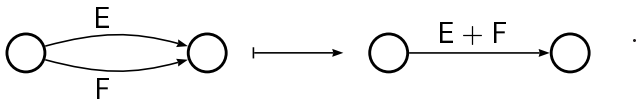
— ∘ —

1.4 Du calcul d'une expression au théorème d'Eggan

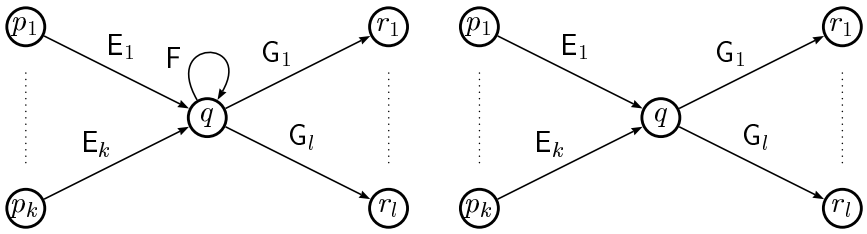
Rappelons une méthode classique, dite méthode d'élimination, pour obtenir une expression rationnelle à partir d'un automate.

Soit $\mathcal{A} = \langle\, Q,A,E,I,T \,\rangle$ un automate et ω un ordre total sur Q, ensemble des états de $\mathcal{A}$. On normalise $\mathcal{A}$, c'est-à-dire qu'on rajoute deux états i (initial) et t (terminal), des transitions spontanées (c'est-à-dire étiquetées par 1) entre i et chaque état de I d'une part, et entre chaque état de T et t d'autre part.
On obtient l'automate $\mathcal{A} = \langle\, Q \cup \{i,t\},A,E',i,t \,\rangle$.

On va ensuite éliminer les états de $\mathcal{A}_0$ dans l'ordre ω, en modifiant les transitions de sorte qu'à chaque étape, le langage représenté par l'automate reste le même. Pour cela, on considère que les étiquettes de l'automate sont des expressions rationnelles (au départ, réduites à des lettres). On profite donc de cette généralisation pour fixer une règle qui permet de n'avoir qu'une seule transition entre un couple donné d'états :



Lorsqu'on supprime un état q dans une des configurations suivantes,



on crée respectivement les transitions suivantes :

Tout calcul passant par p est étiqueté par un mot qui se factorise en $u.v.w$, où u étiquette un chemin de i à p_i, w un chemin de r_j à t et v un chemin de p_i à r_j dont les seuls états intérieurs sont p. Donc w appartient bien au langage dénoté par $E_i \cdot F^\star \cdot G_j$ et, après suppression de p, l'automate reconnaît bien au moins tous les mots qu'il reconnaissait avant. Réciproquement, tout mot du langage dénoté par $E_i \cdot F^\star \cdot G_j$ (ou par $E_i \cdot G_j$, ce qui est encore plus simple) peut se décomposer en $u.v^k.w$, avec $u \in |E_i|$, $v \in |F|$ et $w \in |G_j|$. Avant suppression de p, il existe donc un chemin de p_i à r_j, dont le seul état intérieur est p et dont l'étiquette contient $u.v^k.w$.

Lorsqu'on a supprimé tous les états de Q, on est donc dans la configuration suivante :

Comme à chaque étape le langage dénoté est toujours le même, l'expression E représente le langage.

PROPOSITION 4.5 *L'expression obtenue à partir d'un automate émondé $\mathcal{A}$ en suivant la méthode d'élimination des états selon l'ordre ω représente le langage reconnu par l'automate $\mathcal{A}$. La hauteur d'étoile de cette expression est égale à l'enlacement de $\mathcal{A}$ selon ω.*[5]

On suit la démonstration faite dans [42].

Démonstration. La preuve est par récurrence sur le nombre d'états n de $\mathcal{A}$. (En fait l'automate a $n+2$ états, puisqu'il a un état initial et un état final qui lui sont ajoutés.) On montre que l'indice $\mathrm{i_E}$ d'un automate généralisé de taille n selon ω est égal à la hauteur d'étoile de l'expression obtenue en éliminant les états dans l'ordre ω.

Si $n = 0$, l'automate ne comporte au plus qu'une transition entre l'état initial et l'état terminal. C'est la configuration finale de l'algorithme d'élimination. La hauteur d'étoile de cette expression est justement l'index $\mathrm{i_E}$ de l'automate.

Supposons la propriété vraie pour tout entier strictement inférieur à n et $\mathcal{A}$ un automate à $n(+2)$ états.

5. Il faut souligner que le calcul de l'enlacement de $\mathcal{A}$ selon ω consiste à traiter les états dans l'ordre décroissant selon ω, alors que la méthode d'élimination le fait dans l'ordre croissant.

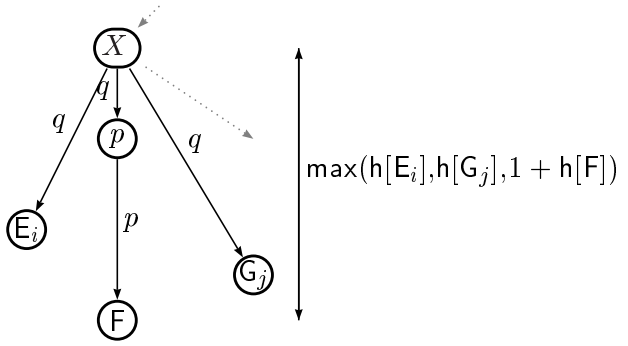
Si l'automate $\mathcal{A}$ (abstraction faite des états i et t) n'est pas une pelote, l'expression rationnelle obtenue par l'algorithme d'élimination est une somme ou un produit d'expressions qui, soit étiquettent des transitions en dehors des pelotes, soit correspondent à une pelote. La hauteur d'étoile de l'expression est le maximum des hauteurs d'étoile de toutes ces composantes. Par hypothèses de récurrence, la hauteur d'étoile des expressions issues de pelotes est égale à l'indice i_{E} de ces pelotes; la hauteur d'étoile de l'automate est donc égale à son indice i_{E}.

Si l'automate $\mathcal{A}$ (abstraction faite des états i et t) est une pelote, on pose p le plus petit état de $\mathcal{A}$ (selon ω). Soit $\mathcal{A}'$ l'automate obtenu après suppression de p dans l'algorithme d'élimination. Remarquons que, si $\mathcal{A}$ est une pelote, $\mathcal{A}'$ aussi.
L'algorithme de calcul de l'indice i_{E} va se dérouler de la même manière dans $\mathcal{A}$ et $\mathcal{A}'$ jusqu'à la dernière étape (c'est-à-dire l'examen de p), car les transitions qui relient leurs états sont les mêmes.

Les étiquettes des transitions de $\mathcal{A}'$ se divisent en deux parties :

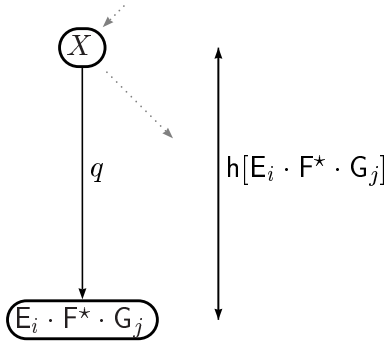
- d'une part, les transitions identiques à celles de $\mathcal{A}$;

- d'autre part, des transitions issues de la suppression de p par l'algorithme d'élimination et qui sont, soit de la forme $\mathsf{E} \cdot \mathsf{G}$ si p n'a pas de boucle dans $\mathcal{A}$, soit de la forme $\mathsf{E} \cdot \mathsf{F}^\star \cdot \mathsf{G}$, sinon. Dans le premiers cas, les enlacements de $\mathcal{A}$ et $\mathcal{A}'$ sont évidemment égaux; dans le second cas, examinons l'arbre qui représente le calcul de l'indice i_{E}. Soit X l'ensemble d'états qui étiquette le père de p dans l'arbre correspondant à $\mathcal{A}$ et q le plus grand élément de X. La suppression de q dans le calcul conduit à une fragmentation de la pelote en différentes pelotes dont $\{p\}$. Localement, l'arbre à donc la forme suivante :



Dans $\mathcal{A}'$, comme p n'existe plus, la suppression de q entraîne que les transitions étiquetées par les expressions $\mathsf{E}_i \cdot \mathsf{F}^\star \cdot \mathsf{G}_j$ sont soit supprimées, soit se retrouvent en

dehors de toute pelote, l'arbre a donc la forme suivante :



Les deux arbres correspondants aux deux calculs ont donc la même hauteur, et l'indice i_{E} de $\mathcal{A}'$ est égal à celui de $\mathcal{A}$.

Ceci permet de conclure la démonstration. □

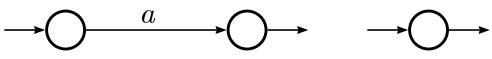
Démonstration du théorème 4.1. D'après la proposition précédente, la hauteur d'étoile d'un langage est inférieure ou égale à l'enlacement minimal des automates qui reconnaissent le langage. En effet, étant donné un automate d'enlacement minimal, on peut calculer une expression dont la hauteur d'étoile est égale à l'enlacement de l'automate.

On montre maintenant l'inégalité dans l'autre sens, c'est-à-dire qu'à partir d'une expression, on peut construire un automate qui décrive le même langage et dont l'enlacement est égal à la hauteur d'étoile de l'expression. Pour cela, on construit un automate en suivant les règles de formation d'une expression rationnelle. On peut noter qu'à part pour le langage vide représentable par un automate vide dont l'enlacement est bien égal à la hauteur d'étoile du langage, dans toute expression rationnelle, on peut supprimer la constante $\mathsf{0}$ en utilisant les règles suivantes :

$$(\mathsf{0} \cdot \mathsf{E}) \mapsto \mathsf{0}, \quad (\mathsf{0} + \mathsf{E}) \mapsto \mathsf{E}, \quad \text{et } (\mathsf{0})^\star \mapsto \mathsf{1}.$$

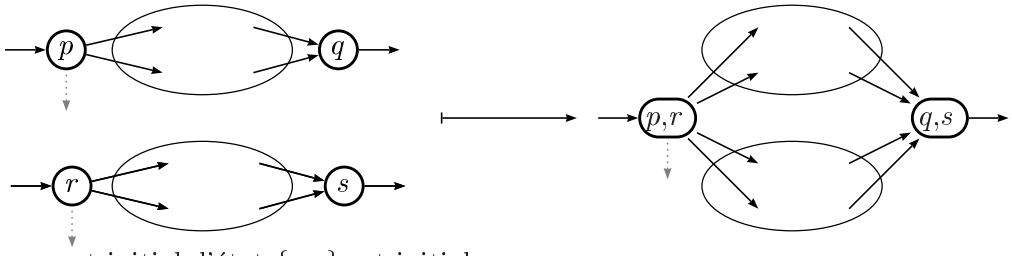
Si l'application de ces règles ne conduit pas à une suppression de $\mathsf{0}$, c'est que le langage est vide. On considère donc les expressions rationnelles sans $\mathsf{0}$. On construit récursivement un automate qui décrit le même langage que l'expression en assurant que l'enlacement du premier est égal à la hauteur de la seconde. On construit des automates quasi-normalisés, dans le sens où il n'y a pas de transition spontanée et que, éventuellement, l'état initial peut être final (en plus de l'état final « unique »).

i) Si l'expression est a ou 1, le langage est reconnu respectivement par un des deux automates suivants d'enlacement 0.



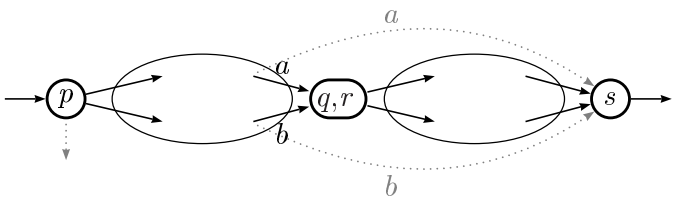
ii) Si E et F sont deux expressions pour lesquelles il existe deux automates $\mathcal{A}$ et $\mathcal{B}$ qui décrivent le même langage, et dont l'enlacement respectif est égal à la hauteur d'étoile respective de chacune des expressions, alors $\mathsf{E}+\mathsf{F}$ représente le langage reconnu par l'automte

suivant, dont l'enlacement est le maximum des enlacements de $\mathcal{A}$ et $\mathcal{B}$, donc la hauteur d'étoile de $\mathsf{E}+\mathsf{F}$:



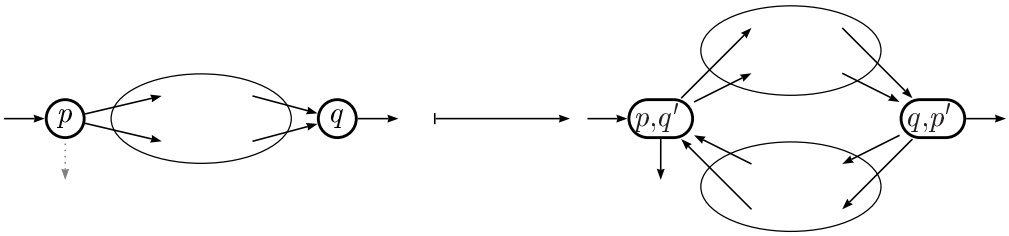
Si p ou r est initial, l'état $\{p,r\}$ est initial.

iii) Pour le produit, la construction est similaire :



Si p est final dans le premier automate, il l'est dans le résultat; si r est final, on crée dans le résultat, pour toute transition qui arrive en $\{q,r\}$, une transition étiquetée par la même lettre qui arrive en s.

iv) Pour l'étoile, à partir d'un automate, on crée deux copies et on réunit l'état initial de la première avec l'état final de la seconde et réciproquement :



Si on enlève l'état initial ou l'état final, on obtient un automate dont l'enlacement est égal à celui de départ, l'enlacement est donc inférieur à la hauteur d'étoile de E plus 1. Si on enlève un état interne, il appartient à l'une des deux copies; l'autre copie n'est pas modifiée, l'enlacement diminue donc au plus d'1. □

— ∘ —

2 Hauteur d'étoile des langages à groupe

De même que pour l'étude de l'automate universel, l'algorithme présenté pour les automates à groupe, qui est une adaptation de celui de R. McNaughton [47], se veut un avant-goût de celui qui est présenté pour les automates réversibles.

Soit $\mathcal{L}$ un langage à groupe et $\mathcal{A}$ son automate minimal. Cet automate est une pelote, son monoïde de transition est le groupe syntaxique G dont l'unique idempotent est 1_G.

Dans tout automate $\mathcal{B}$ qui accepte $\mathcal{L}$, on peut trouver un état r qui se comporte dans $\mathcal{B}$ sensiblement de la même façon que l'état initial dans $\mathcal{A}$.

LEMME 4.6 *Soit $\mathcal{L}$ un langage dont le monoïde syntaxique est un groupe G. Son automate minimal est $\mathcal{A} = \langle Q,A,E,i,T \rangle$ et soit $\mathcal{B}$ un automate équivalent. Soit g un élément du groupe syntaxique G correspondant à une partie du langage, H_g son image inverse dans A^* (H_g est inclus dans $\mathcal{L}$) et H_{1_G} l'image inverse de 1_G. Il existe un état r de $\mathcal{B}$ tel que :*

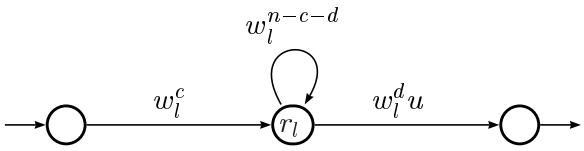
i) *$H_{1_G} \cap \mathsf{Past}_{\mathcal{B}}(r) \neq \emptyset$ et $H_g \cap \mathsf{Fut}_{\mathcal{B}}(r) \neq \emptyset$.*

ii) *Pour tout mot v de H_{1_G} [6], il existe u dans H_{1_G} et w dans A^* tels que $u.v.w$ étiquette une boucle autour de r.*

Démonstration. Pour tout entier l positif, on définit B_l l'ensemble des mots de longueur inférieure à l qui étiquettent une boucle autour de i dans $\mathcal{A}$. Il existe un entier k (l'ordre du groupe syntaxique G) tel que pour tout mot u de B_l, le mot u^{k} appartient à H_{1_G}. On pose :

$$w_l = \prod_{u \in B_l} u^{\mathsf{k}}.$$

Soit n le nombre d'états de $\mathcal{B}$. Pour tout u appartenant à H_g, le mot $(w_l)^n.u$ appartient aussi à H_g, donc à $\mathcal{L}$. Il est accepté par $\mathcal{B}$. Ce calcul peut se décomposer de la façon suivante:



L'état r_l respecte i) et ii) pour v de longueur inférieure à l. Lorsqu'on fait tendre l vers l'infini, on obtient une suite d'états $(r_l)_{l \in \mathbb{N}}$ dans laquelle un état r de $\mathcal{B}$ apparaît infiniment souvent. Cet état vérifie les conditions i) et ii). □

THÉORÈME 4.2 *Si le monoïde syntaxique d'un langage $\mathcal{L}$ est un groupe, l'automate universel contient un sous-automate reconnaissant $\mathcal{L}$ et d'enlacement minimal. Plus précisément, il existe des pelotes de l'automate universel dont l'union forme un automate qui reconnaît $\mathcal{L}$ avec un enlacement minimum.*

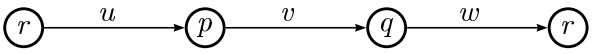
Démonstration. Soit $\mathcal{B}$ un automate d'enlacement minimal reconnaissant $\mathcal{L}$. Soit g un élément du groupe syntaxique qui correspond à une partie H_g de $\mathcal{L}$. Il existe un état r qui vérifie les deux conditions du lemme. Soit φ un morphisme de $\mathcal{B}$ dans $\mathcal{U}_{\mathcal{L}}$ et $s = r\varphi$.

$H_{1_G} \cap \mathsf{Past}_{\mathcal{B}}(r) \neq \emptyset$, donc, comme $\mathsf{Past}_{\mathcal{B}}(r) \subseteq \mathsf{Past}_{\mathcal{U}_{\mathcal{L}}}(s)$, et que le passé de s est reconnu par G, $H_{1_G} \subseteq \mathsf{Past}_{\mathcal{U}_{\mathcal{L}}}(s)$, donc s est initial. De même, H_g est inclus dans le futur de s. Considérons la pelote $\mathcal{P}$ de $\mathcal{U}_{\mathcal{L}}$ qui contient s. Il existe un mot v de A^* tel que, pour tout mot u de H_g, le mot $u.v$ appartient à H_{1_G}. Donc il existe une boucle autour de s étiquetée par $u.v$. Soit t l'état de cette boucle atteint à partir de s en lisant u. Le mot u appartient

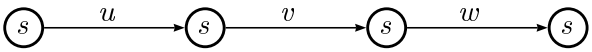
6. qui étiquette donc une boucle autour de i (état initial de $\mathcal{A}$)

à $\mathsf{Fut}_{\mathcal{U}_\mathcal{L}}(s)$, donc le mot vide appartient à $\mathsf{Fut}_{\mathcal{U}_\mathcal{L}}(t)$, qui est terminal. La pelote $\mathcal{P}$ accepte donc (au moins) tous les mots de H_g.

Montrons que $\mathcal{P}$ est d'enlacement inférieur à $\mathcal{B}$. Tout chemin de $\mathcal{P}$ est contenu dans une boucle autour de s qui, quitte à tourner plusieurs fois, est étiquetée par un mot v de H_{1_G}. Il existe donc u dans H_{1_G} et w dans A^* tels que $u.v.w$ étiquette une boucle autour de r dans $\mathcal{B}$:



L'image de cette boucle par φ est une boucle autour de s étiquetée par $u.v.w$. Comme u est dans H_{1_G}, il étiquette une boucle autour de s, de même que v :



Cette boucle est celle qui contient le chemin de départ, puisque $\mathcal{P}$ est déterministe. Tout chemin de $\mathcal{P}$ a donc une image inverse dans $\mathcal{B}$. L'enlacement de $\mathcal{P}$ est donc inférieur ou égal à celui de $\mathcal{B}$.

On peut faire ce raisonnement pour tout élément de G image d'une partie de $\mathcal{L}$. On obtient ainsi une union de pelotes d'enlacement inférieur à celui de $\mathcal{B}$. Elles forment donc un automate d'enlacement minimal qui reconnaît $\mathcal{L}$. □

— ∘ —

Corollaire 4.7 *Soit $\mathcal{A}$ l'automate minimal d'un langage à groupe. Si $\mathcal{A}$ a un seul état final, il est d'enlacement minimal.*

Démonstration. D'après le corollaire 3.7, dans ce cas, l'automate universel est isomorphe à l'automate minimal. La seule pelote de l'automate universel est donc $\mathcal{A}$ lui-même qui est donc d'enlacement minimal. □

Une conséquence triviale de ce corollaire est le résultat de F. Dejean et M.-P. Schützenberger [21] selon lequel, pour tout entier n, il existe des langages rationnels de hauteur d'étoile n. Il suffit de considérer les langages à groupe $\{u \in \{a,b\}^* \mid |u|_a = |u|_b \mod 2^n\}$, dont l'automate minimal a un seul état final et est d'enlacement n.

L'algorithme qui permet de trouver un sous-automate d'enlacement minimum ne nécessite pas une énumération de tous les sous-automates de l'automate universel. Il suffit de calculer l'enlacement des pelotes et de considérer les automates successifs formés par l'adjonction des pelotes en commençant par celles d'enlacement minimum, jusqu'à ce qu'on obtienne un automate qui reconnaisse le langage. De plus, si l'automate universel est construit à partir de l'automate minimal, il est facile de savoir quelle est la part du langage reconnue par chaque pelote.

Proposition 4.8 *Soit $\mathcal{A} = \langle Q,A,E,i,T \rangle$ l'automate minimal d'un langage à groupe $\mathcal{L}$. Soit $\mathcal{P}$ une pelote de l'automate universel de $\mathcal{L}$ construit à partir de $\mathcal{A}$. Les états de $\mathcal{P}$*

sont indicés par des sous-ensembles de Q. Le passé d'un état X dans $\mathcal{P}$ est égal à l'union des passés des états p dans $\mathcal{A}$, pour p dans X :

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

Démonstration.

$$u \in \mathsf{Past}_{\mathcal{P}}(X) \Leftrightarrow Y = u \cdot X \text{ initial} \Leftrightarrow i \cdot u \in X \Leftrightarrow u \in \bigcup_{p \in X} \mathsf{Past}_{\mathcal{A}}(p)$$

□

Le langage accepté par une pelote étant l'union des passés des états terminaux de la pelote, il est très simple de savoir si une union de pelotes réalise ou non tout le langage.

Exemple 15.2 L'enlacement de l'automate minimal de $\mathcal{L}_{g_2}$ (figure 3 page 83) est égal à 3. Dans cet exemple, l'automate minimal est isomorphe au graphe de Cayley du groupe. Chaque état correspond donc à un élément particulier du groupe et son passé à l'image inverse de cet élément par le morphisme canonique de A^* dans le groupe. On notera par exemple H_t l'image inverse de l'élément du groupe correspondant à l'état t. L'examen de l'automate universel donne le tableau suivant, dans lequel chaque pelote est désignée par un de ses états :

Pelote	Enlacement	langage reconnu
$\{p\}$	3	$H_q \cup H_s \cup H_t \cup H_u$
$\{p,s\}$	2	$H_q \cup H_t$
$\{p,t\}$	2	$H_q \cup H_u$
$\{p,u\}$	2	$H_q \cup H_s$
$\{p,q,r\}$	2	$H_s \cup H_t \cup H_u$
$\{p,q,r,s\}$	3	$H_q \cup H_s \cup H_t \cup H_u$

On voit qu'il suffit de prendre, par exemple, les pelotes des états $\{p,q,r\}$ et $\{p,s\}$ pour former un automate d'enlacement 2 qui reconnaît le langage (figure 4). D'après le théorème précédent, cet automate est d'enlacement minimal pour le langage. La hauteur d'étoile du langage $\mathcal{L}_{g_2}$ est donc 2.

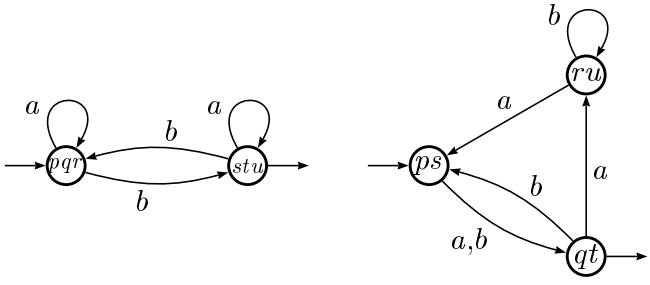


Fig. 4 – *Un automate d'enlacement minimum pour $\mathcal{L}_{g_2}$*

— ◦ —

3 Hauteur d'étoile des langages réversibles

Les langages réversibles étant la généralisation naturelle des langages à groupe, il paraît naturel d'essayer d'y étendre la méthode de R. McNaughton.

Les langages considérés par R. Cohen [17] et K. Hashiguchi et N. Honda [31] font partie d'une classe plus restreinte que celle des langages réversibles, celle des langages reset-free. Un langage est **reset-free** si son automate minimal est réversible. On voit que le problème de l'appartenance à cette classe est trivialement réglé de même que la construction d'un automate réversible pour un de ces langages.

R. Cohen montre que si l'automate minimal d'un tel langage n'a qu'un état minimal, l'automate minimal est d'enlacement minimal pour le langage. C'est la généralisation du corollaire 4.7 au langages reset-free. K. Hashiguchi et N. Honda étendent quant à eux la méthode de R. McNaughton. Il faut souligner ici une différence profonde entre les algorithmes de R. McNaughton, N. Hashiguchi et N. Honda et ceux présentés ici. Dans le premier cas, à partir d'un automate connu ayant les bonnes propriétés (de groupe ou de réversibilité), on construit des automates parmi lesquels on assure qu'on en trouvera d'enlacement minimum dont l'union reconnaît le langage. Dans le second cas, on construit l'automate universel à partir de l'automate minimal, dans lequel on cherche des sous-automates ayant les propriétés voulues. On peut montrer que cet automate universel respecte certaines propriétés à partir d'un automate qu'on ne sait *a priori* pas construire.

Avant de passer à la démonstration du résultat, on présente deux lemmes.

Définition 4.11 *On dit qu'un mot u est un* ***mot idempotent pour un automate $\mathcal{A}$*** *si son image dans le monoïde de transition de $\mathcal{A}$ est un idempotent.*

Lemme 4.9 *Soit $\mathcal{A}$ un quipu réversible. Soit w un idempotent pour $\mathcal{A}$. Quels que soient les mots u et v de A^*, si $u.w.v$ est accepté par une branche de $\mathcal{A}$, $u.v$ est accepté par la même branche.*

Démonstration. Ceci vient du fait que dans un automate réversible, un idempotent ne peut étiqueter qu'une boucle. En effet, si w est un idempotent pour $\mathcal{A}$ qui étiquette un chemin de p à q, il étiquette aussi une boucle autour de q et comme l'automate est réversible, on obtient $p = q$. Par conséquent, de tout calcul de $\mathcal{A}$ étiqueté par $u.w.v$, avec w idempotent pour $\mathcal{A}$, on peut extraire un calcul étiqueté par $u.v$. □

Le monoïde de transition qui reconnaît un langage réversible n'est pas caractéristique d'un tel langage, puisqu'il faut, en plus, respecter certaines propriétés de fermeture. De fait, contrairement à ce que nous avons fait pour les groupes, on ne peut pas effectuer une partition du langage réversible par rapport aux élément du monoïde syntaxique de manière à obtenir des langages réversibles. Par exemple, on ne peut pas séparer le langage $a^* + b^*$ en deux langages réversibles disjoints unions d'images réciproques d'éléments de son monoïde syntaxique. En fait, on va décomposer le langage en un ensemble de langages réversibles (non nécessairement disjoints); chacun de ces langages correspondant à une corde d'un quipu réversible minimal. Chaque corde compte un certain nombre d'états-clés qui sont

les états d'entrée et de sortie des différentes pelotes. Tout chemin réussi dans une corde emprunte tous ces états.

DÉFINITION 4.12 *Soit $\mathcal{D} = \langle Q,A,E,i,t \rangle$ une corde. Les pelotes de $\mathcal{D}$ sont totalement ordonnées :*

$$\mathcal{P} < \mathcal{P}' \Leftrightarrow \forall p \in \mathcal{P}, \forall p' \in \mathcal{P}, \exists u \in A^*,\ p \cdot u = p'.$$

Soit $\mathcal{P}_1 < \mathcal{P}_2 < \ldots < \mathcal{P}_k$ les pelotes de $\mathcal{D}$. Pour tout j de $[1;k]$, l'état p_j (resp. q_j) est le seul état de $\mathcal{P}_j$ qui soit état d'arrivée (resp. de départ) d'une transition extérieure à $\mathcal{P}_j$. Si l'état i appartient à $\mathcal{P}_1$, $p_1 = i$; de même, si l'état t appartient à $\mathcal{P}_k$, $q_k = t$. Le $2k$-uplet $(p_1,q_1,p_2,\ldots,p_k,q_k)$ est appelé jalon de $\mathcal{D}$. Tout mot u accepté par $\mathcal{D}$ se décompose de façon unique en $u = v_0.u_1.v_1.u_2 \ldots u_k.v_k$, avec

$$\begin{aligned} i \cdot v_0 &= p_1 & \qquad p_j \cdot u_j &= q_j,\ \forall j \in [1;k] \\ q_k \cdot v_k &= t & \qquad q_j \cdot v_j &= p_{j+1},\ \forall j \in [1;k+1] \end{aligned}$$

Le lemme suivant est l'exacte généralisation du lemme 4.6.

LEMME 4.10 *Soit $\mathcal{L}$ un langage réversible et $\mathcal{A}$ un quipu réversible minimal qui reconnaît le langage $\mathcal{L}$ et $\mathcal{B}$ un automate équivalent. Soit $\mathcal{D}$ une corde de $\mathcal{A}$, $(p_1,q_1,\ldots,q_k)$ le jalon de $\mathcal{D}$ et u un mot accepté par $\mathcal{D}$, dont la décomposition selon $\mathcal{D}$ est $v_0,u_1,v_1,\ldots,v_k$. Il existe k idempotents pour $\mathcal{A}$, $w_1, \ldots, w_k$ et k états $r_1, \ldots, r_k$ de $\mathcal{B}$ tels que, pour tout i de $[1;k]$:*

i) *Le mot w_i étiquette une boucle autour de l'état p_i.*

ii) *Il existe des entiers h_i et l_i tels que*

$$\begin{aligned} v_0.w_1^{h_1} &\in \mathsf{Past}_{\mathcal{B}}(r_1), \\ w_k^{l_k}.u_k.v_k &\in \mathsf{Fut}_{\mathcal{B}}(r_k), \\ w_i^{l_i}.u_i.v_i.w_{i+1}^{h_{i+1}} &\in \mathsf{Trans}_{\mathcal{B}}(r_i,r_{i+1}),\ \textit{pour } i < k. \end{aligned}$$

iii) *Si un mot y étiquette une boucle autour de p_i, il existe un mot x idempotent et un mot z tels que $x.y.z$ étiquette une boucle autour de r_i.*

Démonstration. La situation décrite dans le lemme est représentée figure 5.
Soit l un entier positif. Pour tout i, on pose $E_i = \{u \in A^* \mid p_i \cdot u = p_i \text{ et } |u| \leqslant l\}$. Il existe un entier k tel que pour tout mot u de A^*, le mot u^{k} est un idempotent pour $\mathcal{A}$. On forme le mot

$$w_i = \prod_{u \in E_i} u^{\mathsf{k}}$$

On choisit un ordre arbitraire sur les mots de E_i pour effectuer ce produit. Le mot w_i étiquette une boucle autour de p_i. Soit n le nombre d'états de $\mathcal{B}$. On considère le mot

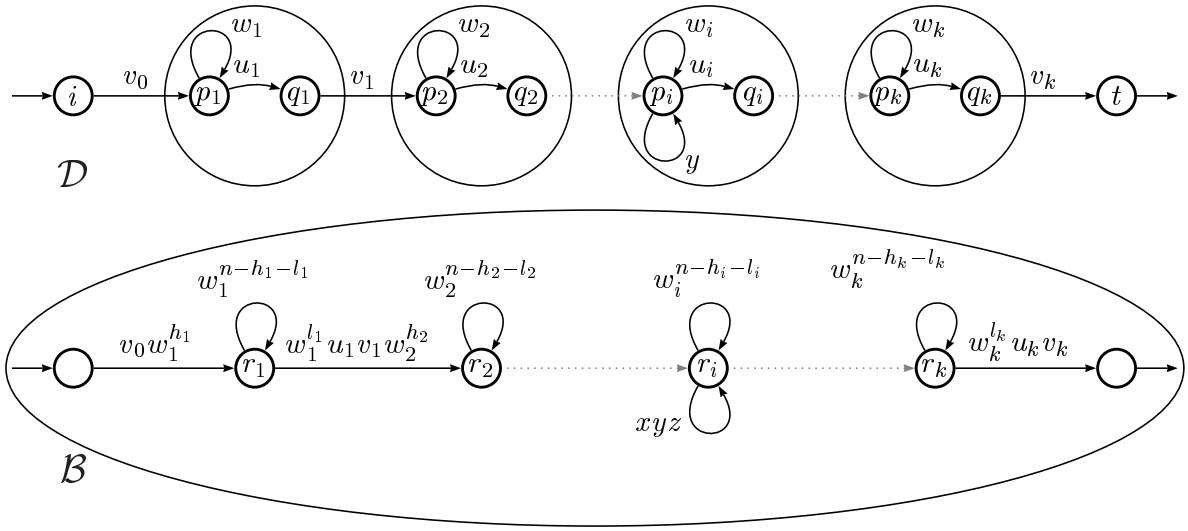
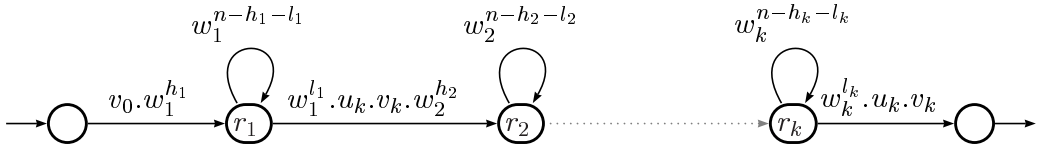


FIG. 5 – *Une corde, ses jalons, et sa trace dans un automate qui accepte le langage.*

$v_0.w_1^n.u_1.v_1 \dots w_k.u_k.v_k$, qui est accepté par $\mathcal{D}$, donc par $\mathcal{B}$. Par le lemme des tiroirs, un calcul de $\mathcal{B}$ étiqueté par ce mot peut se décomposer de la manière suivante :



Les états r_i ainsi définis vérifient i) et ii). Tout mot y de longueur inférieure à l qui fait une boucle autour de p_i, appartient à E_i. Donc $w_i = x_0.y.z$, avec x idempotent pour $\mathcal{A}$, et $w_i^{n-h_i-l_i} = w_i^{n-h_i-l_i-1}.x_0.y.z$. En posant $x = w_i^{n-h_i-l_i-1}.x_0$, on montre iii) pour les mots de longueur inférieure à l.

Pour tout l, on peut trouver des états $(r_1,r_2,\dots,r_k)$ qui vérifient i), ii) et iii) pour les mots de longueur inférieure à l. Quand on fait tendre l vers l'infini, le nombre de k-uplets d'états différents étant fini, il y en a un qui apparaît infiniment souvent. Celui-ci vérifie le lemme pour des mots de longueur arbitrairement grande. □

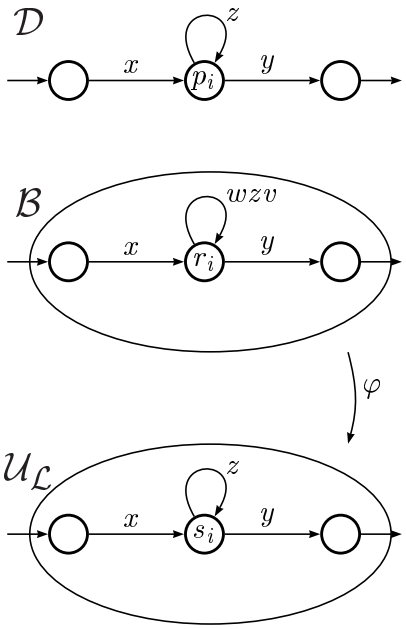
THÉORÈME 4.3 *Soit $\mathcal{L}$ un langage réversible. Il existe un sous-automate quasi-réversible de l'automate universel de $\mathcal{L}$ qui reconnaît $\mathcal{L}$ et qui est d'enlacement minimal.*

Démonstration. La preuve du théorème se fait en plusieurs étapes. D'abord, on pose les outils utilisés.

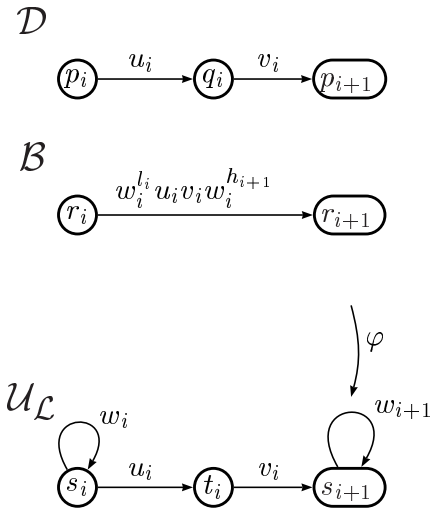
Soit $\mathcal{A}$ un quipu réversible minimal qui reconnaît le langage $\mathcal{L}$ et $\mathcal{B}$ un automate qui reconnaît ce même langage et qui est d'enlacement minimum. $\mathcal{B}$ a n états. On considère une corde $\mathcal{D}$ de $\mathcal{A}$ et u un mot accepté uniquement par cette corde. Soit $(p_1,q_1,p_2,q_2,\dots,q_k)$ le jalon de $\mathcal{D}$ et $(v_0,u_1,u_2,\dots,u_k,v_k)$ la décomposition de u par rapport à $\mathcal{D}$.
Soit $r_1,\dots,r_k$ et $w_1,\dots,w_k$ les états de $\mathcal{B}$ et les idempotents pour $\mathcal{A}$ donnés par le lemme 4.10 appliqué à $\mathcal{D}$ et u. La situation est donc celle schématisée figure 5.
Soit φ un morphisme de $\mathcal{B}$ dans $\mathcal{U}_\mathcal{L}$. On fixe un entier i de $[1;k]$.

On pose $s_i = r_i\varphi$ et $\mathcal{P}_i$ est la pelote de $\mathcal{U}_\mathcal{L}$ qui contient s_i. Il existe des entiers $h_1, \ldots, h_i$ (*resp.* $l_i, \ldots, l_k$) tels que le mot $x = u_0.w_1^{h_1}.v_1.u_1 \ldots u_{i-1}.w_i^{h_i}$ (*resp.* le mot $y = w_i^{l_i}.v_i.u_i \ldots w_k^{l_k}.v_k.u_k$) appartient au passé (*resp.* au futur) de r_i et donc de s_i.

On montre que φ est un morphisme conforme sur $\mathcal{P}_i$. Soit $\mathcal{C}$ un chemin de $\mathcal{P}_i$. On peut supposer, quitte à le prolonger, que $\mathcal{C}$ est une boucle autour de s_i et, quitte à tourner suffisament de fois autour de s_i, que $\mathcal{C}$ est étiqueté par un idempotent z pour $\mathcal{A}$. Le mot $x.z.y$ appartient au langage. D'après le lemme 4.9, toute corde de $\mathcal{A}$ qui accepte ce mot accepte aussi le mot u, donc ce mot est accepté par la seule corde $\mathcal{D}$. Comme la corde est réversible, le début de calcul étiqueté par x mène en p_i et la fin de calcul étiquetée par y vient de p_i. Le mot z étiquette donc une boucle autour de p_i. Donc, d'après les propriétés de l'état r_i, il existe un mot w idempotent pour $\mathcal{A}$ et un mot v tels que $w.z.v$ étiquette une boucle autour de r_i. L'image de cette boucle par φ est une boucle autour de s_i. Comme w est un idempotent (dans le monoïde de transition donc *a fortiori* dans le monoïde syntaxique), d'après la proposition 3.12, il étiquette une boucle autour de s_i. Comme les pelotes sont déterministes, la boucle $\mathcal{C}$ autour de s_i étiquetée par z est bien image de la portion de boucle autour de r_i étiquetée par z. Le morphisme φ est donc conforme sur $\mathcal{P}_i$ dont l'enlacement est, par conséquence, inférieur à celui de $\mathcal{B}$.



On montre qu'on peut relier deux pelotes $\mathcal{P}_i$ et $\mathcal{P}_{i+1}$ sans former de nouvelle boucle. Le mot u_i peut être prolongé de manière à étiqueter une boucle autour de p_i dans $\mathcal{D}$, donc il existe un idempotent x pour $\mathcal{A}$ et un mot z tels que $x.u_i.z$ étiquette un mot autour de r_i, donc de s_i et comme x est un idempotent, il existe un chemin étiqueté par u_i dans $\mathcal{P}_i$ qui part de s_i. Soit t_i son état d'arrivée.
De même, en utilisant la proposition 3.12, les mots w_i et w_{i+1} étiquettent respectivement des boucles autour de s_i et s_{i+1}.



Il existe deux entiers l_i et h_{i+1} tels que $w_i^{l_i}.u_i.v_i.w_{i+1}^{h_{i+1}}$ appartienne à $\mathsf{Trans}_\mathcal{B}(r_i,r_{i+1})$ donc à $\mathsf{Trans}_{\mathcal{U}_\mathcal{L}}(s_i,s_{i+1})$. Posons $s_i = (L_i,R_i)$ et $t_i = (L'_i,R'_i)$. Par définition de l'automate

universel :

$$L_i.w_i^{l_i}.u_i.v_i.w_{i+1}^{h_{i+1}} \subseteq L_{i+1}$$

Soit μ le morphisme syntaxique de $\mathcal{L}$. D'après la proposition 3.11, $(L_i.w_i^{l_i})\mu = L_i\mu$ et $(L_i.u_i)\mu = L'_i\mu$. Donc il existe un chemin étiqueté par $v_i.w_{i+1}^{h_{i+1}}$ entre t_i et s_{i+1}. On a donc :

$$R'_i \subseteq v_i.w_{i+1}^{h_{i+1}}.R_{i+1}$$

De même, $(w_{i+1}^{h_{i+1}}.R_{i+1})\mu = R_{i+1}\mu$, donc il existe un chemin étiqueté par v_i entre t_i et s_{i+1}. Ceci est vrai pour tout i de $[1; k-1]$. On montre de même qu'il existe un chemin d'un état initial à s_1 étiqueté par v_0 et un chemin de s_k à un état terminal étiqueté par v_k.

Si un des états intérieurs d'un de ces chemins étiqueté par v_i appartient à une pelote, on peut factoriser v_i en $x_i.y_i$ et il existe un idempotent w pour $\mathcal{A}$ tel que

$$v_0u_1v_1 \dots u_i.x_i.w.y_i.u_{i+1}.v_{i+1} \dots u_k.v_k \in \mathcal{L}.$$

Il ne peut être accepté que par $\mathcal{D}$ (lemme 4.9), ce qui contredit qu'il n'existe pas de boucle entre q_i et p_{i+1}.

On montre que le sous-automate de $\mathcal{U}_\mathcal{L}$ ainsi formé accepte tous les mots de $\mathcal{D}$. Tout mot accepté par $\mathcal{D}$ est de la forme $v_0.u'_1.v_1 \dots u'_k.v_k$, avec $p_i \cdot u'_i = q_i$. Il suffit de montrer que u'_i appartient à $\mathsf{Trans}_{\mathcal{U}_\mathcal{L}}(s_i,t_i)$. Il existe un mot w_i tel que $u'_i.w_i$ est un idempotent et $u'_i.w_i$ et $u_i.w_i$ étiquettent des boucles autour de p_i. De même que précédemment, ces mots étiquettent des boucles autour de s_i et comme la pelote est co-déterministe, le chemin partant de s_i étiqueté par u'_i arrive dans le même état que celui étiqueté par u_i, c'est-à-dire t_i.

On montre que ce sous-automate est réversible. Les pelotes de l'automate universel sont réversibles. Entre chaque pelote, l'automate ne comporte qu'un seul chemin, par construction. S'il existe une lettre a qui étiquette deux transitions arrivant en s_i, cette lettre est la dernière de v_i et on montre qu'il existe une boucle autour de p_i dont la dernière étiquette est a, ce qui contredit la réversibilité de $\mathcal{D}$. Donc le sous-automate est co-déterministe; il est de même déterministe.

Conclusion. Pour chaque corde $\mathcal{D}$ de $\mathcal{A}$, on peut montrer qu'il existe un sous-automate de l'automate universel d'enlacement inférieur à la hauteur d'étoile du langage, et qui accepte au moins le langage reconnu par cette corde. La superposition de tous ces automates dans l'automate universel forme un automate qui reconnaît le langage $\mathcal{L}$. De plus, on a vu que pour chaque état de chacun des sous-automates, si l'état appartient à une pelote, la pelote en intégralité appartient au sous-automate. Chaque pelote de la « superposition » est donc contenue dans un des sous-automates. L'enlacement de l'automate obtenu est donc égal au maximum des enlacement obtenus en considérant les branches séparément, c'est-à-dire inférieur ou égal à l'enlacement de $\mathcal{B}$ qui est minimum par hypothèse. Si deux

transitions étiquetées par la même lettre arrivent dans un même état, une d'entre elle n'appartient à aucune pelote. Le sous-automate dont elle provient est réversible et contient les pelotes de $\mathcal{U}_\mathcal{L}$ qu'il intersecte en intégralité; la seconde n'appartient donc à aucune pelote non plus. De même pour les transitions partant d'un même état. L'automate est donc quasi-réversible. Ce raisonnement est similaire à celui de la preuve du théorème 3.4. Il existe donc un sous-automate quasi-réversible de l'automate universel qui est d'enlacement minimum pour le langage. □

Le résultat de R. Cohen [17] est alors un corollaire du théorème.

Corollaire 4.11 *Soit $\mathcal{L}$ un langage réversible dont l'automate minimal est réversible et n'a qu'un état final. Alors, l'automate minimal de $\mathcal{L}$ est d'enlacement minimum.*

La raison est exactement la même que pour les groupes. On a en effet vu que dans ce cas, l'automate universel est isomorphe à l'automate minimal (proposition 3.10). Par minimalité, le seul sous-automate de l'automate universel qui accepte le langage est donc l'automate universel lui-même.

Exemple 12.4 Cet algorithme, appliqué à l'automate universel de $\mathcal{L}_{r_1}$, permet d'obtenir l'automate présenté figure 6. Cet automate est quasi-réversible et contient un sous-

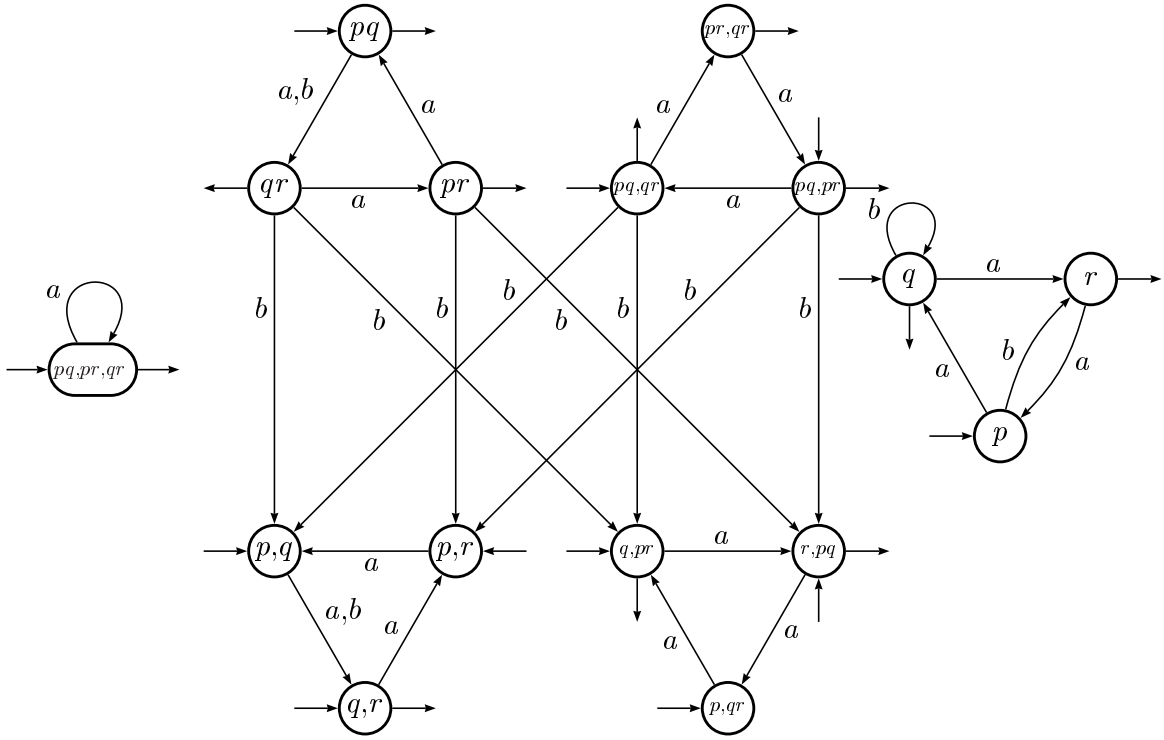


Fig. 6 – *Un automate quasi-réversible pour $\mathcal{L}_{r_1}$.*

automate d'enlacement minimal pour le langage. De plus, on sait qu'il existe un tel sous-automate qui recouvre les pelotes qu'il intersecte. On va donc considérer des sous-automates formés de composantes fortement connexes « entières ». Or, le mot b^4 de $\mathcal{L}_{r_1}$

n'est reconnu que par la composante formée des états p, q et r. L'automate d'enlacement minimal contient donc au moins cette pelote. Celle-ci reconnaît tout le langage; elle forme donc un automate quasi-réversible (ici, même, réversible) d'enlacement minimal (égal à 2) qui reconnaît le langage. La hauteur d'étoile du langage $\mathcal{L}_{r_1}$ est donc égale à 2.

— ∘ —

4 Automate universel et hauteur d'étoile

Rien n'empêche d'exprimer les résultats obtenus pour les langages réversibles ou les langages à groupe dans le cas général.

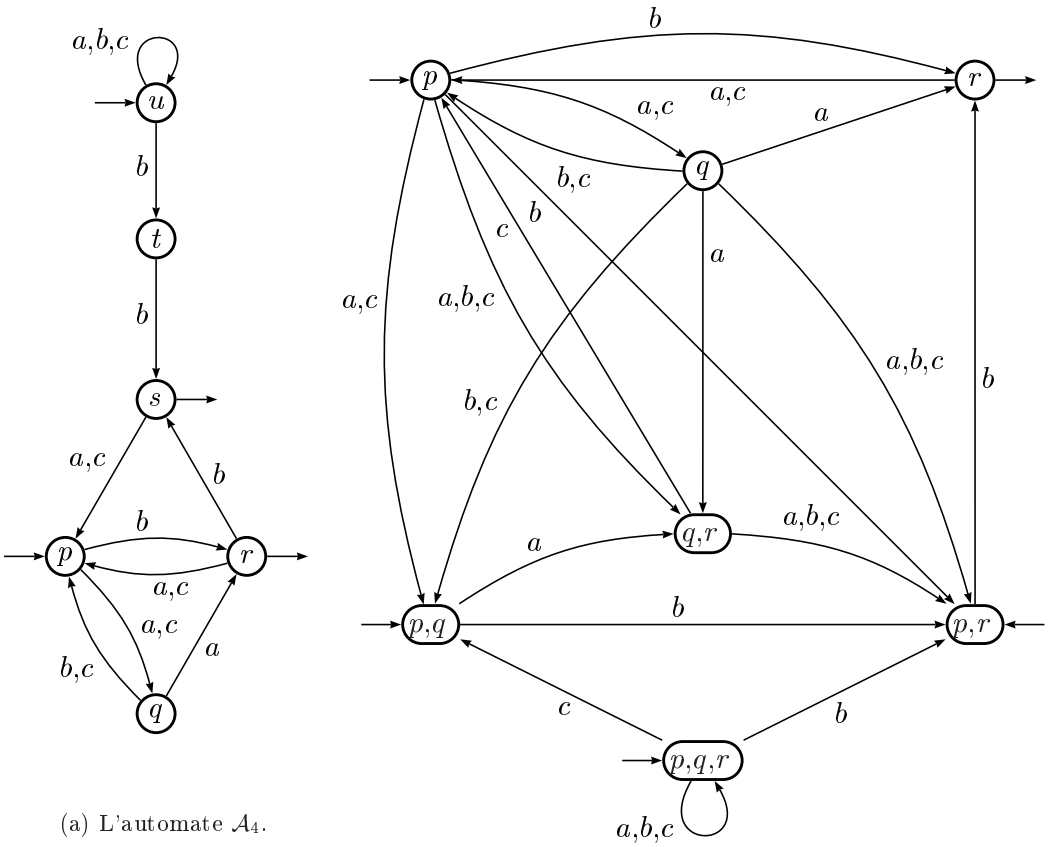
Conjecture 4.1 *Soit $\mathcal{L}$ un langage rationnel de A^* et $\mathcal{U}_\mathcal{L}$ l'automate universel de $\mathcal{L}$. L'automate $\mathcal{U}_\mathcal{L}$ contient un sous-automate qui reconnaît $\mathcal{L}$ avec un enlacement minimum.*

Jusqu'à présent, rien n'est venu contredire cette conjecture. Toutefois, il semble difficile d'adapter les techniques employées précédemment pour résoudre ce problème. L'exemple ci-dessous reflète les difficultés qu'on peut rencontrer.

Exemple 10.3 Si l'on regarde attentivement l'automate minimal de $\mathcal{L}_4$ (voir figure 11 page 60), on s'aperçoit que tout facteur bb mène dans l'état 3 et que le langage $\mathcal{L}_4$ est donc reconnu par l'automate $\mathcal{A}_4$ présenté figure 7 a), dont l'enlacement est égal à 1. La hauteur d'étoile de $\mathcal{L}_4$ est donc 1. L'automate universel de $\mathcal{L}_4$ a un seul état terminal. Les états r et s de $\mathcal{A}_4$ ont donc la même image dans $\mathcal{U}_{\mathcal{L}_4}$ L'image par n'importe quel morphisme de cet automate dans l'automate universel de $\mathcal{L}_4$ est donc d'enlacement 2. On peut trouver dans ce dernier un sous-automate d'enlacement 1 qui accepte tous les mots du langage : l'automate de la figure 7 b) est d'enlacement 1; si on supprime l'état p, la composante fortement connexe composée de tous les états hormi $\{p,q,r\}$ devient acyclique.

Cet exemple pointe la difficulté de prouver le résultat en général. Comment utiliser le fait qu'il existe un automate d'enlacement minimum pour prouver qu'il en existe un dans l'automate universel ? On voit ici que le premier peut n'avoir aucun rapport avec le second.

— ∘ —



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

(b) Un sous automate de $\mathcal{U}_{\mathcal{L}_4}$.

FIG. 7 – *Deux automates d'enlacement minimum pour* $\mathcal{L}_4$.

Annexe B

Nous présentons ici l'algorithme d'Hashiguchi tel qu'on peut le trouver dans [33]. Afin de simplifier les choses, certaines parties de l'algorithme, notamment la décision de la hauteur d'étoile relative nulle sont présentées comme des boîtes noires.

Tout d'abord, une petite définition :

DÉFINITION 6.11 *Soit $\mathcal{L}$ un langage rationnel sur A^* et $\mathcal{C}$ une famille (finie) de langages rationnels sur A^*. L'ensemble* $\mathsf{RExp}\mathcal{C}$ *des expressions rationnelles sur $\mathcal{C}$ est défini de façon similaire à* $\mathsf{RExp}A$*. En revanche, l'interprétation d'une expression de* $\mathsf{RExp}\mathcal{C}$ *est induite par l'interprétation d'un élément $\mathcal{L}_i$ de $\mathcal{C}$ qui est le langage $\mathcal{L}_i$ sur A^*. La* ***hauteur d'étoile relative*** *de $\mathcal{L}$ par rapport à $\mathcal{C}$, noté $\mathsf{h}_{\mathcal{C}}(\mathcal{L})$, est la hauteur d'étoile minimale des expressions de* $\mathsf{RExp}\mathcal{C}$ *qui représentent $\mathcal{L}$. Si $\mathcal{L}$ n'appartient pas à* $\operatorname{Rat}\mathcal{C}^*$*, alors $\mathsf{h}_{\mathcal{C}}(\mathcal{L}) = \infty$.*

Par exemple, la hauteur d'étoile relative de $\mathcal{L}$ par rapport à $\{\mathcal{L}\}$ est nulle.

Présentons tout d'abord l'algorithme pour le calcul de la hauteur d'étoile relative :

Soit $\mathcal{A}$ un automate minimal reconnaissant un langage $\mathcal{L}$ et Q l'ensemble des états de $\mathcal{L}$. Soit M le monoïde syntaxique de $\mathcal{L}$. Soit $\mathcal{C} = \{\mathcal{L}_i \mid i \in I\}$ une famille finie de langages. Chaque langage $\mathcal{L}_i$ est reconnu par un automate $\mathcal{B}_i$ dont les états forment un ensemble Q_i.

Si i est l'état initial de $\mathcal{A}$, on définit $\mathcal{A}_{\mathcal{C}}$ l'automate déterministe sur l'alphabet $\mathcal{C}$ qui représente l'orbite de $\{i\}$ sous l'action des éléments de $\mathcal{C}$. (Les éléments de $\mathcal{C}$ agissent canoniquement sur les sous-ensembles d'états de $\mathcal{A}$.)

On définit une cascade de constantes :

$$
\begin{aligned}
g_1 &= (4^{\mathsf{Card}(M)} + 1)(\mathsf{enl}(\mathcal{A}_{\mathcal{C}}) + 1) \\
g_2 &= \binom{g_1}{(\mathsf{enl}(\mathcal{A}_{\mathcal{C}}) + 1)} \left(\sum_{i \in I} \mathsf{Card}(Q_i) \right) \\
g_3 &= 16^{g_2^2} \\
g_4 &= (g_3 (\mathsf{Card}(Q) + 1)^{\mathsf{Card}(Q)}\ \mathsf{Card}(M))^2 \\
g_5 &= (g_4 + \mathsf{Card}(Q) + 2)^{4\,\mathsf{Card}(Q)}
\end{aligned}
$$

L'algorithme pour le calcul de la hauteur d'étoile relative est le suivant :

1. Vérifier si $\mathsf{h}_{\mathcal{C}}(\mathcal{L}) = \infty$.
2. Vérifier si $\mathsf{h}_{\mathcal{C}}(\mathcal{L}) = 0$.

3. Calculer g_5 et former

$$\mathcal{C}' = \mathcal{C} \cup \bigcup \{ \lfloor (W_1 + W_2 \ldots + W_n)^\star \rfloor \mid n \geqslant 1 \},$$

où chaque W_i est un mot non vide de longueur inférieure à g_5 sur l'alphabet $\mathcal{C}$. Alors $\mathsf{h}_{\mathcal{C}}(\mathcal{L}) = \mathsf{h}_{\mathcal{C}'}(\mathcal{L}) + 1$.

En fait, pour le calcul de la hauteur d'étoile (absolue), on verra que le point 1 est inutile. L'algorithme de calcul de la hauteur d'étoile s'appuie sur le précédent. Il est initié par :

1. Si $\mathcal{L}$ est un langage fini, $\mathsf{h}(\mathcal{L}) = 0$.
2. Soit $g = 16n(n+2)(\mathsf{enl}(\mathcal{A})n(n+2)+1)$, avec $n = \mathsf{Card}(M)$. On pose

$$\mathcal{C} = \{\{a\} \mid a \in A\} \cup \bigcup \{ \lfloor (w_1 + w_2 \ldots + w_n)^\star \rfloor \mid n \geqslant 1, w_i \in A^{\leqslant g} \}.$$

Calculer $\mathsf{h}(\mathcal{L}) = \mathsf{h}_{\mathcal{C}}(\mathcal{L}) + 1$.

Pour illustrer la complexité de cet algorithme, considérons le langage $\mathcal{L}_{r_1}$. Ce langage est un langage réversible. Nous avons calculé sa hauteur d'étoile (égale à 2) dans l'exemple 12 page 119. Le monoïde syntaxique de ce langage est aussi le monoïde de transition de l'automate présenté page 71, il compte trente-et-un éléments (toutes les fonctions injectives d'un ensemble de trois éléments dans lui-même, sauf les trois transpositions); l'enlacement de son automate minimal est égal à 2. On obtient les valeurs suivantes :

$$g = 33505296, \qquad \mathsf{Card}(A^{\leqslant g}) > 2^{33505296}, \qquad \mathsf{Card}(\mathcal{C}) > 2^{2^{33505296}}$$

On peut maintenant appliquer l'algorithme de calcul de la hauteur d'étoile relative. L'automate $\mathcal{A}_{\mathcal{C}}$ a au moins un enlacement égal à 2, puisque $\mathcal{A}$ en est un sous-automate.

$$g_2 > 2^{2^{33505296}}, \qquad g_5 > g_4 > g_3 = 16^{g_2^2} > 2^{2^{2^{33505296}}}$$

On obtient donc :

$$\mathsf{Card}(\mathcal{C}^{\leqslant g_5}) > \left(2^{2^{33505296}}\right)^{2^{2^{2^{33505296}}}}, \qquad \mathsf{Card}(\mathcal{C}') > 2^{\left(2^{2^{33505296}}\right)^{2^{2^{2^{33505296}}}}} > 2^{2^{2^{2^{2^{2^2}}}}}$$

Remarquons que dans la manipulation de ce type de nombres, le seul ordre de grandeur qui reste est la hauteur de la tour d'exponentielle et que les valeurs des nombres dans cette tour ont une signification qui décroît exponentiellement à mesure qu'on s'éloigne du haut. La dernière inégalité est par exemple une minoration brutale, (on minore 33505296 par 65536 et $\left(2^{2^{33505296}}\right)$ par 2), mais si on remplace le 2 du haut par 2,3, elle devient fausse. En revanche, on peut remplacer le 2 du bas par un milliard sans la contredire.

Pour donner un ordre de grandeur, le nombre estimé de particules de l'univers (10^{80}) ou le nombre de cycles qu'aurait effectué un processeur de fréquence 10 GHz depuis la naissance supposée de l'univers (5 $\overline{\text{M}}$A) est très largement majoré par une tour de hauteur 5 (2^{65536}). Ces grandeurs sont négligeables devant une tour de hauteur 9, le rapport entre les deux est quasiment lui-même une tour de hauteur 9.

A ce stade de l'algorithme, on obtient alors (par une opération de complexité non négligeable) que $\mathsf{h}_{\mathcal{C}'}(\mathcal{L}) = 0$. La taille des ensembles manipulés est assez éloquente.

Table des exemples

1 Le langage $\mathcal{L}_1 = A^*abA^*$ 21

1.1 Un graphe orienté . . . 21
1.2 Le monoïde fini M_1 . . . 22
1.3 Langage rationnel . . . 30
1.4 Langage reconnaissable . . . 31
1.5 Automate reconnaissant $\mathcal{L}_1$ . . . 35
1.6 Langage reconnu par un automate . . . 36
1.7 Factorisations de $\mathcal{L}_1$ . . . 46
1.8 Automate universel de $\mathcal{L}_1$ . . . 48
1.9 L'automate universel de $\mathcal{L}_1$ est reconnaissable . . . 51
1.10 Automate A-universel de $\mathcal{L}_1$ . . . 54
1.11 Écorché de l'automate universel de $\mathcal{L}_1$ . . . 61
1.12 Développé d'un automate reconnaissant $\mathcal{L}_1$ . . . 66
1.13 Ecorché du développé . . . 70

2 Distance préfixe 25

3 Le semi-anneau $\mathbb{N}$ 25

4 Le semi-anneau de Boole 26

5 Semi-anneaux principaux 26

6 Etoile de 1/2 27

7 Langages reconnus par la partie $\{1,2\}$ de $\mathbb{Z}/3\mathbb{Z}$ 31

7.1 Langage reconnaissable . . . 31
7.2 Factorisations de $\mathcal{L}_3 = \{1,2\}$ dans le monoïde $N_2 = \mathbb{Z}/3\mathbb{Z}$ . . . 46
7.3 Automate universel du langage $\mathcal{L}_3$ . . . 48
7.4 Factorisations du langage $\mathcal{L}_3' = \{u \in A^* \mid |u|_a \neq |u|_b \mod 3\}$ . . . 52

7.5 Automate {1}-universel de $\mathcal{L}_3$ 53
7.6 Automate universel du langage $\mathcal{L}_3'$ 55
7.7 Écorché de l'automate {1}-universel du langage $\mathcal{L}_3$ 61

8 Le langage $\mathcal{L}_2 = a^+$ 47
8.1 Factorisations du langage $\mathcal{L}_2 = a^+$ dans le monoïde a^* 47
8.2 Automate universel du langage $\mathcal{L}_2$ 48
8.3 Automate $\{a\}$-universel du langage $\mathcal{L}_2$ 53
8.4 Écorché de l'automate universel du langage $\mathcal{L}_2$ 61
8.5 Développé de l'automate minimal du langage $\mathcal{L}_2$ 66

9 Factorisations d'un langage non reconnaissable 47

10 Le langage $\mathcal{L}_4 = ((a+c)(b+c)+(b+(a+c)a)b^*(a+c))^*(b+(a+c)a)$ sur A^* 59
10.1 Calcul de l'automate universel 59
10.2 Ecorché de l'automate universel de $\mathcal{L}_4$ 61
10.3 Un automate d'enlacement minimal pour $\mathcal{L}_4$ 120

11 Ordre sur $\mathcal{P}(Q)$ 64

12 Le langage $\mathcal{L}_{r_1}$ sur A^* (Langage réversible) 71
12.1 Automate réversible $\mathcal{A}_{r_1}$ et développé de $\mathcal{A}_{r_1}$ 71
12.2 Profil des états du développé de $\mathcal{A}_{r_1}$ 86
12.3 Automate universel de $\mathcal{L}_{r_1}$ 87
12.4 Un automate quasi-réversible de $\mathcal{L}_{r_1}$ 119

13 Automate réversible 77

14 Automate minimal et universel de $\mathcal{L}_{g_1}$ (Langage à groupe) 81

15 Le langage $\mathcal{L}_{g_2}$ sur A^* (Langage à groupe) 81
15.1 Automate minimal et universel de $\mathcal{L}_{g_2}$ 81
15.2 Hauteur d'étoile de $\mathcal{L}_{g_2}$ 113

16 Construction d'un automate réversible 94

17 Enlacement d'un graphe 100

18 Morphismes conforme et non conforme 102

19 Indice i_{E} **105**

20 Automate sur $\mathbb{N}_{\mathsf{m}}$ avec valeur minimale éloignée **126**

21 Une série non-séquentielle uniformément bornée **129**

22 L'automate unaire avec multiplicité $\mathcal{A}_{\mathsf{m}_1}$ **130**

22.1 Les paramètres de $\mathcal{A}_{\mathsf{m}_1}$ 130

22.2 Décision de la séquentialité de α_{m_1} 135

22.3 Calcul d'un automate non ambigu pour α_{m_1} 141

23 Fréquence des circuits de poids maximum **133**

24 $\mathsf{E}_4 = (\mathbf{x}(a\cdot(\mathbf{y}\,b)^\star\cdot a)\,\mathbf{y}+\mathbf{y}\,(b\cdot(a\,\mathbf{x})^\star\cdot b)\,\mathbf{x})^\star$ **151**

24.1 Présentation de l'expression 151

24.2 Arbre de l'expression E_4 154

24.3 Automate de l'expression E_4 155

24.4 Dérivation unitaire 170

25 $\mathsf{E}_1 = (\frac{1}{6}a^* + \frac{1}{3}b^*)^*$ **153**

25.1 Terme constant de E_1 153

25.2 Dérivées de E_1 160

25.3 Termes dérivés 162

26 $\mathsf{E}_2 = (a\,b\,a + (a\,(a - b\,a)))$ **153**

26.1 Ecriture de E_2 153

26.2 Dérivées de E_2 161

26.3 Termes dérivés fantômes 163

27 $\mathsf{E}_3 = 5\,((2\,a\,b) + ((3\,b)\cdot(4\,(a\,b)^\star)))^\star$ **151**

27.1 Terme constant de E_3 151

27.2 Dérivation et dérivation unitaire 171

28 Problème sur les termes dérivés fantômes **168**

Index

action, 23
alphabet, 23
automate
    quasi-réversible, 77
    à groupe, 80
    à multiplicité, 41
    complet, 36
    déterminisé d'un automate, 38
    déterministe, co-déterministe, 36
    des termes dérivés, 166
    émondé, 36
    généralisé, 104
    non-ambigu, 34
    normalisé, 32
    réversible, 76
    sous-jacent, 42
    sur un alphabet, 35
    sur un monoïde, 33
    sur un semi-anneau, 31
    universel, 48
    univoque, 144
automates équivalents, 32

boucle, 20

cône, 28
calcul, 36
    victorieux, 125
carré d'un automate, 41
chemin, 20
circuit (élémentaire), 20
composante fortement connexe, 21
corde, 90

dérivée
    cassante, 171
    unitaire, 169
dérivée d'une expression
    par rapport à un mot, 158
    par rapport à une lettre, 157
développé d'un automate, 66
distance préfixe, 24

écorché
    de l'automate universel, 60
    du développé, 70
enlacement
    d'un automate, 104
    d'un graphe orienté, 100
ensembles rationnels, 30
étage de l'automate universel
    d'un langage à groupe, 80
    d'un langage réversible, 86
état accessible, 36
étoile, 27
expression rationnelle, 98
    à multiplicité, 150
    réduite, 154
    valide, 152
expressions équivalentes, 152

facteur, 24, 46
factorisation, 46
    initiale, terminale, 49
fonction de production, 44

graphe
    acyclique, 21
    connexe, 21
    critique, 130
    de Cayley, 22
    fortement connexe, 21
    orienté, 20
    sous-jacent à un automate, 32

hauteur d'étoile

d'un langage rationnel, 99
d'une expression rationnelle, 99

idempotent, 25
image miroir, 24
indice i_E d'un automate généralisé, 105

langage, 23
à groupe, 80
réversible, 76
rationnel, 30
reconnaissable, 31
reconnu par un automate, 36
lettres, 23
longueur d'une expression, 151

monoïde, 21
de transition, 37
libre, 23
syntaxique, 41
morphisme
conforme, 102
de graphes, 21
syntaxque, 41
mot idempotent pour un automate, 114
mots, 23

ordre
lexicographique, 24
radiciel, 24

partie propre d'une série, 29
passé/futur d'un état, 34
pelote, 21
poids d'un circuit, 130
polynôme, 29
polynômes d'expressions, 156
préfixe, 24
produit de deux automates, 41
profondeur d'une expression, 151

quipu, 90
quotient à gauche
d'une série, 30
dans un monoïde, 22

rationnel, 27
relation, 20
représentation linéaire, 37

semi-anneau, 25
idempotent, 25
positif, 167
principal, 26
semi-module, 28
séries, 28
rationnelles, 30
séquentielles, 43
sous-factorisation, 46
sous-graphe, 20
suffixe, 24
support, 20
support d'une série, 29

terme constant
d'une expression rationnelle, 152
d'une série, 29
termes dérivés
d'une expression, 162
fantômes, 163
transducteurs, 44
transitions, 32, 35
transitions spontanées, 36
translatée, 126
transposé d'un automate, 41
type fini, 28

uniformément divergente, 128

vecteur translaté, 137

Bibliographie

[1] A. Ambainis et R. Freivalds, 1-way quantum finite automata: strengths, weaknesses and generalizations. In *39th Ann. Symposium on FOCS* (1998), 332–342.

[2] D. Angluin, Inference of reversible languages. *J. of the ACM* **29** (1982), 741–765.

[3] V. Antimirov, Partial derivatives of regular expressions and finite automaton constructions. *Theoret. Comput. Sci.* **155** (1996), 291–319.

[4] A. Arnold, A. Dicky, et M. Nivat, A note about minimal non-deterministic automata. *Bull ot the EATCS* **47** (1992), 166–169.

[5] M.-P. Béal, O. Carton, C. Prieur, et J. Sakarovitch, Squaring transducers. *Proc. of Latin 2000, Lect. Notes in Comp. Sci.* **1776** (2000), 397–406.

[6] G. Berry et R. Sethi, From regular expressions to deterministic automata, *Theoret. Comput. Sci.* **48** (1986), 117–126.

[7] J. Berstel et J.-E. Pin, Local languages and the Berry-Sethi algorithm, *Theoret. Comput. Sci.* **155** (1996), 439–446.

[8] J. Berstel et Ch. Reutenauer, *Les séries rationnelles et leurs langages.* Masson, 1984. Traduction: *Rational Series and their Languages.* Springer, 1986.

[9] J. A. Brzozowski, Derivatives of regular expressions. *J. Assoc. Comput. Mach.* **11** (1964), 481–494.

[10] A. Buchsbaum, R. Giancarlo, et J. Westbrook, On the Determinization of Weighted Finite Automata. *Proc. of ICALP'98, Lect. Notes in Comp. Sci.* **1443** (1998), 482–493.

[11] P. Caron et M. Flouret, Glushkov construction for multiplicities. *Pre-Proceedings of CIAA'00*, M. Daley, M. Eramian and S. Yu, eds, Univ. of Western Ontario, (2000), 52–61.

[12] O. Carton, Factorisations et matrice des facteurs. Manuscrit (1994).

[13] J.-M. Champarnaud et D. Ziadi, New finite automaton constructions based on canonical derivatives. *Pre-Proceedings of CIAA'00*, M. Daley, M. Eramian and S. Yu, eds, Univ. of Western Ontario, (2000), 36–43.

[14] Ch. Choffrut, Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoret. Comput. Sci.* **5** (1977), 325–337.

[15] M. Chrobak, Finite Automata and Unary Languages. *Theoret. Comput. Sci.* **47** (1986), 149–158.

[16] G. Cohen, P. Moller, J.-P. Quadrat, et M. Viot, Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proc.: Special issue on D.E.S.* **77.1** (1989).

[17] R. Cohen, Star height of certain families of regular events. *J. Computer System Sci.* **4** (1970), 281–297.

[18] R. Cohen et J. A. Brzozowski, General properties of star height of regular events. *J. Computer System Sci.* **4** (1970), 260–280.

[19] J. H. Conway, *Regular algebra and finite machines.* Chapman and Hall, 1971.

[20] B. Courcelle, D. Niwinski, A. Podelski, A geometrical view of the dererminization ond minimization of finite-state automata. *Math. Systems Theory* **24** (1991), 117–146.

[21] F. Dejean et M.-P. Schützenberger, On a question of Eggan. *Inform. and Control* **9** (1966), 23–25.

[22] L. C. Eggan, Transition graphs and the star-height of regular events. *Michigan Mathematical J.* **10** (1963), 385–397.

[23] S. Eilenberg, *Automata, Languages and Machines, volume A.* Academic Press, 1974.

[24] S. Gaubert, Rational Series over Dioids and Discrete Event Systems. *Proc. of the 11th Conf. on Anal. and Opt. of Systems, Lect. Notes in Contr. and Inf. Sci.* **199** (1994).

[25] S. Gaubert, On the Burnside problem for Semigroups of Matrices in the (max,+) Algebra. *Semigroup Forum* **52** (1996), 271–292.

[26] S. Gaubert, Methods and applications of (max,+) linear algebra. *rapport de recherche INRIA* **3088** (1997).

[27] V. Glushkov, The abstract theory of automata. *Russian Mathematical Surveys 16* (1961), 1–53.

[28] Ch. Hagenah et A. Muscholl, Computing ε-Free NFA from regular expressions in $O(n \log^2(n))$ time. *R.A.I.R.O. Inf. Théorique* **34**, (2000), 257–277.

[29] T.E. Hall, Biprefix codes, inverse semigroups and syntactic monoids of injective automata. *Theoret. Comput. Sci.* **32** (1984), 201–213.

[30] T. Harju et J. Karhumäki, The equivalence problem of multitape finite automata. *Theoret. Comput. Sci.* **78** (1991), 347–355.

[31] K. Hashiguchi et N. Honda, The star height of reset-free events and strictly locally testable events. *Inform. and Control* **40** (1979), 267–284.

[32] K. Hashiguchi, Limitedness theorem on finite automata with distance functions. *J. of Comput. Syst. Sci.* **24** (1982), 233–244.

[33] K. Hashiguchi, Algorithms for determining relative star height and star height. *Inform. and Computation* **78** (1988), 124–169.

[34] K. Hashiguchi, Improved limitedness theorem on finite automata with distance functions. *Theoret. Comput. Sci.* **72** (1990), 27–38.

[35] P.-C. Héam, A lower bound for reversible automata. *Theoret. Informatics Appl.* **34** (2000), 331–341.

[36] P.-C. Héam, Contribution à l'algorithmique des automates : complexité et aspects topologiques. *Thèse de doctorat*, Université Paris 7, 2001.

[37] J. Hromkovič, S. Seibert, et T. Wilke, Translating regular expressions into small ε-free nondeterministic finite automata. *Proc. of STACS'97, Lect. Notes in Comp. Sci.* **1200** (1997), 55–66.

[38] D. Krob, Differentiation of K-rational expressions. *Int. J. of Algebra and Computation 2* (1992), 57–87.

[39] W. Kuich et A. Salomaa, *Semirings, Automata, Languages.* Springer, 1986.

[40] G. Lallement, *Semigroups and combinatorial applications.* Wiley, 1979.

[41] H. Leung, Limitedness theorem on finite automata with distance functions: an algebraic proof *Theoret. Comput. Sci.* **81** (1991), 137–145.

[42] S. Lombardy et J. Sakarovitch, On the star height of rational languages, a new presentation for two old results *Proc. of 3rd ICLWC, Kyoto* (2000) (M. Ito, ed.), World Scientific, *à paraître.*

[43] S. Lombardy et J. Sakarovitch, Star height of reversible languages and universal automata, *accepté à Latin'02.*

[44] S. Lombardy, Sequentialization and unambiguity of (max, +) rational series over one letter. *Pre-Proc. of Workshop on Max-plus algebra, Prague* (2001) (S. Gaubert, ed.).

[45] O. Matz et A. Potthoff, Computing small nondeterministic finite automata. *proc. of TACAS'95, BRICS Notes Series* (1995), 74–88.

[46] R. McNaughton et H. Yamada, Regular expressions and state graphs for automata. *IRE Trans. on electronic computers* **9** (1960), 39–47.

[47] R. McNaughton, The loop complexity of pure-group events. *Inform. and Control* **11** (1967), 167–176.

[48] Moebius, *Sur l'étoile.* Gentiane, 1983, Rééd. Casterman, 1990.

[49] M. Mohri, Finite-State Transducers in Language and Speech Processing. *Computat. Ling.* **23.2** (1997), 269–311.

[50] C. Nicaud, Étude du comportement en moyenne des automates finis et des langages rationnels. *Thèse de doctorat*, Université Paris 7, 2000.

[51] J.-E. Pin, On reversible automata. In *Proc. 1st LATIN Conf., (I. Simon, Ed.), Lecture Notes in Comput. Sci.* **583** (1992), 401–416.

[52] J.-E. Pin, Variétés de langages formels. Masson, 1984. Traduction: *Varieties of formal languages* North Oxford Acad. Pub., 1986.

[53] G. Raney, Sequential functions. *J. Assoc. Comput. Mach.* **5** (1958) 177–180.

[54] J. Sakarovitch, A construction on automata that has remained hidden. *Theoret. Comput. Sci.* **204** (1998), 205–231.

[55] J. Sakarovitch, *Éléments de théorie des automates.* Vuibert, *à paraître.*

[56] A. Salomaa, *Jewels of formal language theory.* Computer Science Press, 1981.

[57] P.V. Silva, On free inverse monoid languages. *Theoret. Informatics and Appl.* **30** (1996), 349–378.

[58] I. Simon, Recognizable sets with multiplicities in the tropical semiring. *Proc. of MFCS'88, Lect. Notes in Comp. Sci.* **324** (1988), 107–120.

[59] I. Simon, The non deterministic complexity of a finite automaton. *Mots (M. Lothaire)*, Hermès (1990) 384–400.

[60] I. Simon, On semigroups of matrices over the tropical semiring. *R.A.I.R.O. Inf. Théorique*, (1994), 277–294.

[61] N.J.A. Sloane, *The On-Line Encyclopedia of Integer Sequences.* `http://www.research.att.com/~njas/sequences/`

[62] J.B. Stephen, Presentations of inverse monoids. *J. Pure Appl. Alg.* **63** (1990), 81–112.

[63] M. Szalay, On the maximal order in S_n and S_n^*. *Acta arithm.* **37** (1980), 321–331.