UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SÉRIES FORMELLES À COEFFICIENTS DANS UN CORPS FINI ET AUTOMATES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

MAXIME GÉLINAS

JUIN 2015

REMERCIEMENTS

La rédaction d'un mémoire de maîtrise n'est pas chose légère, je crois maintenant être en position pour le savoir. Seul, je n'aurais jamais même espéré pouvoir accomplir une telle tâche. Cette maigre page de remerciements ne saurait rendre justice au support que j'ai reçu tout au long de mes études universitaires.

J'aimerais d'abord remercier mes professeurs d'université, qui ont pu me maintenir intéressé aux mathématiques jusqu'à la maîtrise. Principalement, j'aimerais remercier Luc Bélair, mon directeur de maîtrise, sans qui la recherche n'aurait pas été possible et sans qui la motivation pour rédiger aurait été beaucoup plus difficile à trouver. J'aimerais aussi remercier Françoise Point, ma co-directrice de maîtrise, qui m'a soutenu tout au long de mon séjour en Belgique, qui m'a permis d'être productif et qui m'a beaucoup aider à travers le dépaysement.

J'aimerais aussi remercier mes amis les plus proches pour m'avoir changé les idées et m'avoir motivé à me rendre aussi loin. Que ce soit à l'école pour organiser un atelier de travail, ou que ce soit dans nos foyers pour se distancer de tout ce stress, il est impossible de surestimer le support moral que peut apporter un cercle d'amis aussi solide. Il serait exhaustif de vous remercier un à un, mais sachez que votre présence dans ma vie m'est plus importante que vous ne pouvez l'imaginer. Un remerciement particulier à Alex, Alice et Marco, avec qui j'ai partagé rires et labeurs, et un remerciement spécial à Simon qui m'encourage à écrire cette page de remerciements.

Finalement, je remercie ma famille : mon père, ma mère et ma sœur. Je leur dois bien plus que le support moral. J'ai droit à leur amour inconditionnel depuis le

moment où je suis né, et sans eux je ne serais pas où je suis aujourd'hui, je ne serais pas la personne que je suis aujourd'hui. Il m'apporte le plus grand bonheur de savoir que je peux les rendre fiers.

TABLE DES MATIÈRES

LISTE DES FIGURES vii

RÉSUMÉ ix

INTRODUCTION 1

CHAPITRE I
NOTIONS PRÉLIMINAIRES 5

1.1 Langages rationnels et langages reconnaissables 5

1.2 La logique de premier ordre 10

1.3 Une structure sur les séries formelles 12

CHAPITRE II
LES ω-AUTOMATES 17

2.1 Les automates sur les mots infinis 17

2.2 Propriétés de fermeture des automates de Büchi 22

2.3 Lemme d'itération pour les ω-langages 26

2.4 Reconnaissabilité des prédicats 32

CHAPITRE III
RECONNAISSABILITÉ ET DÉFINISSABILITÉ 41

3.1 Reconnaissabilité et définissabilité dans les polynômes 41

3.2 Reconnaissabilité et définissabilité dans les séries formelles 46

3.3 Autres approches 50

3.3.1 Approche par la relation $L = \bigcup_{i=1}^{n} X_i Y_i^{\omega}$ 50

3.3.2 Approche par les automates de Büchi déterministes 51

CONCLUSION 55

RÉFÉRENCES 57

[Cette page a été laissée intentionnellement blanche]

LISTE DES FIGURES

Figure		Page
1.1	Automate fini reconnaissant xyz	9
2.1	Automate de Büchi reconnaissant $\{0,1\}^*0^\omega$.	21
2.2	Automate de Büchi pour l'addition	34
2.3	Automate de Büchi pour $\cdot X$	35
2.4	Automate de Büchi pour $\mathbb{F}_p[X]$	36
2.5	Automate de Büchi pour $\prec$	37
2.6	Automate de Büchi pour λ_X	38
2.7	Automate de Büchi pour $X_X(\cdot,\cdot,k)$	39
3.1	Automate de l'exemple 3.3	45
3.2	Automate de Büchi pour $\mathbb{F}_2[X]$	53

[Cette page a été laissée intentionnellement blanche]

RÉSUMÉ

En 2011, Michel Rigo et Laurent Waxweiler ont publié un article au sujet des ensembles reconnaissables de polynômes à coefficients dans un corps fini. Ils ont prouvé que les ensembles *P-reconnaissables*, c'est-à-dire reconnaissables dans une base P, où P est un polynôme non constant, correspondent aux ensembles définissables dans la structure de groupe abélien sur les polynômes, enrichie d'un ordre sur le degré et de quelques fonctions. On démontre dans ce mémoire une propriété semblable des séries formelles à coefficients dans un corps fini. Nous démontrons que les ensembles reconnaissables correspondent aux ensembles définissables dans la structure de groupe abélien sur les séries formelles, enrichie de l'ensemble des polynômes comme sous-ensemble des séries formelles, d'un ordre sur le degré des polynômes, de la multiplication par X et de quelques fonctions et relations.

MOTS-CLÉS : Ensemble reconnaissable, logique de premier ordre, série formelle, automate de Büchi, ω-langage.

INTRODUCTION

Le théorème de Büchi stipule que tout ensemble d'entiers est k-reconnaissable si et seulement s'il existe une formule définissant cet ensemble dans la logique de premier ordre de la structure

$$(\mathbb{N}, +, V_k),$$

où V_k est la fonction définie par $V_k(0) = 1$, et $V_k(n)$ donne la plus grande puissance de k divisant n si $n \geq 1$. Une formule définit un ensemble E si les éléments de E sont exactement ceux qui satisfont la formule. Par ensemble k-reconnaissable, on entend que cet ensemble est reconnaissable par automate fini si ses éléments sont écrit en base k. Ce théorème est tiré de (Büchi, 1960a) et établit un lien clair entre la logique de premier ordre sur les entiers naturels et la théorie des automates. Pour un aperçu sur le sujet, on peut consulter (Bruyère et al., 1994).

Par exemple, l'ensemble des nombres impairs est définissable dans $(\mathbb{N}, +, V_2)$ car la formule $(V_2(x) = 1) \wedge \neg(x = 0)$ est satisfaite si et seulement si x est impair. Par ailleurs, on sait que l'ensemble des nombres impairs est 2-reconnaissable, c'est-à-dire qu'il existe un automate fini qui reconnait les nombres impairs écrits en binaire.

En s'inspirant du théorème de Büchi, M. Rigo et L. Waxweiler ont démontré dans (Rigo et Waxweiler, 2011) que tout ensemble de polynômes à coefficients dans un corps fini est P-reconnaissable si et seulement si cet ensemble est définissable dans la logique de premier ordre de la structure

$$(\mathbb{F}[X], +, \prec, (\cdot C \mid C \in \mathbb{F}[X]), V_P).$$

L'ensemble $\mathbb{F}[X]$ désigne l'ensemble des polynômes à coefficient dans un corps fini. Le prédicat $\prec$ est une relation binaire qui compare le degré de deux polynômes. La fonction $V_P : \mathbb{F}[X] \to P^{\mathbb{N}}$ est définie par $V_P(0) = 1$, et $V_P(A)$ est la plus grande puissance de P qui divise A si A est non-nul. La fonction représentée par $\cdot C$ est la multiplication habituelle par un polynôme C fixé. Un sous-ensemble de $\mathbb{F}[X]$ est dit *P-reconnaissable* s'il existe un automate fini le reconnaissant lorsque ses éléments sont écrits en base P, où P est un polynôme non constant. Pour lier les polynômes à la théorie des automates, chaque polynôme est codé par un mot formé des coefficients de ce polynôme. La première lettre est le coefficient de la plus grande puissance de P avec coefficient non-nul, la dernière lettre est le coefficient de P^0.

Inspiré de ce résultat, ce mémoire a pour but de prouver un résultat similaire dans les séries formelles à coefficients dans un corps fini. Nous voulons démontrer que les ensembles reconnaissables correspondent aux ensembles définissables dans la logique de premier ordre de la structure

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k)\}_{k \in \mathbb{F}_p}),$$

où $\mathbb{F}_p$ est le corps fini à p éléments, où p est un nombre premier, et $\mathbb{F}_p[\![X]\!]$ est l'ensemble des séries formelles à coefficients dans $\mathbb{F}_p$ (voir section 1.3). Pour simplifier l'étude de ce sujet et l'écriture de ce mémoire, nous travaillons uniquement dans la base X des séries formelles, contrairement à (Rigo et Waxweiler, 2011), où les polynômes sont écrits dans une base P quelconque. De façon semblable à (Rigo et Waxweiler, 2011), pour faire le lien entre les séries formelles et la théorie des automates, on code chaque série formelle par un mot infini formé des coefficients de cette série, commençant par le coefficient de X^0.

Le chapitre I établit les notions préliminaires à la compréhension de ce mémoire. Nous établissons les concepts fondamentaux de la théorie des langages et des auto-

mates. Nous définissons aussi la structure sur les séries formelles déjà mentionnées. Dans le chapitre II, nous établissons les concepts d'ω-langage et d'ω-automate, qui découlent des notions vue au chapitre I. Les ω-automates sont nécessaires à la reconnaissabilité des séries formelles. Nous utilisons ces notions pour démontrer la reconnaissabilité des prédicats de la structure sur les séries formelles ci-dessus. Dans le chapitre III, nous étudions d'abord le théorème 14 de (Rigo et Waxweiler, 2011) qui établit le lien entre reconnaissable et définissable dans la structure des polynômes, pour ensuite s'en inspirer pour prouver le théorème 3.5, qui est le résultat principal de ce mémoire. Nous finissons le chapitre en décrivant les différentes approches qui ont été considérées pour prouver le théorème 3.5.

[Cette page a été laissée intentionnellement blanche]

CHAPITRE I

NOTIONS PRÉLIMINAIRES

Ce chapitre vise à établir les notions de base pour la compréhension de ce mémoire. Nous définissons d'abord les concepts de langages rationnels, de langages reconnaissables et d'automates finis. Nous définissons ensuite les notions de logique que nous utilisons. Ces notions aideront le lecteur à mieux assimiler les concepts du chapitre II. Nous définissons aussi la structure logique sur les séries formelles que nous utilisons dans le chapitre III.

1.1 Langages rationnels et langages reconnaissables

Cette section couvre les notions pertinentes de la théorie des langages et des automates pour ce mémoire. Les définitions, théorèmes et autres résultats sont tirés de (Autebert, 1994). Nous travaillons avec des mots infinis. Nous commençons par étudier les langages de mots finis pour ensuite étendre les notions aux langages de mots infinis.

Définition 1.1. Soit E un ensemble. On définit E^* l'ensemble de toutes les concaténations finies possibles d'éléments de E. On appelle * l'*étoile de Kleene*. On dit que E^* est le *monoïde libre* sur E. On appelle *lettres* les éléments de E et *mots* les éléments de E^*. Le nombre de lettres dans un mot w est appelé la *longueur* de w qui est notée $|w|$.

Exemple 1.2. Soit $E = \{a\}$. Alors $E^* = \{a^n \mid n \geq 0\}$. Pour tout $n \geq 0$, $|a^n| = n$.

Définition 1.3. Soit M un monoïde. Nous notons $Rat(M)$ l'ensemble des *sous-ensembles rationnels* de M. Nous définissons $Rat(M)$ comme la plus petite famille $\mathcal{R}$ de sous-ensembles de M telle que

(i) $\emptyset \in \mathcal{R}$, $\{m\} \in \mathcal{R}$, pour tout $m \in M$,

(ii) $X, Y \in \mathcal{R}$ entraîne $X \cup Y$, $XY \in \mathcal{R}$,

(iii) $X \in \mathcal{R}$ entraîne $X^+ = \bigcup_{n \geq 1} X^n \in \mathcal{R}$.

Il est intéressant de noter que si les conditions (i) et (ii) sont remplies, alors la conditions (iii) est équivalente à

(iii') $X \in \mathcal{R}$ implique que $X^* \in \mathcal{R}$.

Définition 1.4. Un *automate fini* est un quintuplet

$$\mathcal{A} = (Q, \Sigma, D, F, \Delta)$$

où Q est un ensemble fini d'états, Σ est un ensemble fini de lettres appelé l'*alphabet* de $\mathcal{A}$, D est un sous-ensemble de Q appelé l'ensemble des *états initiaux*, F est un sous-ensemble de Q appelé l'ensemble des *états acceptants*, et Δ est une relation définie sur $Q \times \Sigma \times Q$ appelée la *relation de transition* de $\mathcal{A}$. On dit que $\mathcal{A}$ est *déterministe* si $|D| = 1$, et si pour tout $q \in Q$ et pour tout $x \in \Sigma$, on a

$$|\{q' : (q, x, q') \in \Delta\}| \leq 1.$$

Soit $w = a_0 a_1 a_2 \ldots a_n$ un mot sur l'alphabet Σ. Un *chemin* étiqueté par w dans un automate $\mathcal{A}$ est une suite d'états $q_0 q_1 q_2 \ldots q_{n+1}$ telle que

- $q_0 \in D$,
- $(q_k, a_k, q_{k+1}) \in \Delta$, pour tout $k = 0, 1, \ldots, n$.

Le mot w est *accepté* par $\mathcal{A}$ s'il existe un chemin étiqueté par w dans $\mathcal{A}$ finissant dans un état de F.

Définition 1.5. Soit L un langage sur l'alphabet Σ. On dit que L est *reconnaissable* s'il existe un automate fini qui accepte exactement les mots de L. On note $Rec(\Sigma^*)$ l'ensemble des langages reconnaissables sur l'alphabet Σ.

Théorème 1.6. *Un langage reconnaissable peut toujours être reconnu par un automate fini déterministe.*

La preuve se trouve notamment dans (Autebert, 1994), p. 45. Cette proposition dit en fait que pour tout automate fini, il existe au moins un automate fini déterministe qui reconnait le même langage.

Théorème 1.7 (Théorème de Kleene). *Soit Σ un alphabet fini. Alors $Rec(\Sigma^*) = Rat(\Sigma^*)$.*

Ce résultat nous assure que tout langage reconnaissable est rationnel, et vice-versa. Une preuve de ce théorème se retrouve dans (Autebert, 1994), p.53.

La prochaine proposition n'est peut-être pas aussi puissante que le théorème 1.6, mais la forme d'automate qu'elle propose peut être pratique lorsqu'on veut montrer les propriétés de fermeture des langages reconnaissables.

Proposition 1.8 ((Bruyère, 1985), prop 1.18). *Étant donné un automate fini, il existe une procédure effective pour construire un automate fini équivalent comportant un unique état initial auquel n'aboutit aucune transition, et un unique état final duquel aucune transition n'est issue.*

Démonstration. Soit $\mathcal{A}$ un automate fini qui reconnait le langage L, possédant un ensemble $D = \{q_0, q_1, \ldots, q_n\}$ d'états initiaux. Soit d un nouvel état. Pour chaque flèche allant d'un état initial à un état q, on crée une flèche de même étiquette allant de d à q. L'état d sera notre unique état initial, et l'état est une source,

aucune transition n'y aboutit. Le comportement du nouvel automate n'est pas différent de l'ancien, les deux automates reconnaissent L.

On procède de manière similaire pour les états finaux. On pose un nouvel état f. Pour chaque flèche allant d'un état q à un état final, on crée une flèche de même étiquette allant de q à f. L'état f sera notre unique état final et est un puits, aucune transition n'en est issue. Le comportement de l'automate n'a toujours pas changé, l'automate que nous avons construit reconnait L.

Il est important de remarquer que même si $\mathcal{A}$ était un automate déterministe, l'automate que nous avons construit peut être non-déterministe. □

Le prochain théorème est un résultat classique de la théorie des automates. Le lemme d'itération (ou *Pumping Lemma*) est souvent utilisé pour démontrer qu'un langage n'est pas reconnaissable. La démonstration est semblable à celle retrouvée dans (Autebert, 1994).

Théorème 1.9. *(Lemme d'itération) Soit L un langage reconnaissable. Alors il existe un entier $N \geq 1$ tel que pour tout mot $w \in L$ avec $|w| \geq N$, on peut écrire $w = xyz$ tel que*

1. $|y| \geq 1$,
2. $|xy| \leq N$,
3. $\forall i \geq 0,\ xy^i z \in L$.

Démonstration. Considérons L un langage sur l'alphabet Σ reconnu par un automate fini $\mathcal{A}$ ayant N états. Supposons que $\mathcal{A}$ est déterministe. Considérons $w \in L$ un mot de longueur $p \geq N$. Soit q_0 l'état initial de $\mathcal{A}$ et soit

$$q_0 q_1 \cdots q_N \cdots q_{p-1}$$

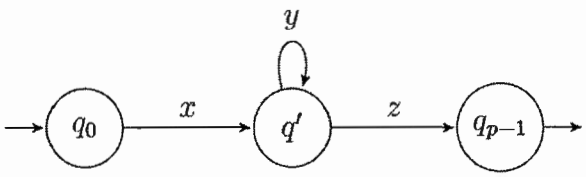

Figure 1.1 Automate fini reconnaissant xyz

le chemin parcouru par w dans l'automate $\mathcal{A}$. Comme $w \in L$, q_{p-1} est un état acceptant.

Ce chemin parcourt au moins $N+1$ états dans $\mathcal{A}$. Or, $\mathcal{A}$ possède N états, cela signifie que w parcourt au moins deux fois un même état dans $\mathcal{A}$. Soit q' cet état. Disons que l'état q' est parcouru une première fois à la s^e lettre de w, et une seconde fois à la t^e lettre de w. On peut supposer que s et t sont inférieurs à N.

Posons $x, y, z \in \Sigma^*$ tels que $|x| = s$, $|y| = t - s$ et tel que $w = xyz$. On peut remarquer que $|y| \geq 1$, car c'est l'étiquette d'un chemin allant de q' à q', visitant l'état q' plus d'une fois. On remarque aussi que $|xy| \leq N$ puisque $|xy| = t \leq N$. Il reste à vérifier que $xy^iz \in L$ pour tout $i \geq 0$. En d'autres mots, nous devons vérifier que xy^iz est l'étiquette d'un chemin acceptant dans $\mathcal{A}$.

On sait que x est l'étiquette d'un chemin allant de q_0 à q', que y est l'étiquette d'un chemin allant de q' à q', et que z est l'étiquette d'un chemin allant de q' à q_{p-1}. La figure 1.1 illustre le comportement de cet automate. On remarque que y^i est l'étiquette d'un chemin allant de q' à q' peu importe la valeur de i. Par exemple, y^0 est le mot vide, qui est évidemment l'étiquette d'un chemin de longueur 0 allant de q' à q'. Ainsi, comme y^i est toujours l'étiquette d'un chemin allant de q' à q', on a que xy^iz est toujours l'étiquette d'un chemin allant de q_0 à q_{p-1}, c'est-à-dire l'étiquette d'un chemin acceptant. □

1.2 La logique de premier ordre

Dans cette section, nous établissons quelques concepts de logique mathématique qui sont pertinents pour la compréhension de ce mémoire. Le but principal de cette section est de définir ce qu'on entend par *ensemble définissable.* Toutes les définitions sont tirées de (Cori et Lascar, 2003).

Définition 1.10. Un *langage de premier ordre* est un ensemble L constitué d'une première partie commune à tous les langages, à savoir :

- d'un ensemble infini dénombrable de variables,
- des parenthèses $(,)$ et des connecteurs logiques $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$,
- du quantificateur universel $\forall$, et du quantificateur existentiel $\exists$,

et d'une seconde partie, propre à chaque langage, à savoir :

- de symboles de constante,
- de symboles de fonction à n places (ou n-aires),
- de symboles de relation à n places (ou n-aires).

Remarque 1.11. Lorsque nous définissons un langage, nous énumérons uniquement les symboles de constante, de fonction et de relation. Il est redondant de spécifier l'inclusion de l'ensemble de variables, des connecteurs et des quantificateurs puisque ces éléments sont communs à tous les langages. De plus, dans le cadre de ce mémoire, tous les langages de premier ordre sont égalitaires, c'est-à-dire que tous les langages définis contiennent un symbole d'égalité $=$ que nous omettons.

Définition 1.12. Soit L un langage de premier ordre. L'ensemble des *termes* de L, noté $\mathcal{T}(L)$, est construit comme suit :

- $\mathcal{T}(L)$ contient les variables et les symboles de constante de L,

- si $t_1, \ldots, t_n$ sont dans $\mathcal{T}(L)$, et si f est un symbole de fonction n-aire dans L, alors $f(t_1, \ldots, t_n)$ est dans $\mathcal{T}(L)$.

Définition 1.13. Une *formule atomique* de L s'obtient par un symbole de relation n-aire R et de n termes du langages, $t_1, \ldots, t_n$, en formant l'expression $R(t_1, \ldots, t_n)$. On construit l'ensemble $\mathcal{F}(L)$ des *formules* du langage de premier ordre L comme suit :

- $\mathcal{F}(L)$ contient toutes les formules atomiques de L,
- si M et N sont dans $\mathcal{F}(L)$, alors l'ensemble contient aussi

$$\neg M,\ (M \wedge N),\ (M \vee N),\ (M \Rightarrow N),\ (M \Leftrightarrow N),$$

et pour toute variable v_n dans le langage L, l'ensemble $\mathcal{F}(L)$ contient

$$(\forall v_n)M,\ (\exists v_n)M.$$

Rien d'autre n'est une formule.

Définition 1.14. On appelle *L-structure* toute structure $\mathcal{D}$ constituée d'un ensemble non-vide D, appelé ensemble de base, muni des composantes suivantes :

- pour chaque symbole de constante c de L, D est muni d'un élément appelé l'interprétation du symbole c dans $\mathcal{D}$,
- pour chaque symbole de fonction n-aire f dans L, D est muni d'une application de D^n vers D appelée l'interprétation du symbole f dans $\mathcal{D}$,
- pour chaque symbole de relation n-aire R dans L, D est muni d'une relation n-aire appelée l'interprétation du symbole R dans $\mathcal{D}$.

Remarque 1.15. Dans le cadre de ce mémoire, nous parlons de *structure logique de premier ordre* plutôt que de L-structure, puisqu'il n'y aura pas d'ambiguïté sur le langage utilisé sur une structure donnée.

Définition 1.16. Soit un langage de premier ordre L et une L-structure $\mathcal{D}$ où D est l'ensemble de base. Soit $k \in \mathbb{N}$, $k \geq 1$, on dit qu'une partie A de D^k est *définissable* dans $\mathcal{D}$ si et seulement s'il existe une formule $\varphi(x_1, \ldots, x_k)$ de L qui est vraie exactement pour les éléments de A. On dit alors que φ définit S.

Exemple 1.17. Soit $L = \{+\}$ un langage de premier ordre, où $+$ est un symbole de fonction binaire. Soit $(\mathbb{N}, +)$ la L-structure où $\mathbb{N}$ est l'ensemble de base et $+$ est interprété par l'opération habituelle d'addition. Nous pouvons définir dans $(\mathbb{N}, +)$ l'ensemble des nombres naturels pairs par la formule

$$(\exists v_0)(= (+(v_0, v_0), v_1)).$$

Les valeurs que peut prendre la variable v_1 pour satisfaire la formule dans les nombres naturels sont exactement les nombres pairs. On dit alors que la formule définit les nombres pairs dans $(\mathbb{N}, +)$. Une notation plus habituelle pour la même formule serait

$$(\exists v_0)(v_0 + v_0 = v_1).$$

C'est une abréviation de la formule de premier ordre, et nous utilisons plutôt cette notation dans le cadre de ce mémoire.

1.3 Une structure sur les séries formelles

Dans cette section, nous définissons une structure logique de premier ordre sur les séries formelles à coefficients dans un corps fini.

Définition 1.18. Soit $\mathbb{F}_p$ le corps fini ayant p éléments, où p est premier. On définit $\mathbb{F}_p[X]$ comme l'ensemble des polynômes à coefficients dans $\mathbb{F}_p$, c'est-à-dire

$$\mathbb{F}_p[X] = \left\{ \sum_{k=0}^{n} a_k X^k \mid a_k \in \mathbb{F}_p, a_n \neq 0 \right\} \cup \{0\}.$$

On définit $\mathbb{F}_p[\![X]\!]$ comme l'ensemble des séries formelles à coefficients dans $\mathbb{F}_p$, c'est-à-dire

$$\mathbb{F}_p[\![X]\!] = \left\{ \sum_{k\geq 0} a_k X^k \mid a_k \in \mathbb{F}_p \right\}.$$

On remarque qu'un polynôme est en fait une série formelle dont les coefficients sont presque tous nuls.

L'opération binaire $+$ additionne les coefficients de deux séries formelles. Soit $P = \sum a_k X^k$ et soit $Q = \sum b_k X^k$, alors $P + Q = \sum (a_k + b_k) X^k$. Le symbole 0 désigne l'élément neutre de cette opération, une série formelle dont tous les coefficients sont nuls. On peut remarquer que chaque série formelle possède un inverse par l'opération $+$.

Définition 1.19. Soit $P = \sum_{k=0}^{n} a_k X^k$ un polynôme, alors le *degré* de P est défini par

$$\deg(P) = \max \{k \mid a_k \neq 0\}.$$

Définition 1.20. La relation binaire $\prec$ est définie sur les polynômes et est valide si le degré du premier est plus petit que le degré du second. C'est-à-dire,

$$P \prec Q \iff \deg(P) < \deg(Q).$$

On écrit $P \preceq Q$ pour désigner que $P \prec Q$ ou que P et Q sont de même degré.

Définition 1.21. L'opération unaire $\cdot X$ est la multiplication d'une série formelle par X. On peut remarquer que l'opération ne fait qu'augmenter de 1 la puissance de X de chaque coefficient. Si $P = \sum_{k\geq 0} a_k X^k$, alors

$$P \cdot X = \sum_{k\geq 0} a_k X^{k+1}.$$

Définition 1.22. L'opération unaire λ_X prend une série formelle et donne la plus grande puissance de X qui apparait avec un coefficient non-nul si une telle

puissance existe, et donne 0 sinon. On remarque que $\lambda_X(P) \neq 0$ si et seulement si P est un polynôme non-nul. Autrement, il n'y a pas de puissance de X dans P qui satisfait la contrainte. Ainsi,

$$\lambda_X(P) = \begin{cases} 0, & \text{si } P = 0 \\ 0, & \text{si } P \text{ n'est pas un polynôme} \\ X^n, & \text{si } P \text{ est un polynôme non-nul de degré } n. \end{cases}$$

Définition 1.23. La relation $X_X(\cdot, \cdot, k)$, où $k \in \mathbb{F}_p$, est une relation binaire. Soit P et Q deux séries formelles. Alors on écrit $X_X(P, Q, k)$ si et seulement si P est une puissance de X, et k est le coefficient de P dans la décomposition de Q. En d'autres mots, $X_X(P, Q, k)$ est vrai si et seulement s'il existe un entier naturel m tel que $P = X^m$, et tel que $Q = \sum_{k \geq 0} a_k X^k$ avec $a_m = k$.

Définition 1.24. Soit p un nombre premier fixé. On fixe le langage de premier ordre

$$L = \{Pol, +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k\}_{k \in \mathbb{F}_p}\}$$

où Pol est un symbole de relation unaire, $+$ un symbole de fonction binaire, 0 un symbole de constante, $\cdot X$ et λ_X sont des symboles de fonction unaire, $\prec$ et chaque $X_X(\cdot, \cdot, k)$ sont des symboles de relation binaire. Nous définissons la structure logique de premier ordre

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k)\}_{k \in \mathbb{F}_p})$$

où $\mathbb{F}_p[\![X]\!]$ est l'ensemble de base, $\mathbb{F}_p[X]$ est l'interprétation du symbole de relation unaire Pol, et l'interprétation des autres symboles est l'interprétation naturelle ou donnée par les définitions précédentes.

Rappelons qu'un ensemble E de $(\mathbb{F}_p[\![X]\!])^d$ est *définissable* dans cette structure s'il existe une formule φ dans la logique de premier ordre de cette structure telle que $\varphi(y)$ est valide si et seulement si $y \in E$.

À l'aide de la structure définie ci-dessus, nous pouvons définir des relations qui sont pratiques dans le dernier chapitre de ce mémoire. On définit d'abord la relation unaire *puissance de* X qui est valide si une série formelle donnée est une puissance de X, et on définit la relation binaire *préfixe* qui est valide si les coefficients d'un polynôme donné correspondent exactement aux premiers coefficients d'une série formelle donnée.

Définition 1.25. Soit Q une série formelle. Alors on écrit $P_X(Q)$ si et seulement si Q est une puissance de X. C'est-à-dire, $Q = X^m$ pour un certain entier naturel m.

On peut définir cette relation par la formule

$$P_X(Q) \longleftrightarrow (\lambda_X(Q) = Q \wedge Q \neq 0).$$

La série Q ne peut pas être nulle puisque 0 n'est pas une puissance de X. Dans ce cas, $\lambda_X(Q) = Q$ si et seulement si la plus grande puissance de X dans la décomposition de Q est Q elle-même, ce qui signifie que Q est une puissance de X.

Définition 1.26. Soient P et Q des séries formelles. Alors on écrit $Pre(P, Q)$ si et seulement si P est un polynômes de degré n, et si les $n+1$ premiers coefficients dans Q sont ceux de P. C'est-à-dire, $Pre(P, Q)$ si et seulement si $P = \sum_{k=0}^{n} a_k X^k$ pour un certain entier naturel n et

$$Q = \sum_{k \geq 0} a_k X^k.$$

On peut définir cette relation par la formule

$$Pre(P, Q) \longleftrightarrow \forall z \left((P_X(z) \wedge (z \preceq \lambda_X(P)) \rightarrow \bigwedge_{k \in \mathbb{F}_p} (X_X(z, P, k) \leftrightarrow X_X(z, Q, k)) \right).$$

Cette formule vérifie que pour chaque puissance de X de degré plus petit ou égal au degré de P, les coefficients dans P et Q sont les mêmes.

Les ensembles $\{Q \in \mathbb{F}_p[\![X]\!] \mid P_X(Q)\}$ et $\{(P,Q) \in (\mathbb{F}_p[\![X]\!])^2 \mid Pre(P,Q)\}$ sont des exemples d'ensembles définissables dans la structure

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot,\cdot,k)\}_{k\in\mathbb{F}_p})$$

selon la définition 1.24.

CHAPITRE II

LES ω-AUTOMATES

Ce chapitre vise à introduire les ω-automates, des automates qui prennent en entrée des mots de longueur infinie, ou ω-mots. Il existe divers types d'ω-automates. Nous utilisons seulement les automates de Büchi, les automates de Muller et les automates de Büchi déterministes.

2.1 Les automates sur les mots infinis

Dans le premier chapitre de ce mémoire, nous avons défini les automates finis. Ces automates prennent en entrée des mots finis. Dans cette section, nous définissons des automates sur des mots infinis, ou *ω-automates*. Les définitions, théorèmes et autres résultats sont tirés de (Thomas, 1990), sauf avis contraire.

Définition 2.1. Soit Σ un ensemble fini de lettres. On note Σ^ω l'ensemble des suites infinies de lettres de Σ. On appelle les éléments de Σ^ω des *mots infinis* ou *ω-mots*. On appelle *ω-langage* tout sous-ensemble de Σ^ω. On note aussi Σ^∞ l'ensemble des mots finis et infinis sur l'alphabet Σ, c'est-à-dire $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$.

Définition 2.2. La classe des *langages ω-rationnels* de Σ^∞ est la plus petite classe $\mathcal{R}$ de parties de Σ^∞ vérifiant

1. $\emptyset \in \mathcal{R}$, $\{a\} \in \mathcal{R}, \forall a \in \Sigma$,
2. $\mathcal{R}$ est fermé par union finie,

3. $\forall X \in \mathcal{P}(\Sigma^*) \cap \mathcal{R}$, $\forall Y \in \mathcal{P}(\Sigma^\infty) \cap \mathcal{R}$, $XY \in \mathcal{R}$,

4. $\forall X \in \mathcal{P}(\Sigma^*) \cap \mathcal{R}, X^* \in \mathcal{R}$ et $X^\infty \in \mathcal{R}$.

On note $Rat(\Sigma^\omega)$ l'ensemble des langages ω-rationnels de Σ^∞ inclus dans Σ^ω.

Définition 2.3. Un *automate de Büchi* est un quintuplet

$$\mathcal{A} = (Q, \Sigma, D, F, \Delta)$$

où Q est un ensemble fini d'états, Σ est un ensemble fini de lettres appelé l'*alphabet* de $\mathcal{A}$, D est un sous-ensemble de Q appelé l'ensemble des *états initiaux*, F est un sous-ensemble de Q appelé l'ensemble des *états acceptants*, et Δ est une relation définie sur $Q \times \Sigma \times Q$ appelée la *relation de transition* de $\mathcal{A}$.

Soit $w = a_0 a_1 a_2 \ldots$ un mot infini sur l'alphabet Σ. Un *chemin* étiqueté par w dans l'automate de Büchi $\mathcal{A}$ est une suite d'états $q_0 q_1 q_2 \ldots$ telle que

- $q_0 \in D$,
- $(q_n, a_n, q_{n+1}) \in \Delta$, pour tout $n \geq 0$.

Le mot infini w est *accepté* par $\mathcal{A}$ s'il existe un chemin étiqueté par w dans $\mathcal{A}$ dans lequel apparait une infinité de fois un état $q_f \in F$.

On peut remarquer que l'unique différence entre la définition d'automate de Büchi et la définition d'automate fini est la condition d'acception.

Définition 2.4. Soit L un ω-langage sur l'alphabet Σ. On dit que L est *ω-reconnaissable* s'il existe un automate de Büchi qui accepte exactement les ω-mots de L.

Théorème 2.5. *Une partie de Σ^ω est ω-reconnaissable si et seulement si elle est ω-rationnelle.*

Pour une preuve de ce théorème, on peut se reporter au théorème 1.1 dans (Thomas, 1990).

Définition 2.6. Un *automate de Muller* est un quintuplet

$$\mathcal{A} = (Q, \Sigma, D, \mathcal{F}, \Delta)$$

où Q est un ensemble fini d'états, Σ est un ensemble fini de lettres appelé l'*alphabet* de $\mathcal{A}$, D est un sous-ensemble de Q appelé l'ensemble des *états initiaux*, $\mathcal{F}$ est un sous-ensemble de $\mathcal{P}(Q)$, et Δ est une relation définie sur $Q \times \Sigma \times Q$ appelée la *relation de transition* de $\mathcal{A}$.

Un mot infini w est *accepté* par l'automate de Muller $\mathcal{A}$ s'il existe un chemin de w dans $\mathcal{A}$ dans lequel apparaissent une infinité de fois exactement les états d'un sous-ensemble d'états $E \in \mathcal{F}$.

L'unique différence entre un automate de Muller et un automate de Büchi est la condition d'acceptation. Un automate de Büchi reconnait un ω-mot si et seulement si un de ses chemins contient une infinité d'états acceptants. Un automate de Muller reconnait un ω-mot si et seulement si l'ensemble d'états parcourus une infinité de fois par un de ses chemins se retrouve dans $\mathcal{F}$.

Malgré cette différence, les ω-langages acceptés par les automates de Büchi coïncident exactement avec les ω-langages acceptés par les automates de Muller.

Théorème 2.7. *(Théorème de McNaughton) Un ω-langage est reconnu par automate de Büchi si et seulement si il est reconnu par automate de Muller.*

Cela signifie qu'un langage est ω-reconnaissable s'il est reconnu par un automate de Büchi, ou un automate de Muller. Il s'agit notamment du théorème 4.4 dans (Thomas, 1990), où l'on peut trouver une preuve complète.

Définition 2.8. Un *automate de Büchi déterministe* est un quintuplet

$$\mathcal{A} = (Q, \Sigma, q_d, F, \delta)$$

où Q est un ensemble fini d'états, Σ est un ensemble fini de lettres appelé l'*alphabet* de $\mathcal{A}$, $q_d \in Q$ est l'*état initial*, $F \subseteq Q$ est l'ensemble des *états acceptants*, et $\delta : Q \times \Sigma \to Q$ est la *fonction de transition* de $\mathcal{A}$.

Soit $w = a_0a_1a_2\ldots$ un mot infini sur l'alphabet Σ. Le *chemin* étiqueté par w dans l'automate $\mathcal{A}$ de w est la suite d'états $q_0q_1q_2\ldots$ telle que

- $q_0 = q_d$,
- $\delta(q_n, a_n) = q_{n+1}$, pour tout $n \geq 0$.

Le mot infini w est *accepté* par $\mathcal{A}$ si son chemin dans $\mathcal{A}$ contient une infinité de fois un état $q_f \in F$.

Contrairement à un automate de Büchi quelconque, un automate de Büchi déterministe doit avoir un unique état initial et sa relation de transition doit être une fonction. Cela signifie que chaque ω-mot n'a qu'un seul chemin dans un automate de Büchi déterministe, alors qu'il peut avoir plusieurs chemins pour un même ω-mot dans un automate de Büchi non-déterministe.

Contrairement aux automates finis, les automates de Büchi déterministes ne reconnaissent pas les mêmes langages que les automates de Büchi. Ils reconnaissent strictement moins de langages. Il s'agit ici d'une propriété bien connue des automates de Büchi. L'automate utilisé dans la démonstration est d'ailleurs un exemple classique d'automate de Büchi qui ne peut pas être rendu déterministe.

Théorème 2.9. *Les automates de Büchi déterministes ne sont pas suffisants pour reconnaitre tous les langages ω-reconnaissables.*

Démonstration. Considérons l'ω-langage $L = \{0,1\}^*0^\omega$ sur l'alphabet $\{0,1\}$. Il s'agit de l'ensemble des ω-mots qui ont une queue de zéros. On remarque que l'automate de Büchi de la figure 2.1 reconnait ce langage.

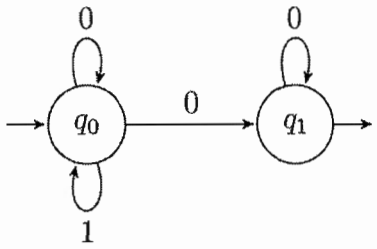

Figure 2.1 Automate de Büchi reconnaissant $\{0,1\}^*0^\omega$.

Supposons qu'un automate de Büchi déterministe $\mathcal{A} = (Q, \{0,1\}, q_d, F, \delta)$ reconnait L. L'automate $\mathcal{A}$ reconnait 0^ω, car $0^\omega \in L$. Cela signifie que 0^ω parcourt un état dans $q'_0 \in F$ une infinité de fois. Notamment, q'_0 est visité après la k_0^e lettre. L'automate $\mathcal{A}$ reconnait également $0^{k_0}10^\omega \in L$. Selon le même raisonnement, l'état q'_1 est visité une infinité de fois. Cela signifie que l'état q'_1 est visité après le préfixe $0^{k_0}10^{k_1}$, pour un entier k_1.

On construit ainsi récursivement le mot infini $w = 0^{k_0}10^{k_1}10^{k_2}\cdots$. Le chemin étiqueté par w dans $\mathcal{A}$ parcourt une infinité d'états de F. L'automate $\mathcal{A}$ reconnait alors w, mais $w \notin L$. Ce qui contredit la supposition que $\mathcal{A}$ reconnait L. Il ne peut donc pas y avoir d'automate de Büchi déterministe qui reconnait le langage ω-reconnaissable L. □

Théorème 2.10 ((Büchi, 1960b), lemme 10). *Un langage L est ω-reconnaissable si et seulement s'il est de la forme*

$$L = \bigcup_{i=1}^{n} X_i Y_i^\omega$$

où X_i et Y_i sont des langages rationnels pour tout i entre 1 et n.

Ce théorème nous permet de caractériser l'écriture rationnelle d'un langage ω-reconnaissable.

2.2 Propriétés de fermeture des automates de Büchi

Dans cette section, nous démontrons les diverses propriétés de fermeture des automates de Büchi. Tout au long de la section, nous considérons les automates de Büchi

$$\mathcal{A} = (Q_A, \Sigma, D_A, F_A, \Delta_A) \text{ et } \mathcal{B} = (Q_B, \Sigma, D_B, F_B, \Delta_B),$$

où Σ est leur alphabet, Q_A et Q_B sont leur ensemble d'états respectif, D_A et D_B sont leurs ensembles d'états initiaux, F_A et F_B sont leur ensemble d'états finaux et Δ_A et Δ_B sont leur relation de transition.

Les langages $L(\mathcal{A})$ et $L(\mathcal{B})$ sont les ω-langages reconnus par les automates $\mathcal{A}$ et $\mathcal{B}$.

Proposition 2.11. *Les langages ω-reconnaissables sont fermés par union.*

Démonstration. Considérons les langages ω-reconnaissables $L(\mathcal{A})$ et $L(\mathcal{B})$, nous cherchons à montrer qu'il existe un automate de Büchi qui reconnait le langage $L(\mathcal{A}) \cup L(\mathcal{B})$.

On peut supposer que $Q_A \cap Q_B = \emptyset$, en ré-étiquetant les états de $\mathcal{B}$ au besoin. Ainsi, l'automate $\mathcal{A}' = (Q_A \cup Q_B, \Sigma, D_A \cup D_B, F_A \cup F_B, \Delta_A \cup \Delta_B)$ reconnait $L(\mathcal{A}) \cup L(\mathcal{B})$. En effet, un ω-mot de $L(\mathcal{A})$ sera reconnu par cet automate puisque les transitions de $\mathcal{A}$ sont inchangée. De même, les ω-mots de $L(\mathcal{B})$ sont aussi reconnus. Inversement, un ω-mot reconnu par cet automate de Büchi est dans $L(\mathcal{A})$ ou $L(\mathcal{B})$. □

Proposition 2.12. *Les langages ω-reconnaissables sont fermés par intersection.*

Démonstration. Considérons les langages ω-reconnaissables $L(\mathcal{A})$ et $L(\mathcal{B})$, nous cherchons à montrer qu'il existe un automate de Büchi qui reconnait le langage $L(\mathcal{A}) \cap L(\mathcal{B})$.

Un automate de Büchi qui reconnait $L(\mathcal{A}) \cap L(\mathcal{B})$ est $\mathcal{A}' = (Q', \Sigma, D', F', \Delta')$, où $Q' = Q_A \times Q_B \times \{1,2\}$, $D' = D_A \times D_B \times \{1\}$, $F' = \{(q_A, q_B, 2) : q_B \in F_B\}$ et $\Delta' = \Delta_{1,1} \cup \Delta_{1,2} \cup \Delta_{2,1} \cup \Delta_{2,2}$ où les relations de transitions $\Delta_{i,j}$ sont définis comme suit :

$$\Delta_{1,1} = \{((q_A, q_B, 1), x, (q'_A, q'_B, 1)) : (q_A, x, q'_A) \in \Delta_A, (q_B, x, q'_B) \in \Delta_B, q_A \notin F_A\}$$

$$\Delta_{1,2} = \{((q_A, q_B, 1), x, (q'_A, q'_B, 2)) : (q_A, x, q'_A) \in \Delta_A, (q_B, x, q'_B) \in \Delta_B, q_A \in F_A\}$$

$$\Delta_{2,1} = \{((q_A, q_B, 2), x, (q'_A, q'_B, 1)) : (q_A, x, q'_A) \in \Delta_A, (q_B, x, q'_B) \in \Delta_B, q_B \in F_B\}$$

$$\Delta_{2,2} = \{((q_A, q_B, 2), x, (q'_A, q'_B, 2)) : (q_A, x, q'_A) \in \Delta_A, (q_B, x, q'_B) \in \Delta_B, q_B \notin F_B\}.$$

Si nous sommes dans un état $(q_A, q_B, 1)$, alors en lisant x, nous allons dans un état $(q'_A, q'_B, 2)$ si $q_A \in F_A$, et nous allons dans un état $(q'_A, q'_B, 1)$ sinon. Si nous sommes dans un état $(q_A, q_B, 2)$, alors en lisant x, nous allons dans un état $(q'_A, q'_B, 1)$ si $q_B \in F_B$, et nous allons dans un état $(q'_A, q'_B, 2)$ sinon. De plus, les transitions doivent respecter les conditions $(q_A, x, q'_A) \in \Delta_A$ et $(q_B, x, q'_B) \in \Delta_B$.

Considérons à présent un ω-mot w sur l'alphabet Σ. Par construction de l'automate $\mathcal{A}'$, un chemin $c' = (q_{A,0}, q_{B,0}, i_0)(q_{A,1}, q_{B,1}, i_1) \cdots$ est parcouru par w dans $\mathcal{A}'$ si et seulement si $c_A = q_{A,0} q_{A,1} \cdots$ est un chemin parcouru par w dans $\mathcal{A}$ et $c_B = q_{B,0} q_{B,1} \cdots$ est un chemin parcouru par w dans $\mathcal{B}$. De plus, les chemin c_A et c_B sont acceptants si et seulement si c' est la concaténation infinie de segments finis d'états « 1 » et d'états « 2 » alternativement. En effet, on passe d'un état « 1 » à un état « 2 » seulement lorsque l'on vient de passer par un état acceptant de $\mathcal{A}$, et nous passons d'un état « 2 » à un état « 1 » que lorsque l'on vient de passer par un état acceptant de $\mathcal{B}$. Ainsi, si l'on passe d'un état « 1 » à un état « 2 » et vice-versa une infinité de fois, alors w parcourt un chemin acceptant dans $\mathcal{A}$ et $\mathcal{B}$. Cette dernière condition est vraie si et seulement si w passe une infinité de fois par les états de F', étant ainsi accepté par $\mathcal{A}'$.

Nous avons donc construit un automate de Büchi qui reconnait $L(\mathcal{A}) \cap L(\mathcal{B})$. □

Proposition 2.13. *Les langages ω-reconnaissables sont fermés par projection.*

Démonstration. Considérons le langage ω-reconnaissable $L(\mathcal{C})$, où

$$\mathcal{C} = (Q_C, \Sigma^d, D_C, F_C, \Delta_C)$$

avec un entier $d \geq 2$. Les lettres des ω-mots de $L(\mathcal{C})$ sont donc des d-uplets. Nous cherchons à montrer qu'il existe un automate de Büchi qui reconnait $\pi_k(L(\mathcal{C}))$, où π_k est la projection des d-uplets qui omet la k^e composante. Désignons la projection de $(x_1, \ldots, x_d)$ qui omet la k^e composante par $(x_1, \ldots, \widehat{x_k}, \ldots, x_d)$.

Un automate de Büchi qui reconnait $\pi_k(L(\mathcal{C}))$ est $\mathcal{C}' = (Q', \Sigma^{d-1}, D', F', \Delta')$, où $Q' = Q_C$, $D' = D_C$, $F' = F_C$ et

$$\Delta' = \{(q, (x_1, \ldots, \widehat{x_k}, \ldots, x_d), q') \mid \exists y, (q, (x_1, \ldots, x_{k-1}, y, x_{k+1}, \ldots, x_d), q') \in \Delta_C\}.$$

L'automate de Büchi $\mathcal{C}'$ ainsi construit reconnait exactement les ω-mots de $L(\mathcal{C})$ auxquelles nous avons enlevé la k^e composante. L'ensemble $\pi_k(L(\mathcal{C}))$ est donc ω-reconnaissable. □

Proposition 2.14. *Les langages ω-reconnaissables sont fermés par concaténation à gauche de langages rationnels.*

Démonstration. Considérons le langage ω-reconnaissable $L(\mathcal{A})$. Considérons aussi un automate fini $\mathcal{C} = (Q_C, \Sigma, D_C, F_C, \Delta_C)$ qui reconnait le langage $L(\mathcal{C})$. Nous cherchons un automate de Büchi qui reconnait $L(\mathcal{C}) \cdot L(\mathcal{A})$.

Supposons $Q_A \cap Q_C = \emptyset$, nous pouvons ré-étiqueter les états de $\mathcal{C}$ au besoin. Construisons alors l'automate de Büchi $\mathcal{A}' = (Q_A \cup Q_C, \Sigma, D', F_A, \Delta')$, où $D' = D_C$ si $D_C \cap F_C = \emptyset$, et $D' = D_A \cup D_C$ sinon. Les transitions sont définies comme suit :

$$\Delta' = \Delta_A \cup \Delta_C \cup \{(q, x, q') : q' \in D_A \text{ et } \exists q_f \in F_C \text{ tel que } (q, x, q_f) \in \Delta_C\}.$$

Les nouvelles transitions dans Δ' nous font passer par les états initiaux de $\mathcal{A}$ lorsque nous lisons potentiellement la lettre finale d'un mot accepté par $\mathcal{C}$. Ainsi, l'automate $\mathcal{A}'$ reconnait un ω-mot w si et seulement si sa première partie (finie) est un mot de $L(\mathcal{C})$ et sa seconde partie (infinie) est un ω-mot de $L(\mathcal{A})$. □

Proposition 2.15. *L'ω-itération d'un langage rationnel qui ne contient pas le mot vide est un langage ω-reconnaissable.*

Démonstration. Considérons un automate fini $\mathcal{C} = (Q_C, \Sigma, D_C, F_C, \Delta_C)$ qui reconnait le langage $L(\mathcal{C})$ qui ne contient pas le mot vide. Nous cherchons un automate de Büchi qui reconnait $L(\mathcal{C})^\omega$.

Sans perte de généralité, par la proposition 1.8, nous pouvons considérer que $\mathcal{C}$ n'a qu'un seul état initial q_d qui est une source et un seul état final q_f qui est un puits. Il n'y a pas de transition qui entre dans q_d et il n'y a pas de transition qui sort de q_f. En particulier, comme le mot vide n'est pas accepté, $q_d \neq q_f$.

Nous construisons alors l'automate de Büchi $\mathcal{A}' = (Q_C, \Sigma, q_d, q_f, \Delta_C \cup \Delta')$ où $\Delta' = \{(q, x, q_d) : (q, x, q_f) \in \Delta_C\}$.

Nous avons pris l'automate $\mathcal{C}$, et à chaque état où il y a une transition vers l'état final, nous ajoutons une transition vers l'état initial. Ainsi, un ω-mot w parcourt un chemin dans $\mathcal{A}'$ qui passe une infinité de fois par q_f si et seulement si w est la concaténation infinie de mots reconnus par $\mathcal{C}$. Donc, $w \in L(\mathcal{C})^\omega$. □

Proposition 2.16. *Les langages ω-reconnaissables sont fermés par le complément.*

Il s'agit du théorème 2.1 dans (Thomas, 1990). Une preuve complète s'y retrouve. Elle est significativement plus longue que les preuves précédentes et elle nécessite la définition de concepts qui ne sont pas réutilisés dans ce mémoire.

2.3 Lemme d'itération pour les ω-langages

Dans cette section, nous énonçons et démontrons un résultat sur les ω-langages qui est semblable au lemme d'itération (théorème 1.9) dans les langages de mots finis. Nous décidons de lui donner le nom de *lemme d'itération oméga* (ou *ω-Pumping Lemma.*) Il ne s'agit pas d'un nouveau résultat, l'énoncé (sans preuve) se retrouve notamment sous une forme similaire dans (Alur et al., 2009), §3. lemme 1. La démonstration rédigée dans ce mémoire est inspirée de la démonstration du théorème 1.9.

Théorème 2.17. *(Lemme d'itération oméga) Soit L un ω-langage reconnaissable. Alors il existe $N > 0$ un entier tel que pour tout $w = u_1w_1u_2w_2...u_iw_i...$ dans L, avec $|w_i| \geq N$ pour tout $i \geq 1$, on peut écrire $w_i = x_iy_iz_i$ tel que*

1. $|y_i| \geq 1$,
2. $|x_iy_i| \leq N$ *et*
3. $\forall (j_i)_{i\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$, $u_1x_1y_1^{j_1}z_1...u_ix_iy_i^{j_i}z_i... \in L$.

Démonstration. Considérons L un ω-langage reconnu par un automate de Büchi à n états. Posons $N = 2n$ et prenons $w \in L$ tel que $w = u_1w_1u_2w_2...u_iw_i...$, avec $|w_i| \geq N$ pour tout $i \geq 1$. Comme $w \in L$, il existe un état final de l'automate qui est parcouru une infinité de fois, notons F cet état.

Comme $|u_i| \geq N$, le chemin dans l'automate parcouru par w_i passe par au moins $2n+1$ états. Il y a donc un état de l'automate qui est visité 3 fois par ce chemin. Soit la séquence des états parcourus $q_0q_1...q_{|w_i|-1}$. Nous pouvons noter dans cette

séquence les occurrences de l'état parcouru 3 fois par q, q' et q''. Ainsi, le parcours du chemin peut s'écrire $q_0 q_1 ... q ... q' ... q'' ... q_{|w_i|-1}$.

Notons $\overline{y_i}$ le mot effectuant le parcours de q à q', et notons $\overline{\overline{y_i}}$ le mot effectuant le parcours de q' à q''. Remarquons que chacun est un sous-mot de w_i.

Si $\overline{y_i}$ passe par l'état F, alors on pose $y_i = \overline{\overline{y_i}}$. Sinon, on pose $y_i = \overline{y_i}$. Nous avons ainsi décomposé chaque w_i en $x_i y_i z_i$, où x_i et z_i découlent du choix de y_i.

Remarquons que $|y_i| \geq 1$ puisque le mot y_i code une boucle non-vide autour d'un état. De plus, on remarque que $|x_i y_i| \leq N$. En effet, le dernier état possiblement visité par y_i est q'', il s'agit de l'état qui a été visité 3 fois, et il s'agit de la troisième visite de cet état. Il est certain que cette visite a eu lieu avant N lectures des lettres de w_i.

Maintenant, nous devons vérifier que nous pouvons itérer chaque y_i et rester dans L. En d'autres mots, on veut $w' \in L$ pour tout $w' = u_1 w'_1 u_2 w'_2 ... u_i w'_i ...$ où $w'_i = x_i y_i^{j_i} z_i$ pour tout $i \geq 1$, pour tout $j_i \in \mathbb{N}$.

Remarquons d'abord que puisque chaque y_i commence et finit sa lecture dans un même état, il est clair que nous pouvons l'itérer ou le retirer de w et garder l'étiquette d'un chemin dans l'automate. C'est-à-dire que si w est l'étiquette d'un chemin dans l'automate, alors w' est aussi l'étiquette d'un chemin dans l'automate. Il nous reste à vérifier que ce chemin est acceptant.

On peut remarquer que si un w_i passe par l'état final F, alors w'_i passe également par l'état F. Et ce, même si $w'_i = x_i z_i$ grâce au choix judicieux de y_i. Nous avons donc $w = u_1 w_1 ... u_i w_i ...$ et $w' = u_1 w'_1 ... u_i w'_i ...$ tels que si w_i visite F, alors w'_i visite F. Ainsi, si w visite une infinité de fois F, alors il en sera de même pour w'. Donc $w' \in L$. □

On peut observer un cas particulier où u_1 est le mot vide et $j_i = 1$ pour tout $i > 1$ et ainsi obtenir facilement le résultat plus simple suivant.

Corollaire 2.18. *Soit L un ω-langage reconnaissable. Alors il existe $N > 0$ un entier tel que pour tout $w = w_0\tilde{w}$ dans L, où w_0 est un mot de longueur $|w_0| \geq N$ et $\tilde{w}$ est un ω-mot, on peut écrire $w_0 = xyz$ tel que*

1. $|y| \geq 1$,
2. $|xy| \leq N$ *et*
3. $xy^n z\tilde{w} \in L, \forall n \in \mathbb{N}$.

Le lemme d'itération oméga nous permet, dans plusieurs cas, de démontrer qu'un ω-langage n'est pas ω-reconnaissable.

Proposition 2.19. *Soit $\mathbb{F}_p[\![X]\!]$ l'ensemble des séries formelles avec coefficients dans le corps fini $\mathbb{F}_p$, et soit P_X l'ensemble des puissances de X. Soit σ la fonction de multiplication à gauche par une puissance de X, c'est-à-dire*

$$\begin{aligned} \sigma : \quad P_X \times \mathbb{F}_p[\![X]\!] &\longrightarrow \mathbb{F}_p[\![X]\!] \\ (z, v) &\longmapsto zv. \end{aligned}$$

Alors le graphe de la fonction σ n'est pas ω-reconnaissable.

Démonstration. Nous voulons démontrer que l'ensemble $L = \{(z, v, \sigma(z, v)) \mid z \in P_X, v \in \mathbb{F}_p[\![X]\!]\} \subseteq P_X \times \mathbb{F}_p[\![X]\!] \times \mathbb{F}_p[\![X]\!]$ n'est pas ω-reconnaissable. On utilise le lemme d'itération oméga.

Par contradiction, supposons que L est ω-reconnaissable. Par le lemme d'itération oméga, il existe alors un nombre entier N tel que pour tout ω-mot $w \in L$ décomposé de telle sorte que $w = u_1 w_1 \cdots u_i w_i \cdots$, avec $|w_i| \geq N$ pour chaque i, on peut écrire $w_i = x_i y_i z_i$ tel que

1. $|y_i| \geq 1$

2. $|x_i y_i| \leq N$

3. $\forall (j_i)_{i \in \mathbb{N}} \in \mathbb{N}^{\mathbb{N}}, u_1 w_1' \cdots u_i w_i' \cdots \in L$, où $w_i' = x_i y_i^{j_i} z_i$ pour tout i.

La fonction σ prend en entrée une série formelle et une puissance de X et les multiplie. Dans notre langage, la fonction prend en entrée un ω-mot v et un ω-mot z, où z a la lettre 1 à une seule position et la lettre 0 partout ailleurs. La fonction *décale* v. Plus précisément, la sortie est v auquel on a concaténé à gauche autant de 0 qu'il y a dans z avant la lettre 1. Par exemple, si la série formelle est $2 + 2X + 2X^2 + \cdots$ et la puissance de X est X^3, alors on a

$$z = 000100\cdots$$
$$v = 222222\cdots$$
$$z \cdot v = 000222\cdots$$

et ainsi l'ω-mot dans L est le triplet

$$(000100\cdots, 222222\cdots, 000222\cdots).$$

Prenons à présent un ω-mot quelconque dans L, notons-le (z, v, zv). Disons que z représente X^N et que $v = a_0 a_1 a_2 \cdots$ représente une série sans coefficient nul (pour simplifier.) Le triplet que nous étudions est alors de la forme suivante :

$$w = (\underbrace{0\cdots0}_{N \text{ fois}} 10\cdots, a_0 a_1 a_2 \cdots, \underbrace{0\cdots0}_{N \text{ fois}} a_0 a_1 a_2 \cdots).$$

Rappelons que nous pouvons décomposer w sous la forme $u_1 w_1 \cdots u_i w_i \cdots$ tant que chaque w_i est de longueur au moins N. Encore une fois, pour simplifier, nous ne regardons que w_1, et nous posons $|u_1| = 0$ et $|w_1| = N$. Ceci est suffisant pour notre preuve. Alors $w = w_1 \tilde{w}$, où $w_1 = (\underbrace{0\cdots0}_{N \text{ fois}}, a_0 \cdots a_{N-1}, \underbrace{0\cdots0}_{N \text{ fois}})$ et $\tilde{w} = (10\cdots, a_N a_{N+1} \cdots, a_0 a_1 \cdots)$.

Selon le lemme d'itération oméga, on peut trouver une décomposition de $w_1 = x_1 y_1 z_1$ tel que

1. $|y_1| \geq 1$,
2. $|x_1 y_1| \leq N$
3. $x_1 y_1^{j_1} z_1 \tilde{w} \in L$, pour tout $j_1 \in \mathbb{N}$.

Étant donné la manière dont nous avons représenté w, il est clair que nous aurons la décomposition $w_1 = x_1 y_1 z_1$ suivante :

$$x_1 = (0 \cdots 0, a_0 \cdots a_{s-1}, 0 \cdots 0)$$
$$y_1 = (0 \cdots 0, a_s \cdots a_{r-1}, 0 \cdots 0)$$
$$z_1 = (0 \cdots 0, a_r \cdots a_N, 0 \cdots 0).$$

On voit en particulier que $|x_1| = s$, $|y_1| = r - s$ et $|z_1| = N - r$. Regardons le cas où $j_1 = 0$. Nous avons ainsi $w_1' = x_1 z_1$. Donc, $w' = w_1' \tilde{w}$ est le triplet :

$$(\underbrace{0 \cdots 0}_{N-r+s \text{ fois}} 10 \cdots, a_0 a_1 \cdots a_{s-1} a_r \cdots, \underbrace{0 \cdots 0}_{N-r+s \text{ fois}} a_0 a_1 \cdots a_{s-1} a_s a_{s+1} \cdots).$$

Ici, on voit facilement que le triplet ne fait pas partie de L. Si nous posons z' comme étant le premier terme du triplet, et v' comme étant le second terme, alors selon la définition de la fonction étudiée, nous devrions avoir le triplet $(z', v', \sigma(z', v'))$. Or, nous avons le triplet $(z', v', \sigma(z', v))$ qui n'est évidemment pas dans le graphe de σ en général. On remarque que nous allons rencontrer ce problème, peu importe la longueur non-nulle choisie pour y_1.

Ainsi, pour le mot infini posé $w \in L$, il existe une suite d'exposants $(j_i)_{i \in \mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$ telle que $w' = u_1 w_1' \cdots u_i w_i' \cdots \notin L$, où $w_i' = x_i y_i^{j_i} z_i$. La suite que nous avons trouvée est $(0, 1, 1, \ldots)$. Cela contredit la troisième propriété du lemme d'itération oméga et ainsi le graphe de la fonction σ n'est pas ω-reconnaissable. □

Proposition 2.20. *Le graphe de la fonction de Frobenius*

$$\begin{aligned} Frob_p : \ \mathbb{F}_p[\![X]\!] &\to \mathbb{F}_p[\![X]\!] \\ Q &\mapsto Q^p \end{aligned}$$

n'est pas reconnaissable par automate de Büchi.

Démonstration. Pour démontrer ce résultat, utilisons le corollaire 2.18.

Soit $Q = a_0 + a_1X + a_2X^2 + \ldots$ une série formelle dans $\mathbb{F}_p[\![X]\!]$. Nous représentons une telle série dans $\mathbb{F}_p[X]^\omega$ par $a_0a_1a_2\ldots$ En particulier, un polynôme sera représenté par le mot infini $w = a_0a_1\ldots a_n000\ldots$ où n est le degré du polynôme. Notons de plus que l'image d'un tel polynôme par $Frob_p$ est représentée par

$$Frob_p(w) = a_0\underbrace{0\ldots0}_{p \text{ fois}}a_1\underbrace{0\ldots0}_{p \text{ fois}}a_20\ldots a_n000\ldots$$

Procédons par l'absurde et supposons que $L = \text{Graph}(Frob_p)$ est ω-rationnel, et soit $N+1$ sa constante d'itération. Considérons $w \in L$ la représentation d'un polynôme de degré N et de son image. Plus précisément, on a

$$w = (a_0a_1\ldots a_N0\ldots, a_0\underbrace{0\ldots0}_{p \text{ fois}}a_10\ldots a_N000\ldots)$$

et supposons chaque a_i non-nul.

Par le corollaire 2.18, pour tout mot w_0 tel que $w = w_0\tilde{w}$ et $|w_0| \geq N+1$, il existe un mot y tel que

- $w_0 = xyz$,
- $|y| \geq 1$,
- $|xy| \leq N+1$
- $xy^nz\tilde{w} \in L, \forall n \geq 0$

Entre autres, on peut prendre $w_0 = (a_0a_1\ldots a_N, a_00\ldots)$. Posons $w_0 = xyz$, selon le corollaire 2.18 on a $xy^0z\tilde{w} \in L$ et $xy^2z\tilde{w} \in L$.

Si $|y| > 1$: Notons $xy^0z\tilde{w} = (v, v')$.

Alors v possède strictement moins de lettres non-nulles que v'. En effet, En enlevant y à w_0, on se trouve à n'enlever que des termes non-nuls dans

la composantes de gauche et au moins un terme nul dans la composante de droite. Cela vient du fait que l'image par $Frob_p$ n'admet jamais deux termes non-nuls consécutifs.

Or, $Frob_p$ laisse constant le nombre de termes non-nuls. Donc $xy^0z\tilde{w} \notin L$.

Si $|y| = 1$: Dans ce cas, y est un couple de lettres.

Si $y = (a_i, 0)$, alors $xy^0z\tilde{w} \notin L$ pour les mêmes raisons que dans le cas précédent. C'est-à-dire que le retrait de y ne garde pas constant le nombre de termes non-nuls.

Si $y = (a_i, a_j)$, où $a_i \neq 0$ et $a_j \neq 0$, alors on a $xy^2z\tilde{w} \notin L$. En effet, notons $xy^2z\tilde{w} = (v, v')$. On obtient le sous-mot a_ia_j dans v', or il est impossible dans l'image de $Frob_p$ d'avoir deux lettres consécutives non-nulles. Donc $xy^2z\tilde{w} \notin L$.

Donc L n'est pas ω-rationnel. □

2.4 Reconnaissabilité des prédicats

Cette section contient les automates de Büchi reconnaissant chaque prédicat de la structure

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k)\}_{k \in \mathbb{F}_p}).$$

Les prédicats de cette structure ont été définis au chapitre I. Cette section montre que chaque ensemble défini par une formule atomique est ω-reconnaissable.

Les ω-automates lisent des ω-mots, c'est-à-dire des suites infinies de lettres. Or, nous parlons de reconnaissabilité de sous-ensembles de $\mathbb{F}_p[\![X]\!]$. Pour qu'un ω-automate puisse lire une série formelle, nous codons une série $a_0 + a_1X + a_2X^2 + \cdots$ par l'ω-mot $a_0a_1a_2 \cdots \in \mathbb{F}_p^\omega$.

Pour qu'un ω-automate puisse lire un d-uplet dans $(\mathbb{F}_p[\![X]\!])^d$, nous codons un d-uplet $(P_1, P_2, \ldots, P_d)$ par le mot infini $(a_{1,0}, a_{2,0}, \ldots, a_{d,0})(a_{1,1}, a_{2,1}, \ldots, a_{d,1})\cdots$ dans $((\mathbb{F}_p)^d)^\omega$, où $P_i = \sum a_{i,j}X^j$ pour chaque i.

Exemple 2.21. Soit $(P, Q) \in (\mathbb{F}_p[\![X]\!])^2$ tel que $P = a_0 + a_1X + a_2X^2 + \cdots$ et $Q = b_0 + b_1X + b_2X^2 + \cdots$. Alors on représente le couple (P, Q) dans un ω-langage sur $(\mathbb{F}_p)^2$ par

$$(a_0, b_0)(a_1, b_1)(a_2, b_2)\cdots$$

Proposition 2.22. *Le graphe de la fonction* $+ : (\mathbb{F}_p[\![X]\!])^2 \to \mathbb{F}_p[\![X]\!]$ *est* ω*-reconnaissable.*

Démonstration. Il suffit de construire un automate de Büchi reconnaissant

$$\{(A, B, C) \in (\mathbb{F}_p[\![X]\!])^3 \mid A + B = C\}.$$

L'addition sur les séries formelles se fait terme à terme, c'est-à-dire que si $A = \sum a_nX^n$ et $B = \sum b_nX^n$, alors

$$A + B = \sum_{n\geq 0}(a_n + b_n)X^n.$$

Prenons l'automate $\mathcal{A} = (\{q_0\}, \mathbb{F}_p^3, \{q_0\}, \{q_0\}, \delta)$, où

$$\delta = \{(q_0, (a, b, a + b), q_0) \mid a, b \in \mathbb{F}_p\}.$$

Puisque les coefficients proviennent d'un corps fini, nous savons que δ est fini. Les seuls triplets (A, B, C) qui sont acceptés sont ceux où chaque lettre est de la forme $(a, b, a + b)$. Ainsi (A, B, C) est reconnu par $\mathcal{A}$ si et seulement si $A + B = C$. □

Proposition 2.23. *Le graphe de la fonction* $\cdot X : \mathbb{F}_p[\![X]\!] \to \mathbb{F}_p[\![X]\!]$ *est* ω*-reconnaissable.*

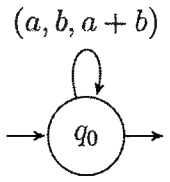

Figure 2.2 Automate de Büchi pour l'addition

Démonstration. Nous cherchons à construire un automate de Büchi reconnaissant l'ensemble $\{(P, Q) \in (\mathbb{F}_p[\![X]\!])^2 \mid P \cdot X = Q\}$. Posons $P = a_0 + a_1X + a_2X^2 + \cdots$ et $Q = b_0 + b_1X + b_2X^2 + \cdots$.

On peut remarquer que $Q = P \cdot X$ si et seulement si $b_0 = 0$ et $b_n = a_{n-1}$ pour tout $n \geq 1$. Un automate de Büchi reconnaissant exactement les paires de séries formelles ayant cette propriété est le suivant :

$$\mathcal{A} = (Q, (\mathbb{F}_p)^2, \{d\}, F, \delta),$$

où

- $Q = \{\overline{0}, \overline{1}, \ldots, \overline{p-1}, d\}$,
- $F = \{\overline{0}, \overline{1}, \ldots, \overline{p-1}\}$,
- $\delta = \{(d, (x, 0), \overline{x}) \mid x \in \mathbb{F}_p\} \cup \{(\overline{x}, (y, x), \overline{y}) \mid x, y \in \mathbb{F}_p\}$.

Chaque état non-initial de $\mathcal{A}$ est associé à un élément de $\mathbb{F}_p$. Si un chemin passe par un état $\overline{x}$, c'est que la dernière lettre lue dans P est x. Ainsi, chaque état garde en mémoire la dernière lettre lue dans P, et donc la prochaine lettre à lire dans Q pour que (P, Q) soit accepté. Nous sommes aussi assurés que la première lettre de Q sera 0, d'où chaque transition sortant de l'état initial doit être de la forme $(d, (x, 0), \overline{x})$.

Par la construction de cet automate, chaque paire (P, Q) sera acceptée si et seulement si elle remplit les conditions pour être dans le graphe de $\cdot X$. □

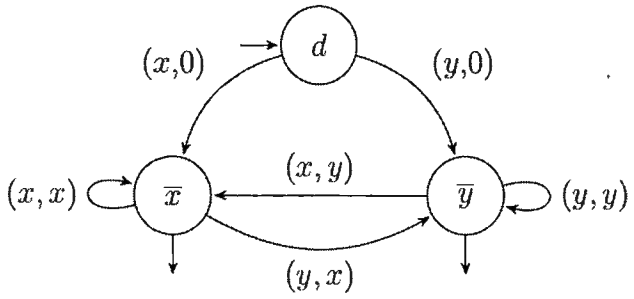

Figure 2.3 Automate de Büchi pour $\cdot X$

Proposition 2.24. *L'ensemble des polynômes est ω-reconnaissable.*

Démonstration. Nous cherchons à construire un automate de Büchi reconnaissant l'ensemble $\{P \in \mathbb{F}_p[\![X]\!] \mid P \in \mathbb{F}_p[X]\}$. Posons $P = a_0 + a_1X + a_2X^2 + \cdots$.

On peut remarquer que P sera un polynôme si et seulement si il existe un entier N tel que $a_n = 0$ pour tout $n > N$. Essentiellement, N est le degré de P dans ce cas. Il suffit alors de construire un automate de Büchi qui reconnait exactement les ω-mots avec une queue de 0.

Cet automate est

$$\mathcal{A} = (Q, \mathbb{F}_p, \{q_0\}, \{q_1\}, \Delta),$$

où

- $Q = \{q_0, q_1\}$,
- $\Delta = \{(q_0, x, q_0) \mid x \in \mathbb{F}_p\} \cup \{(q_0, 0, q_1), (q_1, 0, q_1)\}$.

Un chemin dans cet automate visitera une infinité de fois q_1 si et seulement si son étiquette possède une queue de 0, ce qui est équivalent à être un polynôme. □

Proposition 2.25. *La relation $\prec$ est ω-reconnaissable.*

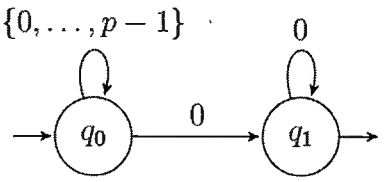

Figure 2.4 Automate de Büchi pour $\mathbb{F}_p[X]$

Démonstration. Nous allons construire un automate de Büchi reconnaissant l'ensemble $\{(P,Q) \in (\mathbb{F}_p[\![X]\!])^2 \mid P \prec Q\}$. Rappelons que $P \prec Q$ si et seulement si P et Q sont des polynômes, et si le degré de P est plus petit que le degré de Q.

Considérons un couple de séries (P,Q). Pour être dans cet ensemble, P et Q doivent avoir un nombre fini de coefficients non-nuls, donc les ω-mots qui les encodent doivent chacun avoir une queue de 0. De plus, le dernier coefficient non-nul de P doit apparaître plus tôt que le dernier coefficient non-nul de Q. Nous proposons l'automate $\mathcal{A} = (Q, \mathbb{F}_p^2, \{q_0\}, \{q_2\}, \delta)$, où

$$\begin{aligned} Q &= \{q_0, q_1, q_2\}, \\ \delta &= \{(q_0, (x,y), q_0) \mid x, y \in \mathbb{F}_p\} \cup \\ &\quad \{(q_0, (0,y'), q_1), (q_1, (0,y'), q_1) \mid y' \in \mathbb{F}_p \setminus \{0\}\} \cup \\ &\quad \{(q_1, (0,0), q_2), (q_2, (0,0), q_2)\}. \end{aligned}$$

Soit w un ω-mot qui code le couple de séries formelles (P,Q). Le chemin étiqueté par w est acceptant dans $\mathcal{A}$ que s'il visite l'état q_2 une infinité de fois. Comme toutes les transitions vers q_2 sont étiquetée $(0,0)$, cela signifie que w encode un couple de polynômes. De plus, ce chemin doit d'abord visiter l'état q_1, qui indique que le dernier coefficient non-nul de P est apparu plus tôt que le dernier coefficient non-nul de Q jusqu'à présent. Ainsi, w est accepté par $\mathcal{A}$ que s'il code un couple de polynômes (P,Q) où le degré de P est plus petit que le degré de Q. □

Proposition 2.26. *Le graphe de la fonction $\lambda_X : \mathbb{F}_p[\![X]\!] \to \mathbb{F}_p[\![X]\!]$ est ω-reconnaissable.*

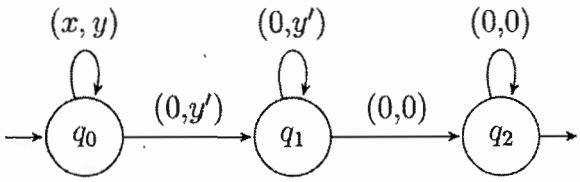

Figure 2.5 Automate de Büchi pour $\prec$

Démonstration. Nous allons construire un automate de Büchi reconnaissant l'ensemble $\{(P,Q) \in (\mathbb{F}_p[\![X]\!])^2 \mid \lambda_X(P) = Q\}$. Rappelons que la fonction λ_X est définie par :

$$\lambda_X(P) = \begin{cases} 0, & \text{si } P = 0 \\ 0, & \text{si } P \text{ n'est pas un polynôme} \\ X^n, & \text{si } P \text{ est un polynôme non-nul de degré } n. \end{cases}$$

Nous proposons l'automate $\mathcal{A} = (Q, \mathbb{F}_p^2, \{q_0\}, F, \delta)$, où

$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_3\}, \\ F &= \{q_0, q_1, q_2\}, \\ \delta &= \{(q_0,(0,0),q_0),(q_2,(0,0),q_2),(q_1,(0,0),q_3),(q_3,(0,0),q_3)\} \cup \\ &\quad \{(q_0,(a,0),q_1),(q_1,(a,0),q_1),(q_3,(a,0),q_1) \mid a \in \mathbb{F}_p \setminus \{0\}\} \cup \\ &\quad \{(q_0,(a,1),q_2),(q_1,(a,1),q_2),(q_3,(a,1),q_2) \mid a \in \mathbb{F}_p \setminus \{0\}\}. \end{aligned}$$

Considérons l'entrée d'un ω-mot (P,Q) dans notre automate $\mathcal{A}$. Si $P = 000\ldots$, alors (P,Q) est accepté si et seulement si $Q = 000\ldots$ aussi, (P,Q) visite dans ce cas l'état q_0 une infinité de fois.

Si P n'est pas un polynôme, c'est-à-dire que P contient une infinité de coefficients a non-nuls, alors l'automate accepte (P,Q) si et seulement si $Q = 000\ldots$, dans ce cas (P,Q) visite l'état q_1 une infinité de fois.

Si P est un polynôme non nul de degré n, c'est-à-dire que $P = a_1a_2\cdots a_n000\ldots$ où a_n est non-nul, alors (P,Q) est accepté si et seulement si Q ne contient que des

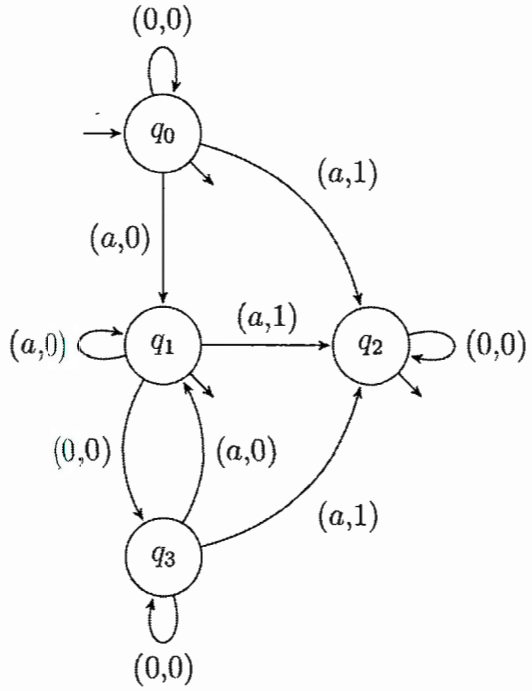

Figure 2.6 Automate de Büchi pour λ_X

0, mais que sa n^e lettre est 1. Si l'automate lit $(a_n, 1)$, alors on passe à l'état q_2 et (P, Q) sera accepté que si P et Q ne contiennent pas d'autre lettre non-nulle. Dans ce cas, on passe une infinité de fois par l'état q_2. Réciproquement, si P est un polynôme mais que Q ne contient aucun 1, alors le seul état visité une infinité de fois sera q_3, et ce n'est pas un état acceptant. □

Proposition 2.27. *Pour tout $k \in \mathbb{F}_p$, la relation $X_X(\cdot, \cdot, k)$ est ω-reconnaissable.*

Démonstration. Il suffit de construire un automate de Büchi qui reconnait l'ensemble $\{(P, Q) \in (\mathbb{F}_p[\![X]\!])^2 \mid X_X(P, Q, k)\}$, pour chaque $k \in \mathbb{F}_p$. Rappelons que $X_X(P, Q, k)$ est valide si et seulement si P est une puissance de X avec coefficient k dans Q. Pour un k donné, nous proposons l'automate $\mathcal{A}_k = (Q, \mathbb{F}_p^2, \{q_0\}, \{q_1\}, \delta)$, où

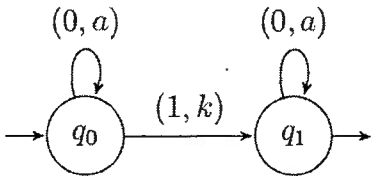

Figure 2.7 Automate de Büchi pour $X_X(\cdot, \cdot, k)$

- $Q = \{q_0, q_1\}$,
- $\delta = \{(q_i, (0, a), q_i) \mid a \in \mathbb{F}_p \text{ et } q_i \in Q\} \cup \{(q_0, (1, k), q_1)\}$.

Considérons (P, Q) une entrée dans l'automate $\mathcal{A}_k$. Comme toutes les transitions sont étiquetées par $(0, a)$, sauf une transition qui a l'étiquette $(1, k)$, il est clair que P doit nécessairement être une puissance de X pour que (P, Q) soit accepté par l'automate. La transition étiquetée $(1, k)$ nous indique que l'automate a lu l'unique coefficient non-nul dans P, et a en même temps lu le coefficient k dans Q. Ceci signifie clairement que la série formelle Q a le coefficient k devant la puissance P. □

[Cette page a été laissée intentionnellement blanche]

CHAPITRE III

RECONNAISSABILITÉ ET DÉFINISSABILITÉ

C'est dans ce chapitre que nous abordons le résultat principal de ce mémoire. Le théorème est analogue au Théorème 14 dans (Rigo et Waxweiler, 2011). Nous commençons par regarder de plus près ce théorème, et de là nous pouvons arriver plus facilement à notre propre théorème.

Nous démontrons l'équivalence entre être ω-rationnel et être définissable dans la structure

$$(\mathbb{F}_p[[X]], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot,\cdot,k)\}_{k\in\mathbb{F}_p}).$$

Rappelons que la relation binaire $\prec$ est définie sur $\mathbb{F}_p[X]$ et ordonne les polynômes par leur degré. L'opération unaire λ_X prend un élément et renvoie la plus grande puissance de X qui apparait avec un coefficient non-nul s'il s'agit d'un polynôme non-nul, et renvoie 0 sinon. Et finalement, la relation $X_X(u,v,k)$ est valide si et seulement si u est une puissance de X, et k est le coefficient de u dans v.

3.1 Reconnaissabilité et définissabilité dans les polynômes

Cette section sert non seulement à récapituler et comprendre les résultats importants de (Rigo et Waxweiler, 2011), mais aussi à les adapter à nos besoins. L'énoncé qui nous intéresse est le suivant.

Théorème 3.1. *Soit $P \in \mathbb{F}[X]$ un polynôme non constant et un entier $d \geq 2$. Alors un sous-ensemble $T \subseteq (\mathbb{F}[X])^d$ est P-reconnaissable si et seulement s'il est P-définissable.*

Par ensembles P-reconnaissables, on entend les ensembles de polynômes écrits en base P reconnaissables par automate fini. Par ensembles P-définissables, on entend les ensembles de polynômes définissables dans

$$(\mathbb{F}[X], +, \prec, \{\cdot C \mid C \in \mathbb{F}[X]\}, V_P),$$

où $\prec$ est une relation qui compare le degré de deux polynômes, et où $V_P(A)$ donne la plus grande puissance de P qui divise A si A est un polynôme non-nul, et $V_P(0) = 1$.

On note que les automates dans (Rigo et Waxweiler, 2011) lisent les polynômes à partir du coefficient du terme de plus haut degré. C'est-à-dire, pour un polynôme $A = a_n X^n + \cdots + a_1 X + a_0$, la représentation en base X, par exemple, sera $a_n \cdots a_1 a_0$ et ce sera le mot lu par l'automate. Malheureusement, puisque nous travaillons avec les séries formelles, il nous est impossible de représenter les séries à partir du coefficient du terme de plus haut degré (car une série formelle n'a pas de tel terme.) Nous devons alors vérifier que le théorème ci-dessus est encore valide en utilisant les représentations commençant par le coefficient du terme de plus petit degré. C'est-à-dire que pour un polynôme $A = a_n X^n + \cdots + a_1 X + a_0$, sa représentation en base X, notée $rep_X(A)$, sera $a_0 a_1 \cdots a_n$. De plus, ajoutons que la structure dans laquelle nous travaillerons sera

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k)\}_{k \in \mathbb{F}_p}).$$

Il est d'ailleurs intéressant de noter que l'ensemble des puissances de X, que nous notons P_X, est définissable dans ce langage. En effet :

$$u \in P_X \longleftrightarrow \lambda_X(u) = u.$$

La formule suivante suit la même idée que celle trouvée dans (Rigo et Waxweiler, 2011), qui découle d'une d'une construction trouvée dans le théorème 2.2 de (Villemaire, 1992). Étant donné un langage reconnu par un automate, nous voulons coder une formule décrivant le comportement de l'automate en utilisant notre représentation.

On suppose qu'une partie $\mathcal{T} \subseteq (\mathbb{F}_p[X])^d$ est P-reconnaissable, et nous posons $\mathcal{A} = (Q, (\mathbb{F}_p)^d, q_0, F, \delta)$ l'automate le reconnaissant, avec un nombre d'états l. Les états de $\mathcal{A}$ sont codés par les l-uplets $(1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)$ dans $(\mathbb{F}_p)^l$. Pour tout $q \in Q$, on note q_j la j-ième composante dans le codage de q. Pour représenter un m-uplet non-nul $(R_1, \ldots, R_m) \in (\mathbb{F}_p[X])^m$, $m \geq 1$, on écrit

$$((R_1(0), \ldots, R_m(0)), \ldots, (R_1(k), \ldots, R_m(k))) \in ((\mathbb{F}_p)^m)^*$$

pour $rep_X(R_1, \ldots, R_m)$, où $(R_1(k), \ldots, R_m(k))$ est non-nul et k dépend des R_i.

Exemple 3.2. Soient $R_1 = X + 2X^2$, $R_2 = 2 + 2X^2$ et $R_3 = 4X^4$ des polynômes de $\mathbb{F}_5[X]$. Alors le triplet (R_1, R_2, R_3) est représenté par

$$rep_X(R_1, R_2, R_3) = (0, 2, 0)(1, 0, 0)(2, 2, 0)(0, 0, 0)(0, 0, 4).$$

Chaque lettre de $rep_X(R_1, R_2, R_3)$ est un triplet car on représente ici un triplet de polynômes. Les éléments des triplets sont les coefficients de R_1, R_2 et R_3. Par exemple, la troisième lettre de $rep_X(R_1, R_2, R_3)$ est $(2, 2, 0)$ car les coefficients de X^2 dans R_1, R_2 et R_3 sont respectivement 2, 2 et 0.

L'idée de la formule est d'introduire un l-uplet $(B_1, \ldots, B_l)$ de polynômes qui code le comportement de l'automate $\mathcal{A}$ quand $\mathcal{A}$ lit le mot $rep_X(A_1, \ldots, A_d)$. Un

d-uplet $(A_1, \ldots, A_d)$ de polynômes est dans $\mathcal{T}$ si et seulement s'il existe un l-uplet $(B_1, \ldots, B_l)$ de polynômes et un entier k tel que

1. $k \leq \max\{deg(A_1), \ldots, deg(A_d)\} < k+1$,
2. $(B_1(0), \ldots, B_l(0))$ est le code pour l'état q_0,
3. pour tout $j \in \{0, \ldots, k\}$, si $(B_1(j), \ldots, B_l(j))$ est le code pour un état q, alors $(B_1(j+1), \ldots, B_l(j+1))$ est le code pour l'état $q' = \delta(q, (A_1(j), \ldots, A_d(j)))$, et
4. $(B_1(k), \ldots, B_l(k))$ est le code pour un état final.

Avec cette information, on peut voir que la formule suivante décrit les points (1)-(4) et qu'elle est satisfaite si et seulement si $(A_1, \ldots, A_d) \in \mathcal{T}$:

$$(\exists y)(\exists B_1)\cdots(\exists B_l)$$

$$\Bigg[P_X(y) \wedge \bigvee_{i=1}^{d}(y \preceq \lambda_X(A_i)) \wedge \bigwedge_{i=1}^{d}(\lambda_X(A_i) \prec y \cdot X) \wedge \bigwedge_{i=1}^{l}(X_X(1, B_i, q_{0(i)})) \wedge$$

$$(\forall z)(P_X(z) \wedge (z \preceq y) \Longrightarrow$$

$$\bigwedge_{\substack{((c_1,\ldots,c_d),q,q') \\ \in \mathbb{F}_p^d \times Q^2; \\ \delta(q,(c_1,\ldots,c_d)) = q'}} \left(\bigwedge_{i=1}^{d} X_X(z, A_i, c_i) \wedge \bigwedge_{j=1}^{l} X_X(z, B_j, q_j) \Rightarrow \bigwedge_{j=1}^{l} X_X(z \cdot X, B_j, q'_{(j)})\right.$$

$$\left.\wedge \bigvee_{q \in F} \bigwedge_{j=1}^{l} X_X(y, B_j, q_{(j)})\right)\Bigg].$$

Avec cette dernière formule, il est possible de définir des langages rationnels dans notre logique de premier ordre, et donc de vérifier si un mot est dans un langage donné. Étant donné la taille de cette formule, nous la remplaçons par la notation φ_A.

Exemple 3.3. Pour illustrer comment fonctionne cette formule, considérons le triplet (R_1, R_2, R_3) de l'exemple 3.2. Considérons à présent un langage $\mathcal{T}$ reconnu

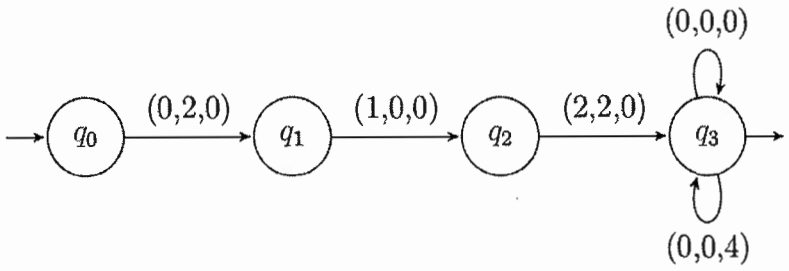

Figure 3.1 Automate de l'exemple 3.3

par un automate à 4 états. Ces 4 états sont q_0, q_1, q_2 et q_3 et sont respectivement codés par les quadruplets $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ et $(0,0,0,1)$. Supposons que q_0 est l'état initial, que q_3 est le seul état final et que les seules transitions possibles sont celles qui vont d'un état q_i à q_{i+1}, ou bien de q_3 à q_3. Un tel automate est représenté à la figure 3.1. Le triplet (R_1, R_2, R_3) fait partie du langage $\mathcal{T}$ si et seulement si il existe un quadruplet de polynômes (B_1, B_2, B_3, B_4) tel que les quatre conditions énumérées ci-haut sont respectées.

Premièrement, $\max\{R_1, R_2, R_3\} = 4$, donc pour respecter l'inégalité du point 1, on pose $k = 4$. Deuxièmement, $(B_1(0), B_2(0), B_3(0), B_4(0))$ doit être le code pour l'état initial. Dans cet exemple, l'état initial est codé par $(1,0,0,0)$, donc B_1 doit avoir le terme constant 1 et B_2, B_3 et B_4 doivent avoir un terme constant nul. Troisièmement, $(B_1(j+1), B_2(j+1), B_3(j+1), B_4(j+1))$ doit coder un état accessible par $(B_1(j), B_2(j), B_3(j), B_4(j))$ en lisant la lettre de $rep_X(R_1, R_2, R_3)$. Par exemple, $(B_1(1), B_2(1), B_3(1), B_4(1))$ doit être $(0,1,0,0)$ suivant les transitions de notre automate, et on doit avoir la transition $(q_0, (0,2,0), q_1)$, car $(0,2,0)$ est la première lettre du mot lu. Finalement, comme nous avons déjà posé $k = 4$, on doit avoir $(B_1(4), B_2(4), B_3(4), B_4(4)) = (0,0,0,1)$ puisqu'il s'agit de notre unique état final.

La suite d'états visités est codée par

$$(1,0,0,0)(0,1,0,0)(0,0,1,0)(0,0,0,1)(0,0,0,1)$$

et les quatre polynômes qui décrivent le comportement de l'automate sont $B_1 = 1$, $B_2 = X$, $B_3 = X^2$, $B_4 = X^3 + X^4$, et ces polynômes respectent les quatre conditions nécessaires pour coder le comportement de l'automate.

Notation 3.4. Nous écrivons $\varphi_{\mathcal{A}}(w)$ si w est dans le langage reconnu par $\mathcal{A}$.

3.2 Reconnaissabilité et définissabilité dans les séries formelles

Dans cette section, nous abordons finalement la preuve de notre théorème principal.

Théorème 3.5. *Soit L un langage de $(\mathbb{F}_p^d)^\omega$. Alors L est ω-reconnaissable si et seulement s'il est définissable dans la structure*

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot,\cdot,k)\}_{k\in\mathbb{F}_p}).$$

Démonstration. Montrons d'abord qu'un langage qui est ω-reconnaissable est définissable. Soit $\mathcal{A} = (Q, \mathbb{F}_p^d, q_d, F, \delta)$ un automate de Muller, où $\mathbb{F}_p^d$ est l'alphabet de l'automate, Q est l'ensemble de ses états, q_d est l'état initial, $F \subseteq \mathcal{P}(Q)$ est l'ensemble des sous-ensembles acceptants de l'automate et δ est la fonction de transition. Rappelons qu'un ω-mot sera accepté par $\mathcal{A}$ si et seulement si l'ensemble des états visités une infinité de fois appartient à F. Rappelons aussi que la classe des automates de Büchi et celle des automates de Muller sont équivalentes, ce qui nous permet d'utiliser les automates de Muller pour cette partie de la preuve.

On cherche alors à définir l'ensemble $L(\mathcal{A}) \subseteq (\mathbb{F}_p[\![X]\!])^d$, l'$\omega$-langage accepté par l'automate de Muller $\mathcal{A}$. Pour cela, nous allons définir la formule suivante pour

un automate $\mathcal{A}$ déterministe :

$$\phi_{(\mathcal{A},q)}(u,v) \leftrightarrow \varphi_{\mathcal{A}_q}(u) \wedge Pre(u,v),$$

où l'automate $\mathcal{A}_q$ est simplement l'automate $\mathcal{A}$ dans lequel nous avons remplacé l'ensemble des états d'acceptation par $\{q\}$. Rappelons que les formules $\varphi_{\mathcal{A}}$ et Pre ont été définies respectivement à la notation 3.4 et à la définition 1.26

Ainsi, pour un automate de Muller (déterministe) $\mathcal{A}$ et un de ses états q, on définit $\phi_{(\mathcal{A},q)}(u,v)$ par « u est un préfixe de v, et le chemin étiqueté par u dans l'automate $\mathcal{A}$ termine à l'état q. »

Soit w une série formelle, nous définissons $w \in L(\mathcal{A})$ par

$$\bigvee_{S \in F} \bigwedge_{\substack{q,q' \in Q \\ q \in S \\ q' \notin S}} [\forall y \exists u (P_X(y) \wedge (u \in \mathbb{F}_p[X]) \wedge (u \succeq y) \wedge \phi_{(\mathcal{A},q)}(u,w)) \wedge \exists y' \nexists v (P_X(y') \wedge (v \in \mathbb{F}_p[X]) \wedge (v \succeq y') \wedge \phi_{(\mathcal{A},q')}(v,w))]$$

La formule dit que, pour un $S \in F$, et pour tout couple d'états (q,q') tel que $q \in S$ et $q' \notin S$, les conditions suivantes sont vérifiées :

1) pour toute puissance de X, il existe un préfixe de w de degré plus grand qui finit son chemin à l'état q,
2) il existe une puissance de X telle qu'aucun préfixe de w de degré plus grand ne finit son chemin à l'état q'.

La condition (1) assure que w visite l'état q une infinité de fois, puisque nous pouvons trouver des préfixes de w arbitrairement grands qui finissent leur chemin dans l'automate $\mathcal{A}$ à l'état q.

La condition (2) assure que w visite l'état q' seulement un nombre fini de fois, puisqu'il existe une puissance de X telle que tout préfixe de w de degré plus grand

ne finira pas à l'état q'. Plus simplement, cela signifie que w cessera de visiter l'état q' après cette puissance de X.

Comme nous vérifions la condition (1) pour tout état se trouvant dans S, et la condition (2) pour tous les autres états, et ce pour un S pris dans l'ensemble des sous-ensembles acceptants F, nous définissons exactement l'acceptation d'un automate de Muller.

Il nous reste alors à montrer que tout ensemble définissable dans la logique de premier ordre de la structure

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot,\cdot,k)\}_{k\in\mathbb{F}_p})$$

est ω-reconnaissable.

Nous avons déjà vérifié au chapitre II que chaque ensemble défini par une formule atomique était ω-reconnaissable. Autrement dit, il existe des automates de Büchi qui reconnaissent les ensembles suivants :

- $\{w \in \mathbb{F}_p[\![X]\!] \mid w \in \mathbb{F}_p[X]\}$,
- $\{(u,v,w) \in (\mathbb{F}_p[\![X]\!])^3 \mid u+v=w\}$,
- $\{(u,v) \in (\mathbb{F}_p[\![X]\!])^2 \mid u \prec v\}$,
- $\{(u,v) \in (\mathbb{F}_p[\![X]\!])^2 \mid u \cdot X = v\}$,
- $\{(u,v) \in (\mathbb{F}_p[\![X]\!])^2 \mid \lambda_X(u) = v\}$,
- $\{(u,v) \in (\mathbb{F}_p[\![X]\!])^2 \mid X_X(u,v,k)\}$ pour tout $k \in \mathbb{F}_p$.

Nous pouvons alors procéder par récurrence sur la complexité d'une formule $\varphi(A_1,\ldots,A_d)$ contenant d variables libres. Nous pouvons supposer que $\varphi(A_1,\ldots,A_d)$ ne contient que les connecteurs logiques $\wedge$, $\vee$ et $\neg$, et le quantificateur existentiel $\exists$.

Si $\varphi(A_1,\ldots,A_d)$ est une formule atomique, alors nous savons qu'il existe un automate de Büchi reconnaissant l'ensemble qu'elle définit.

Si $\varphi(A_1, \ldots, A_d)$ est une formule de la forme $\neg\psi(A_1, \ldots, A_d)$, alors l'automate qui reconnait l'ensemble défini par $\varphi(A_1, \ldots, A_d)$ est l'automate qui reconnait le complément du langage défini par ψ. Comme les langages ω-reconnaissables sont fermés par le complément, nous savons que cet automate existe.

Si $\varphi(A_1, \ldots, A_d)$ est une formule de la forme $\psi_1(A_1, \ldots, A_d) \wedge \psi_2(A_1, \ldots, A_d)$, alors l'automate reconnaissant l'ensemble défini par $\varphi(A_1, \ldots, A_d)$ est celui qui reconnait l'intersection des langages définis par ψ_1 et ψ_2. Encore une fois, l'existence de cet automate provient du fait que les langages ω-reconnaissables sont fermés par l'intersection.

Si $\varphi(A_1, \ldots, A_d)$ est une formule de la forme $\psi_1(A_1, \ldots, A_d) \vee \psi_2(A_1, \ldots, A_d)$, alors l'automate reconnaissant l'ensemble défini par $\varphi(A_1, \ldots, A_d)$ est celui qui reconnait l'union des langages définis par ψ_1 et ψ_2. On sait que cet automate existe car les langages ω-reconnaissables sont fermés par l'union.

Finalement, si $\varphi(A_1, \ldots, A_d)$ est une formule de la forme $(\exists A_0)\psi(A_0, A_1, \ldots, A_d)$, alors l'automate qui reconnait l'ensemble défini par $\varphi(A_1, \ldots, A_d)$ est l'automate qui reconnait le projection sur $(A_1, \ldots, A_d)$ de l'ensemble défini par la formule $\psi(A_0, A_1, \ldots, A_d)$. Encore une fois, un tel automate existe grâce à la fermeture des langages ω-reconnaissables par la projection. □

La démonstration du théorème 3.5 assure que le théorème suivant de Hodgson s'applique, et on obtient que la structure

$$(\mathbb{F}_p[\![X]\!], \mathbb{F}_p[X], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k)\}_{k \in \mathbb{F}_p})$$

est décidable, c'est-à-dire qu'il existe une procédure pour déterminer effectivement si un énoncé quelconque de la logique de premier ordre de la structure est vrai ou non dans cette structure.

Théorème 3.6 ((Hodgson, 1983), théorème 1.3). *Soit $\mathcal{D}$ une structure relationnelle à laquelle nous associons le langage de premier ordre $L_{\mathcal{D}}$. S'il existe une procédure effective permettant d'associer à toute formule $\phi(x_1, \ldots, x_n)$ de $L_{\mathcal{D}}$ un ω-automate acceptant l'ensemble défini par cette formule, alors la théorie $Th(\mathcal{D})$ est décidable.*

3.3 Autres approches

Cette section a pour but de mettre en évidence certains problèmes que nous avons rencontrés. Cette section montre qu'il y a des résultats négatifs qui expliquent pourquoi certaines approches n'ont pas été utilisées.

3.3.1 Approche par la relation $L = \bigcup_{i=1}^{n} X_i Y_i^{\omega}$

Rappelons le théorème 2.10. Un langage L est ω-rationnel si et seulement s'il peut être écrit sous la forme

$$L = \bigcup_{i=1}^{n} X_i Y_i^{\omega},$$

où tout X_i et tout Y_i est un langage rationnel.

Au départ, nous voulions définir dans notre structure un langage L écrit sous cette forme. Malheureusement, nous n'avons pas réussi à trouver une méthode qui permet de définir $w \in XY^{\omega}$, où X et Y sont des langages rationnels. Nous savons déjà que $u \in X$ est définissable par la formule $\varphi_{\mathcal{A}_X}$, où $\mathcal{A}_X$ est l'automate reconnaissant X, puisque le langage X est rationnel. À partir de $\mathcal{A}_Y$, l'automate reconnaissant le langage rationnel Y, nous avons vu que nous pouvons construire un automate reconnaissant Y^*, disons $\mathcal{B}$. Alors, pour définir Y^{ω}, il suffit d'utiliser la propriété que $w \in Y^{\omega}$ si et seulement si w possède une infinité de préfixes dans

Y^*. Cela peut être défini pas la formule suivante :

$$w \in Y^\omega \longleftrightarrow (\forall z)(\exists v)[P_X(z) \wedge \varphi_{\mathcal{B}}(v) \wedge (z \preceq \lambda_X(v)) \wedge Pre(v, w)].$$

Cette formule dit que $w \in Y^\omega$ si et seulement si pour tout z une puissance de X, il existe un préfixe v de w qui est de degré égal ou plus grand que le degré de z, et qui est reconnu par l'automate $\mathcal{B}$. Ainsi v sera un mot de Y^*. En d'autres mots, $w \in Y^\omega$ si et seulement si w possède une infinité de préfixes dans Y^*.

Cependant, même si nous avons une formule définissant un langage rationnel X et un ω-langage Y^ω, nous n'avons pas réussi à définir l'ω-langage XY^ω. On peut voir que $w \in XY^\omega$ si et seulement si il existe $u \in X$ et $v \in Y^\omega$ tels que $w = u + \lambda_X(u) \cdot v$. Or nous ne pouvons pas multiplier une série par une puissance de X quelconque dans notre structure : le graphe de la multiplication définie sur $P_X \times \mathbb{F}_p[\![X]\!]$ n'est pas ω-reconnaissable, comme l'indique la proposition 2.19.

Cependant, lors d'une correspondance avec Luc Bélair, Françoise Point aurait récemment réussi à définir la concaténation à gauche par un langage rationnel dans une structure semblable à la nôtre. Dans sa lettre, elle définit $u \in XY^\omega$ en contournant le problème soulevé par la proposition 2.19.

3.3.2 Approche par les automates de Büchi déterministes

Nous nous sommes aussi demandé si le résultat ne pouvait pas être au moins partiellement vrai pour les automates de Büchi déterministes. Après tout, un ω-langage est reconnaissable par automate de Büchi déterministe si et seulement si il est la limite d'un langage reconnaissable par automate fini ce qui permet le théorème suivant.

Théorème 3.7. *Si un ω-langage de $(\mathbb{F}_p)^\omega$ est reconnaissable par automate de*

Büchi déterministe, alors il est définissable dans la structure

$$(\mathbb{F}_p[\![X]\!], +, 0, \prec, \cdot X, \lambda_X, \{X_X(\cdot, \cdot, k)\}_{k \in \mathbb{F}_p}).$$

Démonstration. Soit L un ω-langage reconnu par un automate de Büchi déterministe $\mathcal{A} = (Q, \mathbb{F}_p, d, F, \delta)$. Ainsi, on a $L = L^\omega(\mathcal{A})$. Soit $w \in L$, et $\mathrm{Inf}(w)$ l'ensemble des états visités une infinité de fois par un chemin acceptant de w dans $\mathcal{A}$. Comme w est accepté, nous avons $\mathrm{Inf}(w) \cap F \neq \emptyset$, par la définition d'acceptation des automates de Büchi.

Considérons l'automate fini $\mathcal{A}' = (Q, \mathbb{F}_p, d, F, \delta)$. Les automates $\mathcal{A}$ et $\mathcal{A}'$ ont exactement les mêmes états, états initiaux, états acceptants et les mêmes transitions. La seule différence entre les deux est la méthode d'acceptation : l'un est un automate de Büchi et l'autre est simplement un automate fini. On peut facilement remarquer qu'un mot u parcourra le même chemin dans $\mathcal{A}'$ que dans $\mathcal{A}$. Ainsi, on peut remarquer que $w \in L^\omega(\mathcal{A})$ si et seulement si w possède une infinité de préfixes $u_i \in L(\mathcal{A}')$.

Comme l'automate fini $\mathcal{A}'$ est déterministe, on peut définir le langage qu'il reconnait par une formule $\varphi_{\mathcal{A}'}$. La formule suivante définit alors $w \in L^\omega(\mathcal{A})$:

$$(\forall z)(\exists u)(P_X(z) \wedge \mathbb{F}_p[X](u) \wedge z \preceq u \wedge Pre(u, w) \wedge \varphi_{\mathcal{A}'}(u)).$$

Pour tout z qui est une puissance de X, il existe un polynôme u tel que u est de degré plus grand que le degré de z, tel que u est un préfixe de w et tel que u est accepté par l'automate fini $\mathcal{A}'$ construit à partir de l'automate de Büchi déterministe $\mathcal{A}$. Cela signifie que w est un mot accepté par l'automate de Büchi déterministe $\mathcal{A}$. □

Cependant, bien que tout langage reconnu par automate de Büchi déterministe

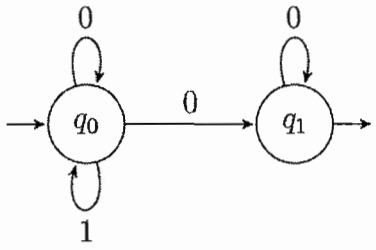

Figure 3.2 Automate de Büchi pour $\mathbb{F}_2[X]$

peut être défini dans notre structure, nous ne pouvons reconnaitre toute formule définissable.

En effet, bien que nous puissions définir l'ensemble des polynômes dans notre structure, aucun automate de Büchi déterministe ne peut les reconnaitre. Prenons l'automate de Büchi reconnaissant $\mathbb{F}_2[X]$. Cet automate reconnait l'ensemble des polynômes à coefficient dans $\mathbb{F}_2$, or il s'agit justement de l'automate de la figure 2.1. Cet automate n'est pas équivalent à un automate de Büchi déterministe, tel que vu dans la preuve du théorème 2.9.

De plus, la classe des automates de Büchi déterministes n'est pas fermée par la projection, ce qui cause un problème lorsque l'on veut construire un automate reconnaissant un ensemble défini par une formule contenant un quantificateur existentiel.

CONCLUSION

Inspiré par (Rigo et Waxweiler, 2011), ce mémoire avait pour but de trouver une structure logique de premier ordre sur les séries formelles dans laquelle toute formule définit un ensemble reconnaissable, et dans laquelle pour tout ensemble reconnaissable, il existe une formule le définissant.

La section 3.3 a décrit les différents problèmes que nous avons rencontrés pour démontrer le théorème 3.5. Terminer ce mémoire par cette section permet de considérer en partie quels problèmes il serait intéressant de résoudre. Notamment, nous n'avons pas réussi à définir directement la concaténation à gauche dans notre structure. Or, comme la concaténation à gauche est ω-reconnaissable, on sait qu'il doit être possible d'écrire une formule dans notre structure logique qui définit la concaténation. La missive de Françoise Point mentionnée précédemment semble le confirmer.

Nous avons aussi démontré que tout ensemble reconnaissable par automate de Büchi déterministe était définissable, et qu'il existe au moins un ensemble définissable qui ne peut être reconnu par automate de Büchi déterministe. Il n'y a donc pas d'équivalence entre reconnaissable (par Büchi déterministe) et définissable dans la structure donnée, mais il serait intéressant de trouver une structure pour laquelle l'équivalence tient. Cette structure devra être strictement moins expressive, tout comme les automates de Büchi déterministes par rapport aux non-déterministes.

Le sujet des ω-automates et de leurs liens avec la logique est beaucoup plus vaste que ce qui est couvert ici. Pour couvrir la matière plus profondément, je recommande les ouvrages cités dans les références.

RÉFÉRENCES

Rajeev Alur, Aldric Degorre, Oded Maler et Gera Weiss. (2009). On omega-languages defined by mean-payoff conditions. Dans Luca de Alfaro, *Foundations of Software Science and Computational Structures* (pp. 333-347), Lecture Notes in Computer Science, *5504*. Berlin. Springer.

Jean-Michel Autebert. (1994). *Théorie des langages et des automates.* Paris, Masson.

Véronique Bruyère. (1985). *Entiers et automates finis.* Mémoire de licence, Mons (Belgique), Université de l'état à Mons.

Véronique Bruyère, Georges Hansel, Christian Michaux et Roger Villemaire. (1994). Logic and p-recognizable sets, *Bulletin Belgian Mathematical Society. 1*(2), pp.191-238.

Richard Büchi. (1960a) Weak second-order arithmetic and finite automata. Dans S. Mac Lane et D. Siefkes, *The Collected Works of J. Richard Büchi* (pp. 398-424). New York. Springer-Verlag.

Richard Büchi. (1960b). On a Decision Method in Restricted Second Order Arithmetic. Dans S. Mac Lane et D. Siefkes, *The Collected Works of J. Richard Büchi* (pp. 425-435). New York. Springer-Verlag.

René Cori et Daniel Lascar. (2003). *Logique mathématique*, tome 1. Paris, Dunod.

Bernard R. Hodgson. (1983). Décidabilité par automate fini. *Annales des Sciences Mathématiques du Québec.* 7(1), pp. 39-57.

Michel Rigo et Laurent Waxweiler. (2011). Logical characterization of recognizable sets of polynomials over a finite field. *International Journal of Foundations of Computer Science. 22*(7), pp. 1549-1563. 10.1142/S0129054111008878.

Wolfgang Thomas. (1990). Automata on infinite objects. Dans J. Van Leeuwen, *Formal models and semantics* (pp. 133-191). Université de Kiel, Allemagne. Elsevier Science Publishers.

Roger Villemaire. (1992). The theory of $\langle N, +, V_k, V_l \rangle$ is undecidable. *Theoretical Computer Science. 106*(2), pp. 337-349.