AIX-MARSEILLE UNIVERSITÉ
E.D. 184
U.F.R. SCIENCES
Laboratoire d'informatique fondamentale de Marseille, CNRS UMR 7279

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : Informatique

El Makki VOUNDY

Langages ε-sûrs et caractérisations des langages d'ordres supérieurs

Soutenue le 15/11/2017 devant le jury composé de :

Sylvain SALVATI	Cristal, Université de Lille 1	Rapporteur
Olivier SERRE	IRIF, Université Paris Diderot	Rapporteur
Irène GUESSARIAN	IRIF, Université Paris Diderot	Examinateur
Jean-Eric PIN	IRIF, Université Paris Diderot	Examinateur
Séverine FRATANI	LIF, Aix-Marseille Université	Directeur de thèse
Jean-Marc TALBOT	LIF, Aix-Marseille Université	Directeur de thèse

Résumé

Une ligne de recherche présente dans la littérature depuis les années soixante est celle des *théorèmes de représentation*. Son résultat fondateur est le théorème de Chomsky–Schützenberger qui stipule qu'un langage est algébrique si et seulement si il est l'image par homomorphisme de l'intersection entre un langage régulier et le langage de Dyck. Ce résultat a connu depuis diverses généralisations à différentes familles de langages.

Dans cette thèse, nous proposons plusieurs généralisations de ce résultat aux langages d'ordres supérieurs. En particulier, nous introduisons une notion de langages de Dycks d'ordres supérieurs, nous définissons et étudions des classes de transductions que nous qualifions d'ε-sûres et nous montrons qu'un langage appartient à un niveau $k + l$ de la hiérarchie des ordres supérieurs si et seulement si il est l'image d'un langage de Dyck de niveau k par une transductions ε-sûre de niveau l. Ces résultats nous permettent aussi d'obtenir d'autres types de caractérisations tels les caractérisations logiques.

Abstract

Amongst the classical results of the language theory, one can cite the known characterization of algebraic languages proved by Chomsky and Schützenberger and which states that a language is algebraic if and only if it is the homomorphic image of a regular set intersected with the Dyck language. This result has opened a new line of research and defined a new type of characterizations known as *representation theorems*.

In this thesis, we prove various representation theorems for the higher order languages hierarchy. In particular, we introduce a notion of higher order Dyck languages and a hierarchy of classes of transductions that we call ε-stable (or ε-reducible) transductions and we prove that a language belongs to some level $k + l$ of the higher order hierarchy if and only if it can be represented as the image of a level-k Dyck language by a level-l ε-stable transduction. These representations also allow us to approach other types of characterizations such as logical characterizations.

Remerciements

C'est en écrivant ces lignes que je comprends enfin la pertinence d'avoir une représentation finie d'un objet qui ne l'est pas. Si votre nom n'apparait pas explicitement dans cette liste, n'en soyez point vexé.

Je remercie en premier lieu mes encadrants, Séverine et Jean-Marc, qui m'ont accompagné pendant ces année avec beaucoup de patience et de bienveillance.

Je remercie Sylvain Salvati et Olivier Serre d'avoir accepté d'être rapporteur et pour leurs retours constructifs. En particulier Sylvain qui a sans le savoir, contribué aux développement de ces travaux et à l'intérêt que je porte à la théorie des langages par un simple envoi de références. Je remercie aussi Jean-Eric Pin et Irène Guessarian d'avoir accepté d'être examinateurs.

Je remercie les membres du LIF dont la bonne humeur fait de ce lieu de travail un lieu de vie. Une attention particulière aux doctorants et postdocs du labo avec lesquels j'ai partagé les joies et les peines de la thèse. Sans tous les citer : Antoine, Didier, Eloi, « Lady Christina », Elie, Mathieu, Flo, Olivier, Seb, les « Jerems », Manon, Tibs. Sans oublier Luc « the dude ».

Je remercie aussi les membres du « GT Jeux de société » pour ce rendez-vous de bonne humeur hebdomadaire.

Puis dans le fond, comment parler de vecteurs de bonne humeur sans mentionner les STARs du labo ? Martine, Sylvie R. et Sylvie R., Nadine, Manu dit « l'œil de Moscou », Kai mais aussi Sonia de l'ED et Gisèle du département ; puis notre cher directeur Liva.

Je remercie toute l'équipe MoVe pour son accueil. En particulier Pierre-Alain qui a des choses très intéressantes à apprendre aux jeunes doctorants lors des JSL :-) et Fred qui mérite remerciements et félicitations pour le fait d'être Fred.

Il y'a plus important encore que nos conditions de travail, de la réussite de notre thèse ou même de notre réussite sociale–si pouvait-on mettre un sens objectif à ces mots. Il s'agit de l'amour que l'on reçoit au quotidien. Je remercie tous ces gens qui me rendent heureux : ma famille et mes amis. Mes parents, mes nombreux frères et sœurs, le cartel des « Clairsolois », les Talon-Esmieu, Vinçoune et Nath, les « cabaverdiens », Tati, Marius. Il n'existe pas de bisou assez gros pour vous remercier alors je me contente de vous dédier cette thèse.

Hammer time !

Sommaire

II Caractérisations des langages d'ordres supérieurs

Introduction

Théorie des langages

Un langage formel est un ensemble de mots (suites de symboles) qui respectent une certaine propriété. Qu'ont-ils d'intéressant ? Répondons que la théorie des langages (l'étude des langages formels) est commune à diverses disciplines ; en particulier à la linguistique, aux mathématiques et à l'informatique. Les linguistes s'y intéressent pour l'étude et la formalisation des langues naturelles au moyen de systèmes de réécriture ou grammaires formelles. La théorie des langages s'applique aussi à diverses branches et problématiques des mathématiques et de l'informatique telles que la calculabilité, la modélisation de programmes, l'analyse syntaxique, la compilation et autres. L'ensemble d'instances positives d'un problème informatique (mathématique) peut être décrit comme un langage formel. N'oublions pas de mentionner que les automates finis de Kleene et les systèmes de Lindenmayer ont été introduits pour la modélisation de systèmes biologiques. Même la biologie n'est épargnée par présence de langages formels. Cette omniprésence de la théorie des langages la rend riche en problématiques tirées de diverses disciplines.

Les origines de la théorie des langages sont bien évidemment multiples. D'une part, des mathématiques, en particulier, de certain problèmes de combinatoire et d'algèbre. Dans cette ligne de recherche, nous pouvons citer Thue qui au début du 20ème, a étudié les motifs « évitables » et « inévitables » dans les mots infinis (voir [Ber92]). Avec Post [Pos36], ils ont introduit la notion de *systèmes de réécriture*. Une autre origine est évidemment celle de la théorie de la calculabilité où les travaux de Turing [Chu37] sont d'une grande importance.

Hiérarchie de Chomsky

La discipline qui a le plus motivé l'étude des langages formels est sans doute la linguistique. En particulier l'étude des grammaires par Chomsky [Cho56] a fortement contribué au développement de la théorie des langages. Ce dernier classifie les grammaires en quatre types : les grammaires de type 0, sans restriction, qui sont équivalentes aux machines de Turing, et décrivent les langages dits *récursivement énumérables*. Les grammaires de type 1, qui décrivent des langages dits *contextuels* ou *context-sensibles*. Les grammaires de type 2, qui décrivent des langages dits *hors-contexte* ou *algébriques*. Enfin, les grammaires de type 3 qui décrivent des langages *réguliers*. C'est ainsi que naquit la hiérarchie de Chomsky. Les différentes classes de cette hiérarchie sont depuis lors étudiées en informatique pour leurs diverses applications, mais aussi d'une motivation purement théorique.

Hiérarchie des ordres supérieurs (Hiérarchie OI)

En 1968, Aho [Aho67] introduit les grammaires indexées. Fischer [Fis68] introduit en même temps les grammaires à macros, équivalentes aux grammaires indexées d'Aho lorsqu'on consi-

dère le sens de dérivation « OI ». Les langages indexés prennent une certaine importance du fait d'être strictement plus expressifs que les langages algébriques tout en gardant les mêmes propriétés de clôture et de décidabilité que ces derniers. Un exemple typique de langage indexé non-algébrique est le langage de copies $\{ww \mid w \in A^*\}$. Les langages (grammaires) indexés permettent de décrire des dépendances syntaxiques plus fortes que les algébriques et Gazdar [Gaz88] propose leur usage à la linguistique.

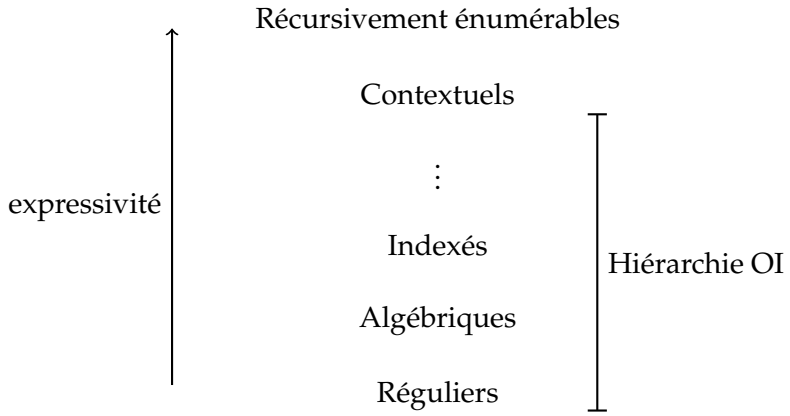


FIGURE 1 – La hiérarchie OI dans celle de Chomsky.

Maslov [Mas74] introduit par la suite une hiérarchie de grammaires généralisant celles d'Aho ainsi qu'une classe d'automates équivalente : les « multilevel stack automata ». Les langages générés par ces grammaires et reconnus par ces automates de différents niveaux forment une hiérarchie *stricte* et *infinie* incluse dans la classe des langages contextuels et dans laquelle en particulier, le niveau 0 est celui des langages réguliers, le niveau 1 est celui des langages algébriques et le niveau 2 est celui des langages indexés. Celle-ci est connue sous le nom de « hiérarchie des langages d'ordres supérieurs » ou « hiérarchie OI ». Le panorama global est celui de la figure 1. En particulier, le nom « hiérarchie OI » vient de son équivalence avec la généralisation des grammaires à macro OI de Fischer introduite par Damm [Dam82]. Tout au long de leur histoire, différents types de formalismes reconnaissant ces langages ont été introduits, parmi lesquels, les automates à *piles itérées* de Damm, Goerdt et Engelfriet [DG86, Eng83].

Un des intérêts portés à ces objets est qu'ils permettent de modéliser les schémas récursifs de programmes. Damn l'illustre en s'intéressant aux langages de programmation de type ALGOL. Ils sont aussi d'un grand intérêt théorique. Dans [FS06], par exemple, les auteurs s'intéressent aux suites d'entiers calculables par des automates de niveaux supérieurs ou encore dans [Car05], l'auteur s'intéresse à la notion de *régularité* des suites d'instructions d'automates de niveaux supérieurs. Dans cette thèse, nous présentons différentes caractérisations des langages de la hiérarchie OI.

Problèmes de Caractérisation

Une problématique courante de la théorie des langages est celle de la caractérisation de famille de langages : peut-on dire qu'une famille de langages correspond précisément à l'ensemble de langages décrits par un tel formalisme ? Autre que les problématiques de calculabilité/complexité qui motivent souvent ces questionnements– en particulier lorsqu'on s'intéresse

à caractériser une famille de langages par un type de « machines »– les caractérisations de familles de langages permettent d'avoir une vision différente d'un même objet, ou disons, une approche différente d'un même problème. Dans ce manuscrit, nous parlerons de ces différents types de caractérisations.

Théorèmes de représentation (caractérisations homomorphiques)

Au début des années soixante, Chomsky et Schützenberger [CS63] prouvent une caractérisation particulière des langages algébriques : ils montrent qu'un langage est algébrique si et seulement si il peut s'exprimer comme l'image par homomorphisme de l'intersection entre un langage régulier et le langage de Dyck. Le langage de Dyck est l'ensemble de mots bien *parenthésés* pour un certain nombre de paires de parenthèses. Ce résultat a été généralisé aux langages récursivement énumérables [HN81, HOY85] : un langage est récursivement énumérable si et seulement si il est l'image par homomorphisme de l'intersection entre un langage algébrique et le langage de Dyck. Ces résultats ont initié une ligne de recherche encore présente aujourd'hui ; on parle de *théorèmes de représentation* ou tantôt de *caractérisations homomorphiques*.

Un intérêt immédiat de ces caractérisations est qu'elles permettent de comparer des familles de langages. Par exemple, une conséquence du théorème de Chomsky–Schützenberger est que toute famille de langages qui contient le langage de Dyck et qui est fermée par intersection avec des langages réguliers et homomorphismes contient nécessairement toute la famille des langages algébriques (voir [Pig16] pour une application récente de ce principe à certaines restrictions de machines de Turing).

Ces deux résultats susmentionnés ont gagné en ampleur par les résultats respectifs de Nivat et Eilenberg. Nivat [Niv68] montre qu'une transduction (transformation de mots) est rationnelle si et seulement si elle peut s'exprimer comme la composition d'un homomorphisme inverse, d'une intersection avec un langage régulier et un homomorphisme. Eilenberg [Ber79], montre qu'une transduction est algébrique si et seulement si elle peut s'exprimer comme la composition d'un homomorphisme inverse, d'un intersection avec un langage algébrique et d'un homomorphisme. Ces deux types de transductions admettent de plus des modèles de calcul. La famille des langages algébriques est alors l'ensemble d'images du langage de Dyck par transductions rationnelles ; et celle des langages récursivement énumérables est l'ensemble d'images du langage de Dyck par transductions algébriques.

Représentation par homomorphismes inverses. Les théorèmes de représentation s'appliquent aussi à la complexité. En effet, dire qu'un problème X se réduit à un problème Y signifie qu'il existe une fonction calculable f telle que l'ensemble des instances positives de X peut s'exprimer comme l'image inverse par f de l'ensemble d'instance positives de Y ($pos(X) = f^{-1}(pos(Y))$). Un cas particulier des théorèmes de représentation est la représentation dite par homomorphismes inverses. Dans [Gre73], Greibach montre l'existence d'un langage connu comme « le langage algébrique le plus dur ». Il s'agit d'un langage algébrique pour lequel tout autre langage algébrique en est l'image inverse par un homomorphisme effectivement calculable depuis une grammaire algébrique (en forme normale de Greibach). Ce dernier est alors LOGCFL-complet. Ce résultat a connu plusieurs généralisations dont une des plus récente est celle d'Okhotin [Okh16] pour les langages générés par des grammaires « booléennes » et « conjonctives ».

Théorie des familles agréables. La présence de divers théorèmes de représentation et l'importance de plus en plus visibles des propriétés de clôture de familles de langages, et la présence

croissante de modèles de calculs dans la littérature ont contribué au développement de la *théorie des familles agréables de langages*. Dans [GG67, Gin75], Greibach et Ginsburg classifient des familles de langages par différents types en fonction de leurs propriétés de clôture. Ils introduisent en même temps la notion de familles principales : une famille de langages est dite principale vis-à-vis d'un ensemble d'opérations si elle peut se décrire comme la fermeture d'un langage par ces opérations. Il nait ainsi une nouvelle problématique, « que peut-on dire sur une famille de langages en ayant comme unique information ses propriétés de clôture, et éventuellement le fait qu'elle soit principale ? ». Ce questionnement est d'autant plus important que la plupart des familles de langages présentes dans la hiérarchie de Chomsky partagent les mêmes propriétés de clôture. Cette théorie rencontre un succès fulgurant et s'emploie même à des problèmes d'indécidabilité (voir [Gre68, ZKL17]).

Caractérisations logiques

Les années 60 ont connu la crise du logiciel (*software crisis*). C'est le nom qui a été donné au constat que l'informatique était peu rentable et fiable lors d'une conférence de *génie logiciel*. En réponse à ce constat, certaines problématiques du génie logiciel, telles que la *spécification* et la *vérification* de systèmes ont pris beaucoup d'importance . (Spécification :) Comment décrire de façon précise les comportements autorisés d'un système, et (vérification :) comment vérifier (automatiquement) qu'un comportement est conforme à la spécification du système ? Les modèles de calcul tels que les *automates* sont adéquats pour la tâche de vérification du fait qu'ils décrivent les différentes étapes d'une fonction de test. En ce qui concerne la tâche de spécification, ils sont disons, beaucoup moins pratiques à utiliser ; et certainement beaucoup moins que la logique mathématique. La théorie des langages et la *complexité descriptive* partagent alors par cette motivation une longue tradition de caractérisations logiques de familles de langages. Le résultat fondateur en est le théorème de Büchi–Elgot–Trakhtenbrot [Bŏ0, Elg61, Tra61] qui stipule qu'un langage est régulier si et seulement si il est l'ensemble de mots modèles d'une formule du second ordre monadique. Ce résultat a été généralisé à plusieurs sous-classes des langages réguliers. En 1994, Lautemann, Schwentick et Thérien [LST94] proposent l'usage de fragments sémantiques de la logique du second ordre existentiel ($\exists$SO). Celle-ci consiste à restreindre les valuations d'une variable dans une formule logique à une certaine classe de relations. Caractériser une famille de langages en utilisant cette approche consiste alors à trouver la bonne classe de relations pour cette famille de langages. Ce type de caractérisation suit la même logique que les théorèmes de représentations et nous aurons l'occasion de voir que la classe de relations considérée dans une telle logique peut être identifiée au générateur d'une famille de langages.

Contributions et Plan

Les contributions de ces travaux sont réparties dans les parties I et II. Nous présentons chaque chapitre de contributions par un bref résumé.

Partie I

La motivation initiale des travaux de cette partie était l'obtention d'une caractérisation logique des langages indexés. Afin de généraliser la caractérisation de Lautemann, Schwentick et Thérien aux langages indexés, il nous est utile d'introduire une classe de transductions algébriques permettant de générer les langages indexés comme images du langage de Dyck. Ceci nous a conduit à développer la notion de langages et transductions ε-sûrs. Dans cette partie,

nous introduisons ces classes de langages et transductions, nous prouvons des théorèmes de représentation des langages indexés par ces classes et nous l'appliquons à l'obtention de caractérisations logiques des langages indexés.

Chapitre 3 : Langages algébriques ε-sûrs

Nous introduisons dans ce chapitre les classes de langages et transductions algébriques ε-sûrs. Nous disons qu'un langage algébrique est ε-sûr s'il peut être généré par une grammaire algébrique pour laquelle le langage associé à chaque variable se réduit au mot vide « ε » par la loi de réduction du groupe libre $a\bar{a} = \bar{a}a = \varepsilon$. Une transduction algébrique est dite ε-sûre si son domaine est ε-sûr. Bien que, comme nous le montrons, la famille de langages algébriques ε-sûrs est fermée par la plupart des opérations qui préservent la réduction à ε, celle-ci est strictement incluse dans la famille des langages algébriques qui se réduisent à ε. Nous montrons de plus qu'il est indécidable de savoir si un langage algébrique de cette dernière classe est ε-sûr. Nous présentons aussi des théorèmes de représentation pour les langages et transductions algébriques ε-sûrs.

Chaptire 4 : Théorèmes de représentation des langages indexés

Le théorème de Chomsky–Schützenberger a connu plusieurs généralisations pour diverses familles de langages, parmis lesquelles, celle de Weir [Wei88] pour langages *indexés linéaires*. Celui-ci définit un langage de Dyck « indexé linéaire » et montre qu'un langage est indexé linéaire si et seulement si il en est l'image par une transduction rationnelle. Ce même résultat fut généralisé par Kanazawa [Kan14b] et Sorokin [Sor14] aux frontières de forêts algébriques « simples ». Nous généralisons ces résultats aux langages indexés. Nous prouvons en plusieurs variantes les caractérisations des langages indexés comme images du langages de Dyck par transductions algébriques ε-sûres. Nous définissons de même un langage de Dyck « indexé » dont la famille des langages indexés en est la fermeture par transductions rationnelles.

Chapitre 5 : Caractérisations logiques des langages indexés

Nous étudions la définissabilité des langages indexés dans des fragments sémantiques du second ordre existentiel. Nous nous intéressons aussi aux liens entre certains fragments sémantiques du second ordre existentiel et cônes (croissants) principaux.

Partie II

Dans cette partie, nous prouvons des théorèmes de représentations pour les langages de la hiérarchie OI ainsi qu'une représentation par homomorphismes inverses pour une sous-classe de cette hiérarchie.

Chapitre 7 : Itérations du langage de Dyck

Nous introduisons une notion de « langages de Dyck d'ordres supérieurs ». Un langage de Dyck de niveau $k+1$ est un ensemble de mots de Dyck pour lesquels l'image par un homomorphisme ε-sûr donné appartient à un langage de Dyck de niveau k donné. Nous montrons que chaque niveau $k \geq 1$ de la hiérarchie OI est la fermeture par transductions rationnelles des langages de Dyck du même niveau. Une conséquence de ce fait est qu'il existe pour tout $k \geq 1$ une classe de transductions que nous appelons ε-sûres, telle que pour tout $l \geq 1$ la famille du

niveau $k + l$ des ordres supérieurs est exactement l'ensemble d'images de langage de Dyck de niveau l par des transductions ε-sûres de niveau k. Nous nous intéressons aussi de façon générale aux familles de langages générées par des formes quelconques d'itérations du langage de Dyck.

Chapitre 8 : Les langages les plus durs

Nous nous intéressons de façon générale aux cônes fidèles principaux. Un cône fidèle est une famille de langages fermée homomorphismes non-effaçants, intersection avec des langages réguliers et homomorphismes inverses. Nous montrons des conditions suffisantes à ce qu'un cône fidèle principal admette un théorème de représentation par homomorphismes inverses. De cette façon, ces conditions suffisent à pouvoir trouver pour tout cône fidèle principal $\mathcal{F}$, un langage L tel que le problème d'appartenance du mot de tout autre langage de $\mathcal{F}$ puisse se réduire à celui de L. Comme application de ce résultat, nous montrons que pour tout niveau k de la hiérarchie OI, la classe des langages reconnus par des automates de niveau k en temps « quasi-réel » satisfait ces conditions et admet un tel langage. Nous généralisons ainsi le « langage algébrique le plus dur » de Greibach à toute ces classes de langages.

Chapitre 1

Définitions de base

1.1 Mots et langages

Monoïde libre et Mots. Un *monoïde* est une structure algébrique $(M, \cdot_M, 1_M)$ où M est un ensemble, $\cdot_M$ est une opération binaire associative, (pour tous $m_1, m_2, m_3 \in M : (m_1 \cdot_M m_2) \cdot_M m_3 = m_1 \cdot_M (m_2 \cdot_M m_3) = m_1 \cdot_M m_2 \cdot_M m_3$) et 1_M est un élément neutre pour $\cdot_M$ (pour tout $m \in M : 1_M \cdot_M m = m \cdot_M 1_M = m$).

Monoïde libre. Un *alphabet* est un ensemble fini de symboles $A = \{a_1, \ldots, a_n\}$. Un *mot* sur un alphabet A est une suite, possiblement vide, de symboles de A. Le *mot vide* est noté ε et l'ensemble de mots (resp. non vides) sur l'alphabet A est noté A^* (resp. A^+).

Étant donné deux mots $u = a_1 \cdots a_n$ et $v = b_1 \cdots b_m$, la *concaténation* de u et v est le mot $uv = a_1 \cdots a_n b_1 \cdots b_m$ obtenu en juxtaposant les mots u et v. Le mot vide « ε » est l'élément neutre pour la concaténation. Pour tout mot u et pour tout entier n, le mot u^n se définit comme suit : $u^0 = \varepsilon$ et $u^{n+1} = uu^n$. L'ensemble A^* accompagné de la concaténation comme produit et de ε comme élément neutre est le monoïde *libre engendré par* A. Celui-ci est simplement noté A^*.

Notations. Poursuivons par quelques notations sur les mots. Étant donné un mot u, $|u|$ désigne la longueur de u et $|u|_a$ désigne le nombre d'occurrences du symbole a dans le mot u, $premier(u)$ désigne sa première lettre et $dernier(u)$ sa dernière. Pour tout autre mot v, le mot u est dit préfixe de v, et noté $u \preccurlyeq v$, s'il existe un mot w tel que $v = uw$. L'image miroir de $u = a_1 a_2 \cdots a_n$, est le mot $u^R = a_n \cdots a_1$.

Soit A un alphabet. Nous désignons par $\overline{A} = \{\bar{a} \mid a \in A\}$ une copie disjointe de cet alphabet et par $\widehat{A}$ l'union de A et $\overline{A}$. L'alphabet A est dit *positif* et $\overline{A}$ *négatif*. Nous adoptons les conventions suivantes : $\bar{\bar{a}} = a$ pour tout symbole $a \in \widehat{A}$; $\bar{\varepsilon} = \varepsilon$ et pour tout mot $u = a_1 \cdots a_n \in \widehat{A}^*$, $\overline{u} = \bar{a_n} \cdots \bar{a_1}$.

Langages et opérations Passons maintenant à la notion de *langages*. Un langage sur un alphabet A est un ensemble $L \subseteq A^*$ de mots sur cet alphabet. Les opérations d'union, intersection, différence sont les opérations ensemblistes habituelles. La concaténation de deux langages L_1 et L_2 est le langage $L_1 L_2 = \{uv \mid u \in L_1 \text{ et } v \in L_2\}$.

–Étoile de Kleene. Pour tout langage L, les puissances de L se définissent de façon similaire aux mots ; $L^0 = \{\varepsilon\}$ et $L^{n+1} = LL^n$ pour tout entier $n \in \mathbb{N}$. L'étoile de Kleene d'un langage L est l'union infinie de ses puissances : $L^* = \bigcup_{i \geq 0} L^i$. L'étoile de Kleene *stricte* est quant à elle

$L^+ = \bigcup_{i\geq 1} L^i$.

–*Préfixes.* La fermeture par préfixes d'un langage $L \subseteq A^*$ sera notée $L^{\preccurlyeq}$, et se définit comme le langage $L^{\preccurlyeq} = \{u \in A^* \mid \exists v \in L : u \preccurlyeq v\}$.

–*Quotients.* Pour tout langage $L \subseteq A^*$ et pour tout mot $u \in A^*$, le quotient gauche (resp. droit) de L par u est le langage

$$u^{-1} \cdot L = \{v \in A^* \mid uv \in L\} \quad (\text{resp. } L \cdot u^{-1} = \{v \in A^* \mid vu \in L\}).$$

1.2 Morphismes

Soient M et N deux monoïdes. Un morphisme de monoïde de M dans N est une application $f : M \longrightarrow N$ telle que $f(1_M) = 1_N$ et quels que soient $m_1, m_2 \in M : f(m_1 \cdot_M m_2) = f(m_1) \cdot_N f(m_2)$. L'image inverse d'un élément est l'ensemble de ces antécédents :

$$f^{-1}(y) = \{x \in M \mid f(x) = y\} \qquad \forall y \in N.$$

Comme il en est le cas pour toute fonction, étant donné un morphisme $f : M \longrightarrow N$, et des ensembles $X_1, X_2 \subseteq M$ et $Y_1, Y_2 \subseteq N$, nous avons les égalités suivantes :

$$f(X_1 \cup X_2) = f(X_1) \cup f(X_2), \qquad f^{-1}(Y_1 \cup Y_2) = f^{-1}(Y_1) \cup f^{-1}(Y_2),$$
$$f^{-1}(Y_1 \cap Y_2) = f^{-1}(Y_1) \cap f^{-1}(Y_2), \qquad f(X_1 \cap f^{-1}(Y_1)) = f(X_1) \cap Y_1.$$

De plus, si f est *injectif*, c.-à-d « $\forall m_1, m_2 \in M : f(m_1) = f(m_2) \Rightarrow m_1 = m_2$ », alors

$$f(X_1 \cap X_2) = f(X_1) \cap f(X_2).$$

1.2.1 Substitutions

Une substitution est un morphisme $\nu : A^* \longrightarrow 2^{B^*}$ qui à chaque symbole a associe un langage $\nu(a) \subseteq B^*$; satisfaisant $\nu(\varepsilon) = \{\varepsilon\}$, $\nu(uv) = \nu(u)\nu(v)$ pour tout $u, v \in A^*$; mais étendu aux langages comme suit :

$$\nu(L) = \bigcup_{u \in L} \nu(u) \qquad \forall L \subseteq A^*.$$

Exemple 1.1. *Si $L = \{ab, a\}$ et ν est tel que*

$$\nu(a) = \{a^n b^n \mid n \geq 0\}; \quad \nu(b) = \{bba\},$$

alors $\nu(L) = \{a^n b^n bba \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$.

Homomorphismes. Un *homomorphisme* est un morphisme entre deux structures algébriques similaires. Tout au long de ce manuscrit, cette appellation sera réservée aux homomorphismes entre monoïdes libres. Soient deux alphabets A et B ; et $f : A^* \longrightarrow B^*$ un homomorphisme. Ce dernier est dit

— *alphabétique* si $f(a) \in B \cup \{\varepsilon\}$ pour tout $a \in A$,
— *non-effaçant* si $f(a) \neq \varepsilon$ pour tout $a \in A$,
— *strictement alphabétique* si alphabétique et non-effaçant.

1.2.2 Projections

Soient A et B deux alphabets tels que $B \subseteq A$. La projection de A dans B est l'homomorphisme $\pi_{A,B} : A^* \longrightarrow B^*$ tel que $\forall a \in A$:

$$\pi_{A,B}(a) = \begin{cases} a \text{ si } a \in B \\ \varepsilon \text{ sinon} \end{cases}.$$

On notera généralement π_B au lieu de $\pi_{A,B}$ si cela ne soulève aucune ambiguïté.

Projections de composantes. Soit $A_1 \times A_2 \times \cdots \times A_k$ un produit de k alphabets et $i \in \{1, \ldots, k\}$ un entier. La projection de $A_1 \times A_2 \times \cdots \times A_k$ sur la composante i est l'homomorphisme $\pi_i : (A_1 \times A_2 \times \cdots A_k)^* \longrightarrow A_i^*$ tel que $\pi_i((a_1, a_2, \ldots, a_k)) = a_i$ pour tout $(a_1, a_2, \ldots, a_k) \in A_1 \times A_2 \times \cdots \times A_k$.

1.3 Ensembles rationnels

Dans cette partie, nous présentons brièvement les langages et relations (transductions) rationnelles.

Définition 1.2 (Partie rationnelle)**.** *Soit M un monoïde.* $\text{RAT}(M)$ *désigne le plus petit $\mathcal{F} \subseteq 2^M$ tel que :*

1. $\emptyset \in \mathcal{F}$ *et* $\forall m \in M : \{m\} \in \mathcal{F}$,
2. *pour tout $L_1, L_2 \subseteq M$:*
   (a) $L_1 \cup L_2 \in \mathcal{F}$,
   (b) $L_1 L_2 \in \mathcal{F}$,
   (c) *et* ${L_1}^* \in F$.

$\text{RAT}(M)$ *est appelée* partie rationnelle *du monoïde M.*

Définition 1.3. *Soit M un monoïde et $X \subseteq M$ un ensemble. L'ensemble X est dit* rationnel *si il appartient à la partie rationnelle de M.*

1.3.1 Langages Réguliers et automates finis

Définition 1.4 (Automates finis)**.** *Un automate fini est une structure $\mathcal{A} = (Q, A, \Delta, q_0, F)$ dans laquelle :*

— *Q est un ensemble fini. Les éléments $q \in Q$ sont appelés* états *;*
— *A est l'alphabet d'entrée ;*
— *$\Delta \subseteq Q \times A \times Q$ est une relation dite de* transitions *;*
— *$q_0 \in Q$ est l'état* initial *;*
— *$F \subseteq Q$ est un ensemble d'états dits* finaux *ou* acceptants.

Calculs. Un calcul est une suite $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \cdots \xrightarrow{a_n} q_{n+1}$ telle que pour tout $i \in \{1, \ldots, n\}$: $(q_i, a_i, q_{i+1}) \in \Delta$. On dira ici que le mot $a_1 a_2 \cdots a_n$ admet un calcul de q_1 vers q_n ; et on le notera $q_1 \overset{a_1 a_2 \cdots a_n}{\underset{*}{\Rightarrow}} q_{n+1}$. Un mot u est dit *accepté* ou *reconnu* par $\mathcal{A}$ s'il admet un calcul de q_0 vers un état acceptant $q \in F$. Le langage reconnu (accepté) par $\mathcal{A}$ est alors

$$\mathscr{L}(\mathcal{A}) = \{u \in A^* \mid \exists q \in F : q_0 \overset{u}{\underset{*}{\Rightarrow}} q\}.$$

$\text{REG}(A)$ désigne l'ensemble de langages sur l'alphabet A reconnaissables par des automates finis.

Théorème 1.1 (Théorème de Kleene [Kle56]). $\text{REG}(A) = \text{RAT}(A^*)$.

Théorème 1.2 (voir [Pin83]). *Un langage $L \subseteq A^*$ est régulier si et seulement si il existe un monoïde fini M, un ensemble $X \subseteq M$ et un morphisme $\mu : A^* \longrightarrow M$ tel que $L = \mu^{-1}(X)$.*

Définition 1.5 (Localité). *Un langage (régulier) $L \subseteq A^*$ est dit local si il existe des ensembles $D, F \subseteq A$ et $I \subseteq AA$ tels que*

$$L = (DA^* \cap A^*F) - A^*IA^* \text{ ou } L = \{\varepsilon\} \cup (DA^* \cap A^*F) - A^*IA^*.$$

Cette appellation vient du fait que pour vérifier qu'un mot appartient à L, il suffit de vérifier que sa première lettre appartient à D, sa dernière à F et qu'aucun facteur de taille 2 n'appartient à I. L'ensemble I est appelé *ensemble de transitions interdites*, et l'ensemble $A^2 - I$, *ensemble de transitions autorisées*. Tout langage local est bien évidemment régulier.

Exemple 1.6. *Le langage $\{ab\}^*$ est local car il est composé du mot vide et de mots commençants par le symbole a, se terminant par b et tels qu'aucun facteur de taille 2 n'appartient à $\{aa, bb\}$.*

1.3.2 Transductions Rationnelles et transducteurs finis

Soient A et B deux alphabets. Le monoïde $A^* \times B^*$ représente le produit cartésien entre A^* et B^*. Son élément neutre est $(\varepsilon, \varepsilon)$ est son produit se définit comme suit :

$$\forall (u_1, v_1), (u_2, v_2) \in A^* \times B^* : (u_1, v_1)(u_2, v_2) = (u_1u_2, v_1v_2).$$

Une relation entre A^* et B^* est un ensemble $R \subseteq A^* \times B^*$. La relation R est donc dite *rationnelle* si elle appartient à $\text{RAT}(A^* \times B^*)$.

Exemple 1.7. *La relation $R = \{(a^{n+1}, b^n c) \mid n \geq 0\}$ est rationnelle car elle peut se définir comme*

$$R = \{(a, b)\}^*\{(a, c)\}.$$

Transductions.

Une autre vision de la notion de relation est celle de *transductions*. On peut considérer une relation $R \subseteq A^* \times B^*$ comme une fonction partielle

$$\begin{array}{rccc} \tau : & A^* & \longrightarrow & 2^{B^*} \\ & u & \longmapsto & \{v \mid (u, v) \in R\} \end{array}$$

Les domaines et co-domaines de τ sont alors

$$\begin{aligned} \text{dom}(\tau) &= \{u \mid \exists v : (u, v) \in \tau\} \\ \text{co-dom}(\tau) &= \{v \mid \exists u : (u, v) \in \tau\}. \end{aligned}$$

Étant donné un langage $X \subseteq A^*$, l'image de X par τ est $\tau(X) = \bigcup_{u \in X} \tau(u)$.

Note. Nous confondons généralement ces deux notions. Nous parlerons de relation ou de transductions dans différents contextes en fonction de ce qui s'avérera plus pratique.

Différents types de transductions. Une transduction $\tau \subseteq A^* \times B^*$ est dite
- lettre-à-lettre si $\forall (u,v) \in \tau : |u| = |v|$;
- fidèle si $\forall v \in B^* : \tau^{-1}(v)$ est fini ;
- croissante si $\forall (u,v) \in \tau : |u| \leq |v|$;
- continue si $\forall (u,v) \in \tau :$ si $u \in A^+$ alors $v \in B^+$.

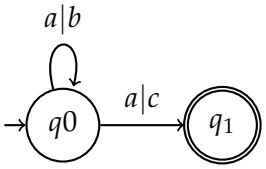
Les transductions rationnelles sont aussi décrites par transducteurs finis. Ceux-ci sont des automates finis munis d'une bande d'écriture sur laquelle ils écrivent tout au long du calcul. Formellement, un transducteur fini se décrit comme suit.

Définition 1.8 (Transducteurs finis)**.** *Un transducteur fini est une structure $\mathcal{T} = (Q, A, B, \Delta, q_0, F)$ dans laquelle*
- *A et B sont respectivement les alphabets d'entrée et de sortie ;*
- *Q, q_0, F sont respectivement l'ensemble d'états, l'état initial, et l'ensemble d'états acceptants ;*
- *$\Delta \subseteq Q \times (A \cup \{\varepsilon\}) \times (B \cup \{\varepsilon\}) \times Q$ est la relation de transitions. Les transitions sont notées $(q, a|b, p)$ signifiant que la machine lit le symbole a sur son ruban d'entrée, écrit le symbole b sur le ruban de sortie, et change son état interne de q à p.*

La transduction décrite par $\mathcal{T}$ est

$$\mathcal{R}(\mathcal{T}) = \{(u,v) \mid \exists q \in F : q_0 \overset{u|v}{\underset{*}{\Rightarrow}} q\}.$$

Exemple 1.9. *Soit le transducteur $\mathcal{T} = (Q, A, B, \Delta, q_0, F)$ tel que $Q = \{q_0, q_1\}$, $A = \{a\}$, $B = \{b, c\}$ et $\Delta = \{(q_0, a|b, q_0), (q_0, a|c, q_1)\}$; comme illustré ci-dessous.*



La transduction décrite est $\mathcal{R}(\mathcal{T}) = \{(a^{n+1}, b^n c) \mid n \geq 0\}$.

Théorème 1.3 (voir [Ber79])**.** *Une transductions est rationnelle si et seulement si elle peut être décrite par un transducteur fini.*

Théorème de Nivat.

Une caractérisation importante des transductions rationnelles, et celle qu'on utilisera le plus, est la caractérisation de Nivat.

Théorème 1.4 (Théorème de Nivat [Niv68])**.** *Soit $\tau \subseteq A^* \times B^*$. La transduction τ est rationnelle si et seulement si il existe un alphabet C, un langage régulier $R \subseteq C^*$ et deux homomorphismes (alphabétiques) $g : C^* \longrightarrow A^*$ et $f : C^* \longrightarrow B^*$ tels que*

$$\tau = \{(g(u), f(u)) \mid u \in R\}.$$

Remarque 1.10. *L'image d'un mot $u \in A^*$ par τ est alors $\tau(u) = f(R \cap g^{-1}(u))$; et les domaines et co-domaines de τ sont*

$$\begin{aligned} \mathrm{dom}(\tau) &= g(R) \\ \text{co-dom}(\tau) &= f(R). \end{aligned}$$

Notations. Nous employons quand nécessaire la notation d'Eilenberg. La transduction τ susmentionnée pourra aussi être notée $\tau = f \circ \cap R \circ g^{-1}$.

Le théorème de Nivat peut s'expliquer comme suit : le langage régulier R décrit l'ensemble de calculs acceptants du transducteur–vu comme un langage sur l'alphabet des transitions. L'homomorphisme g associe à chaque calcul acceptant $u \in R$, le mot d'entrée associé, et h y associe le mot de sortie. L'ensemble des couples de la relation est alors l'ensemble des couples $(g(u), h(u))$ tels que $u \in R$.

Exemple 1.11. *La transduction* $\tau = \{(a^{n+1}, b^n c) \mid n \geq 0\}$ *peut être décrite comme*

$$\tau = \{(g(u), f(u)) \mid u \in R\}$$

$$\text{avec } R \equiv \alpha^*\beta, \quad g : \begin{cases} \alpha \mapsto a \\ \beta \mapsto a \end{cases} \quad \text{et } f : \begin{cases} \alpha \mapsto b \\ \beta \mapsto c \end{cases}$$

Théorème 1.5 (Boasson–Nivat [BN73])**.** *Soit* $\tau \subseteq A^* \times B^*$. *La transduction* τ *est rationnelle, fidèle et continue si et seulement si il existe un alphabet* C, *un langage régulier* $R \subseteq C^*$ *un homomorphisme* $g : C^* \longrightarrow A^*$ *et un homomorphisme strictement alphabétique* $f : C^* \longrightarrow B^*$ *tels que*

$$\tau = \{(g(u), f(u)) \mid u \in R\}.$$

Théorème 1.6 (Leguy [Leg80])**.** *Soit* $\tau \subseteq A^* \times B^*$. *La transduction* τ *est rationnelle croissante si et seulement si il existe un alphabet* C, *un langage régulier* $R \subseteq C^*$ *un homomorphisme alphabétique* $g : C^* \longrightarrow A^*$ *et un homomorphisme non-effaçant* $f : C^* \longrightarrow B^*$ *tels que*

$$\tau = \{(g(u), f(u)) \mid u \in R\}.$$

1.4 Familles agréables de langages (propriétés de clôture)

La notion de familles agréables de langages [1] a été introduite par Greibach et Ginsburg [GG67](voir aussi [Gin75]). Ceux-ci, sous l'observation que la plupart des familles de langages de la hiérarchie de Chomsky partagent des propriétés de clôtures communes, se sont intéressés à l'incidence de ces propriétés de clôture sur les propriétés globales de ces familles. Nous présentons ici en substance la notion de familles agréables de langages.

Définition 1.12 (Cônes rationnels)**.**

- *Un cône rationnel croissant est une famille de langage fermée par homomorphismes non-effaçants, intersection avec des réguliers et images inverses d'homomorphismes alphabétiques.*
- *Un cône rationnel fidèle est un cône rationnel croissant fermé par homomorphismes inverses.*
- *Un cône rationnel est un cône rationnel fidèle fermé par homomorphismes.*

De part les caractérisations homomorphiques des transductions rationnelles (rationnelles croissantes, rationnelles fidèles), les différents cônes rationnels présentés ci-dessus se définissent de façons équivalentes comme suit.

Définition 1.13 (Cônes rationnels)**.** *Un cône rationnel (resp. rationnel fidèle, rationnel croissant) est une famille de langages fermée par transductions rationnelles (rationnelles fidèles et continues, rationnelles croissantes).*

1. en anglais : *full-AFL (full abstract family of languages)*

1.4.1 Cônes principaux

Pour toute famille de langage $\mathcal{F}$, $\mathscr{C}(\mathcal{F})$ (resp. $\mathscr{C}^f(\mathcal{F})$, $\mathscr{C}^<(\mathcal{F})$) désigne le plus petit cône rationnel (resp. rationnel fidèle, rationnel croissant) qui contient $\mathcal{F}$.

Définition 1.14 (Cônes principaux)**.** *Un cône rationnel (resp. rationnel fidèle, rationnel croissant) $\mathcal{F}$ est dit principal si il existe un langage L tel que*

$$\mathcal{F} = \mathscr{C}(\{L\}) \quad \textit{(resp. } \mathcal{F} = \mathscr{C}^f(\{L\}), \mathcal{F} = \mathscr{C}^<(\{L\})\textit{).}$$

Quand tel est le cas, nous écrivons $\mathscr{C}(L)$ plutôt que $\mathscr{C}(\{L\})$.

Du fait que la classe de transductions rationnelles (resp. rationnelles fidèles, rationnelles croissantes) soit fermée par composition, il s'en suit les équivalences suivantes.

$$\begin{aligned}
\mathscr{C}(L) &= \{\tau(L) \mid \tau \text{ est une transduction rationnelle}\}\\
\mathscr{C}^f(L) &= \{\tau(L) \mid \tau \text{ est une transduction rationnelle fidèle et continue}\}\\
\mathscr{C}^<(L) &= \{\tau(L) \mid \tau \text{ est une transduction rationnelle croissante}\}.
\end{aligned}$$

Définition 1.15 (Familles Agréables de Langages (FAL))**.** *Une Famille Agréable de Langages (fidèle, croissante) est un cône rationnel (fidèle, croissant) clos par union, concaténation et étoile de Kleene stricte.*

Définition 1.16 (Fermeture par substitution)**.** *Soit $\mathcal{F}$ une famille de langages. Une $\mathcal{F}$-substitution est une substitution $\nu : A \longrightarrow 2^{B^*}$ telle que pour tout $a \in A : \nu(a) \in \mathcal{F}$.*

Une famille de langages $\mathcal{F}$ est dite fermée par substitutions si elle est fermée par $\mathcal{F}$-substitutions.

Chapitre 2

Langages algébriques et indexés

Dans ce chapitre, nous présentons premièrement les langages et transductions algébriques. Pour cette présentation des langages algébriques, nous présentons les grammaires dites *hors-contexte*, les automates à pile ainsi que le théorème de Chomsky–Schützenberger qui est au cœur de notre étude. Nous présentons ensuite les langages indexés. Ceux-ci sont l'objet des caractérisations que nous présentons dans la première partie de contributions de ce manuscrit.

2.1 Langages algébriques

Nous présentons ici la famille des langages dits *hors-contexte* ou *algébriques*. Ceux-ci tiennent ces noms du fait d'être décrits, de façon équivalente, par des systèmes de réécriture dans lesquels les dérivations des variables sont indépendantes de leurs contextes ; et par des systèmes d'équations polynomiales.

Nous présentons ici de façon brève les langages et transductions algébriques ainsi que le théorème de Chomsky–Schützenberger et quelques variations qui sont présentes tout au long de ce manuscrit. Nous employons le matériel de [ABB97, Ber79] pour cette présentation et toute affirmation qui n'est pas accompagnée d'une référence en provient.

2.1.1 Grammaires algébriques et systèmes d'équations

Définition 2.1 (Syntaxe des grammaires algébriques)**.** *Une grammaire algébrique est une structure* $\mathcal{G} = (N, T, S, P)$ *dans laquelle*

- *N est un ensemble fini de symboles appelés* non-terminaux *ou* variables *;*
- *T est l'alphabet de symboles* terminaux *;*
- $S \in N$ *est le non-terminal dit* initial*, aussi appelé* axiome *;*
- $P \subseteq N \times (N \cup T)^*$ *est l'ensemble de productions. Les productions* (X, Θ) *sont notées* $X \longrightarrow \Theta$*. Dans une telle production,* X *est appelé* membre gauche *et* Θ membre droit *de la production.*

Dérivations. Une *phrase* ou *forme sententielle* est un mot Θ de $(N \cup T)^*$. Soit « $\underset{\mathcal{G}}{\Longrightarrow}$ » la relation binaire entre phrases telle que pour toute paire de phrases $\Theta, \Theta' \in (N \cup T)^*$, $\Theta \underset{\mathcal{G}}{\Longrightarrow} \Theta'$ si il existe $\theta_0, \theta_1, \theta \in (N \cup T)^*, X \in N$ tels que

$$\Theta = \theta_0 X \theta_1; \quad \Theta' = \theta_0 \theta \theta_1; \quad \text{et } X \longrightarrow \theta \in P.$$

C'est à dire que Θ' peut être obtenue en substituant dans Θ un non-terminal par le membre droit d'une production à laquelle il est associé. La relation $\underset{\mathcal{G}}{\Longrightarrow}$ décrit de cette façon une étape de dérivation de la grammaire. Soit $\overset{*}{\underset{\mathcal{G}}{\Longrightarrow}}$ sa clôture réflexive et transitive. Une phrase Θ' est dite dérivable depuis une phrase Θ si $\Theta \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} \Theta'$. À chaque $\Theta \in (N \cup T)^*$ est associé le langage

$$\mathscr{L}_{\mathcal{G}}(\Theta) = \{u \in T^* \mid \Theta \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} u\}.$$

En particulier :
- pour tout $a \in T \cup \{\varepsilon\} : \mathscr{L}_{\mathcal{G}}(a) = \{a\}$;
- pour toute paire de phrases $\Theta_1, \Theta_2 : \mathscr{L}_{\mathcal{G}}(\Theta_1\Theta_2) = \mathscr{L}_{\mathcal{G}}(\Theta_1)\mathscr{L}_{\mathcal{G}}(\Theta_2)$;
- et pour tout non-terminal $X \in N : \mathscr{L}_{\mathcal{G}}(X) = \bigcup_{X \longrightarrow \Theta \in P} \mathscr{L}_{\mathcal{G}}(\Theta)$.

Le langage *généré* par la grammaire $\mathcal{G}$ est noté $\mathscr{L}(\mathcal{G})$ et se définit comme

$$\mathscr{L}(\mathcal{G}) = \mathscr{L}_{\mathcal{G}}(S).$$

Notations.
- Les non-terminaux sont toujours notés en majuscule : $X, Y, Z, \ldots$; les symboles terminaux en minuscule $a, b, c, \ldots$; et les phrases sont toujours notées $\Theta, \Theta_1, \Theta_2, \ldots$.
- Si plusieurs productions ont le même membre gauche, par exemple

  $$p_1 : X \longrightarrow \Theta_1; \quad p_2 : X \longrightarrow \Theta_2; \cdots \quad p_n : X \longrightarrow \Theta_n;$$

  nous les notons sous *forme contractée*

  $$p_1, p_2, \ldots, p_n : X \longrightarrow \Theta_1 + \Theta_2 + \cdots + \Theta_n.$$

- Nous écrivons $\mathscr{L}(\Theta)$ plutôt que $\mathscr{L}_{\mathcal{G}}(\Theta)$ lorsque cela ne lève aucune ambiguïté ; si par exemple il n'y a qu'une seule grammaire définie dans le contexte.

Nomenclature. Une production $X \longrightarrow \Theta$ est dite *linéaire* si Θ contient au plus un non-terminal. Elle est dite *quadratique* si Θ contient au plus deux non-terminaux.

Exemple 2.2. *Soit la grammaire donnée par les productions suivantes*

$$S \longrightarrow XY + Y;$$
$$X \longrightarrow a;$$
$$Y \longrightarrow aab.$$

Nous avons $\mathscr{L}(X) = \{a\}$, $\mathscr{L}(Y) = \{ab\}$ *et* $\mathscr{L}(S) = (\mathscr{L}(X)\mathscr{L}(Y)) \cup \mathscr{L}(Y) = \{aaab, aab\}$.

Définition 2.3 (Langages algébriques)**.** *Un langage L est dit* algébrique *ou* hors-contexte *si il existe une grammaire algébrique $\mathcal{G}$ telle que $L = \mathscr{L}(\mathcal{G})$.*

Nous désignons par ALG *la famille des langages algébriques.*

Définition 2.4 (Equivalence de grammaires)**.** *Deux grammaires (algébriques) $\mathcal{G}$ et $\mathcal{G}'$ sont dites équivalentes si $\mathscr{L}(\mathcal{G}) = \mathscr{L}(\mathcal{G}')$.*

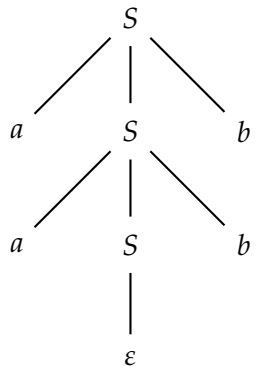
Exemple 2.5. *Soit la grammaire* $\mathcal{G} = (N, T, S, P)$ *avec* $N = \{S\}$, $T = \{a, b\}$ *et* P *est constitué des productions*

$$p_1 : S \longrightarrow \varepsilon; \quad p_2 : S \longrightarrow aSb.$$

Le langage généré par $\mathcal{G}$ *est* $\bigcup_{n\geq 0} a^n b^n$. *Une dérivation pour le mot aabb est la suivante*

$$\begin{aligned} S &\underset{\mathcal{G}}{\Longrightarrow} aSb && (par\ p_2)\\ &\underset{\mathcal{G}}{\Longrightarrow} aaSbb && (par\ p_2)\\ &\underset{\mathcal{G}}{\Longrightarrow} aabb && (par\ p_1) \end{aligned}$$

Arbres de dérivations. Une dérivation d'une grammaire algébrique peut être représentée par un arbre dans lequel l'utilisation d'une production $X \longrightarrow \theta_1\theta_2\cdots\theta_n$ est représentée par un nœud X ayant pour fils n nœuds $\theta_1, \theta_2, \ldots, \theta_n$. La dérivation de l'exemple 2.5 peut être représentée par l'arbre ci-dessous.



Théorème 2.1. ALG *est une famille agréable de langages.*

Régularité des dérivations

Soit $\mathcal{G} = (N, T, P, S)$ une grammaire algébrique. Une étape de dérivation $\Theta \underset{\mathcal{G}}{\Longrightarrow} \Theta'$ est dite *gauche*, et notée $\Theta \underset{g,\mathcal{G}}{\Longrightarrow} \Theta'$, si le non-terminal dérivé est le plus à gauche : Θ est de la forme $uX\Theta_0$ avec $u \in T^*$, $X \in N$ et la phrase Θ' peut être obtenue en substituant dans Θ, le non-terminal X par le membre droit d'une production dont il est le membre gauche. Une dérivation gauche est alors une dérivation $\Theta \overset{*}{\underset{g,\mathcal{G}}{\Longrightarrow}} \Theta'$ où $\overset{*}{\underset{g,\mathcal{G}}{\Longrightarrow}}$ désigne la clôture réflexive et transitive de $\underset{g,\mathcal{G}}{\Longrightarrow}$. De façon similaire, on parle de dérivation *droite* ($\theta \overset{*}{\underset{d,\mathcal{G}}{\Longrightarrow}} \theta$) lorsque le non-terminal dérivé à chaque étape de dérivation est celui le plus à droite. L'ordre par lequel on dérive les non-terminaux n'a pas d'importance. Les suites de non-terminaux obtenues sont les mêmes : $\forall \Theta \in (N \cup T)^*$

$$\mathcal{L}(\Theta) \;=\; \{u \in T^* \mid \Theta \overset{*}{\underset{g,\mathcal{G}}{\Longrightarrow}} u\} \;=\; \{u \in T^* \mid \Theta \overset{*}{\underset{d,\mathcal{G}}{\Longrightarrow}} u\}.$$

Si on s'intéresse toutefois aux suites de symboles terminaux et non-terminaux dérivables, l'ordre de dérivation à une incidence sur la complexité de ces ensembles de suites comme en témoignent le théorème suivant.

Théorème 2.2 ([Gre67]). *Soit $\mathcal{G} = (N, T, P, S)$ une grammaire algébrique. Pour tout $x \in (N \cup T)$ et $y \in (N \cup T)^*$, soient*

— *$L_{x,y} \subseteq (N \cup T)^*$ l'ensemble de phrases Θ telles qu'il existe une dérivation gauche $x \overset{*}{\underset{g,\mathcal{G}}{\Longrightarrow}} \Theta_0 \underset{g,\mathcal{G}}{\Longrightarrow} y\Theta$ telle que y n'est pas préfixe de Θ_0 (les derniers symboles de y ont été produits lors de la dernière étape de dérivation $\Theta_0 \underset{g,\mathcal{G}}{\Longrightarrow} y\Theta$) ;*

— *$R_{x,y} \subseteq (N \cup T)^*$ l'ensemble de phrases Θ telles qu'il existe une dérivation droite $x \overset{*}{\underset{d,\mathcal{G}}{\Longrightarrow}} \Theta_0 \underset{d,\mathcal{G}}{\Longrightarrow} \Theta y$ telle que y n'est pas suffixe de Θ_0.*

Les langages $L_{x,y}$ et $R_{x,y}$ sont réguliers.

Exemple 2.6. *Soit la grammaire décrite par les productions suivantes*

$$S \longrightarrow X + aY + SX;$$
$$X \longrightarrow aXa + b;$$
$$Y \longrightarrow a.$$

Les dérivations gauches générant le symbole a sont de la forme

$$(1) S \underset{g,\mathcal{G}}{\Longrightarrow} X \underset{g,\mathcal{G}}{\Longrightarrow} aXa;$$
$$(2) S \underset{g,\mathcal{G}}{\Longrightarrow} aY;$$
$$(3) S \overset{*}{\underset{g,\mathcal{G}}{\Longrightarrow}} SX^n \underset{g,\mathcal{G}}{\Longrightarrow} aYX^n \quad \text{avec } n \geq 1;$$
$$(4) S \overset{*}{\underset{g,\mathcal{G}}{\Longrightarrow}} SX^n \underset{g,\mathcal{G}}{\Longrightarrow} XX^n \underset{g,\mathcal{G}}{\Longrightarrow} aXaX^n \quad \text{avec } n \geq 1;$$

Le langage $L_{S,a}$ est alors $(Xa + Y)X^*$. On peut observer de cette même façon que le langage $L_{X,b} = \varepsilon$ et $L_{Y,b} = \emptyset$. Nous pouvons dans le cas présent calculer le langage $L_{S,ab}$ en substituant dans $L_{S,a}$ (dans $(Xa + Y)X^*$) les premiers symboles $x \in N$ par $L_{x,b}$ comme suit.

$$L_{S,ab} = (L_{X,b}a + L_{Y,b})X^* = (a + \emptyset)X^* = aX^*.$$

Formes normales

Il existe plusieurs formes normales pour les grammaires algébriques. Nous présentons parmi celles-ci, la forme normale de Chomsky et deux formes normales de Greibach.

Forme réduite. Nous disons qu'une grammaire algébrique (N, T, P, S) est *réduite* si pour tout non-terminal $X \in N$, il existe une dérivation de la forme

$$S \overset{*}{\Longrightarrow} \Theta_1 X \Theta_2, \quad \Theta_1, \Theta_2 \in (N \cup T)^*,$$

et une dérivation $X \overset{*}{\Longrightarrow} u, \quad u \in T^*$.

Théorème 2.3 (Forme normale de Chomsky)**.** *Pour toute grammaire algébrique G, il existe une grammaire algébrique équivalente G' dans laquelle toute production est sous l'une des formes suivantes :*

$$(1) S \longrightarrow \varepsilon; \qquad (2) X \longrightarrow YZ, \text{ avec } Y, Z \in N - \{S\} \qquad (3) X \longrightarrow a, \text{ avec } a \in T.$$

Théorème 2.4 (Forme normale de Greibach)**.** *Pour toute grammaire algébrique G, il existe une grammaire algébrique équivalente G' dans laquelle toute production est sous l'une des formes suivantes :*

$$(1) S \longrightarrow \varepsilon; \quad (2) X \longrightarrow a\Theta, \text{ avec } a \in T, \Theta \in (N - \{S\})^*.$$

Théorème 2.5 (Forme normale double de Greibach)**.** *Pour toute grammaire algébrique G, il existe une grammaire algébrique équivalente G' dans laquelle toute production est sous l'une des formes suivantes :*

$$(1) X \longrightarrow \varepsilon; \quad (2) X \longrightarrow a\Theta b, \text{ avec } a, b \in T, \Theta \in (N - \{S\})^*.$$

Systèmes d'équations.

Présentons brièvement les systèmes d'équations algébriques. Nous nous en servirons le plus souvent afin d'illustrer nos propos quand le besoin sera de prêter attention aux langages décrits par les non-terminaux d'une grammaire plutôt qu'à la forme de ses dérivations. Les langages algébriques peuvent en effet être décrits comme des éléments de plus petites solutions de systèmes d'équations polynomiaux–d'où le terme algébrique introduit par Chomsky et Schützenberger [CS63].

Soient $N = \{X_1, X_2, \ldots, X_{|V|}\}$ et T deux alphabets disjoints. Un système d'équations sur (N, T) est un ensemble d'équations de la forme

$$X_i = P_i, \quad i \in \{1, \ldots, |N|\}$$

où chaque P_i est un sous-ensemble fini de $(N \cup T)^*$. Les éléments de N sont appelés *variables* du système. Une équation

$$X_i = \{\Theta_{i,1}, \Theta_{i,2}, \ldots, \Theta_{i,m}\}$$

sera notée

$$X_i = \Theta_{i,1} + \Theta_{i,2} + \cdots + \Theta_{i,m}.$$

Solutions. Une solution du système d'équations est un vecteur $\boldsymbol{L} = (L_1, L_2, \ldots, L_{|N|})$ tel que si on substitue chaque variable X_i par L_i dans les membres gauches et droits des équations, l'ensemble d'équations obtenu est un ensemble d'égalités. Plus formellement, le vecteur $\boldsymbol{L}$ définit une substitution $\nu_{\boldsymbol{L}} : (N \cup T)^* \longrightarrow 2^{(N \cup T)^*}$ tel que

$$\nu_{\boldsymbol{L}}(a) = \{a\} \ \ \forall a \in T; \qquad \nu_{\boldsymbol{L}}(X_i) = L_i \ \ \forall i \in \{1, \ldots, |N|\}.$$

Le vecteur $\boldsymbol{L}$ est alors une solution du système si

$$\nu_{\boldsymbol{L}}(X_i) = L_i = \nu_{\boldsymbol{L}}(P_i) \quad \forall i \in \{1, \ldots, |N|\}.$$

Exemple 2.7. *$X = XX + a$ admet a^* comme solution puisque $a^* = a^*a^* \cup a$.*

(Non-)Unicité des solutions. Un système d'équations peut admettre plusieurs solutions. Par exemple, l'équation de l'exemple 2.7 admet comme solution tout langage fermé par concaténation qui contient a ; *ex :* les langages $\{a, ab\}^*$ et $\{a\}^+$ en sont des solutions. De façon générale, on s'intéresse plutôt à la *plus petite solution* d'un système d'équations.

Exemple 2.8. *Le langage a^+ est la plus petite solution de l'équation $X = XX + a$. En effet a^+ est le plus petit langage L tel que $a \in L$ et pour tous $u, v \in L$: $uv \in L$.*

La forme des équations peut toutefois garantir l'unicité d'une solution. Un système d'équations est dit *strict* si le membre droit de chaque équation est une somme d'éléments de $\{\varepsilon\} \cup (N \cup T)^* T (N \cup T)^*$. Si tel est le cas, le système admet une unique solution.

Exemple 2.9. *L'équation $X = aXb + \varepsilon$ admet $\sum_{n\geq 0} a^n b^n$ comme unique solution puisque $\sum_{n\geq 0} a^n b^n = a(\sum_{n\geq 0} a^n b^n)b + \varepsilon$ et que cette équation est stricte.*

Correspondance avec les grammaires. La correspondance entre grammaires algébriques et systèmes d'équations consiste à remplacer tout ensemble de productions

$$X_i \longrightarrow \Theta_{i,1} + \Theta_{i,2} + \cdots + \Theta_{i,m}$$

en équation

$$X_i = \Theta_{i,1} + \Theta_{i,2} + \cdots + \Theta_{i,m}$$

et inversement. De plus, étant donné une grammaire $(N = \{X_1, \ldots, X_n\}, T, P, S)$, le vecteur $\boldsymbol{L} = (\mathscr{L}(X_1), \ldots, \mathscr{L}(X_n))$ est la plus petite solution du système d'équations associé.

Exemple 2.10. *Soit la grammaire donnée par les productions*

$S \longrightarrow XY$;
$X \longrightarrow aXY + c$;
$Y \longrightarrow b$.

Le système d'équation associé est

$$\begin{cases} S = XY; & (1) \\ X = aXY + c; & (2) \\ Y = b. & (3) \end{cases}$$

En substituant le membre droit de (3) *dans* (2) *on obtient l'équation* $(2')X = aXb + c$. *Sa plus petite solution est le langage $\sum_{n\geq 0} a^n c b^n$. On obtient ainsi l'ensemble d'équations*

$$\begin{cases} S = XY; \\ X = \sum_{n\geq 0} a^n c b^n; \\ Y = b. \end{cases}$$

Par substitution, on obtient finalement

$$\begin{cases} S = (\sum_{n\geq 0} a^n c b^n)b; \\ X = \sum_{n\geq 0} a^n c b^n; \\ Y = b. \end{cases}$$

2.1.2 Automates à pile

Un automate à pile est une structure $\mathcal{A} = (Q, A, q_0, \Gamma, \Delta, F)$ dans laquelle
— Q est un ensemble fini d'états,
— A est l'alphabet de travail,
— $q_0 \in Q$ est l'état initial, $F \subseteq Q$ est l'ensemble d'états acceptants,
— Γ est l'alphabet de pile, celui-ci ne contient pas le symbole $\perp$;
— et $\Delta \subseteq Q \times A \cup \{\varepsilon\} \times (\overline{\Gamma}^* \Gamma^*) \times Q$ est la relation de transition.

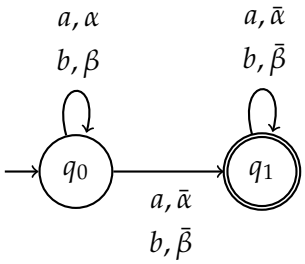
Configurations. Une configuration est une description instantanée de la machine. Elle se décrit par un triplet $c = (q, u, \omega)$ dans lequel $q \in Q$ est l'état courant, $\omega \in \Gamma^* \perp$ le contenu de la pile et $u \in A^*$ est le mot restant à lire sur le ruban d'entrée. La relation $\underset{\mathcal{A}}{\vdash}$ décrit une étape de calcul ; étant donné deux configurations $c_1 = (q, au, \alpha\omega)$ et $c_2 = (p, u, \beta\omega)$;

$$c_1 \underset{\mathcal{A}}{\vdash} c_2 \text{ si et seulement il existe } (q, a, \bar{\alpha}\beta, p) \in \Delta.$$

Soit $\underset{A}{\vdash}^*$ la clôture réflexive et transitive de $\underset{A}{\vdash}$. Le langage reconnu par $\mathcal{A}$ est

$$\mathscr{L}(\mathcal{A}) = \{u \in A^* \mid \exists q \in F : (q_0, u, \perp) \underset{\mathcal{A}}{\vdash}^* (q, \varepsilon, \perp)\}.$$

Exemple 2.11. *Soit $\mathcal{A} = (Q, A, q_0, \Gamma, \Delta, F)$ avec $A = \{a, b\}$, $Q = \{q_0, q_1\}$, $F = \{q_1\}$, $\Gamma = \{\alpha, \beta\}$ et Δ décrit par le graphe de transitions suivant.*



Le langage reconnu est $\{uu^R \mid u \in \{a, b\}^+\}$. Un calcul pour le mot abba est

$$\begin{aligned}(q_0, abba, \perp) &\underset{\mathcal{A}}{\vdash} (q_0, bba, \alpha\perp)\\ &\underset{\mathcal{A}}{\vdash} (q_0, ba, \beta\alpha\perp)\\ &\underset{\mathcal{A}}{\vdash} (q_1, a, \alpha\perp)\\ &\underset{\mathcal{A}}{\vdash} (q_1, \varepsilon, \perp).\end{aligned}$$

2.2 Transductions algébriques

La notion d'algébricité peut être étendue aux relations tout comme la notion de rationalité. On parle alors de relations (transductions) algébriques. Une transduction $\tau \subseteq A^* \times B^*$ est dite *algébrique* si elle peut être générée par une grammaire algébrique pour laquelle les suites de terminaux sont des éléments du monoïde $A^* \times B^*$; ou par une machine à piles capables de produire un mot de sortie lors de la lecture d'un mot sur la bande d'entrée. De plus, les transductions algébriques bénéficient elles aussi d'un analogue au théorème de Nivat. Nous présentons brièvement ces différents formalismes comme suit.

Définition 2.12 (Transducteurs Algébriques)**.** *Un transducteur algébrique est une structure $\mathcal{T} = (N, A, B, S, P)$ dans laquelle*

— *N est l'ensemble de non-terminaux,*

— *A et B sont respectivement, l'alphabet d'entrée et l'alphabet de sortie,*

— $S \in N$ *est le non-terminal initial*
— *et* P *est cette fois ci un sous-ensemble de* $N \times (N \cup (A^* \times B^*))^*$.

Les dérivations s'effectuent de façon similaire à celles des grammaires algébriques, à la difference que le produit appliqué aux terminaux est celui du monoïde $A^* \times B^*$. *Pour tout non-terminal* $X \in N$: $\mathcal{R}(X)$ *désigne l'ensemble de couple dérivable depuis celui-ci*

$$\mathcal{R}(X) = \{(u, v) \in A^* \times B^* \mid X \overset{*}{\underset{\mathcal{T}}{\Longrightarrow}} (u, v)\}.$$

La relation générée par le transducteur est $\mathcal{R}(\mathcal{T}) = \mathcal{R}(S)$.

Une transduction est dite algébrique *si elle peut être générée par un transducteur algébrique.*

Exemple 2.13. *Soit le transducteur algébrique donné par les productions*

$$S \longrightarrow (a, c)S(b, \varepsilon) + X;$$
$$X \longrightarrow (b, d).$$

La relation générée est $\{(a^n b^{n+1}, c^n d) \mid n \geq 0\}$.

De même que les transductions rationnelles, les transductions algébriques bénéficient d'une caractérisation homomorphique.

Théorème 2.6 (Eilenberg [Ber79])**.** *Soit* τ *une relation.* τ *est algébrique si, et seulement si, il existe un langage algébrique* K *et deux homomorphismes* f *et* g *tels que*

$$\tau = \{(g(u), f(u)) \mid u \in K\}.$$

Dans ce qui suit, nous présentons différentes caractérisations des langages algébriques que nous généraliserons à d'autres familles de langages.

2.3 Langage de Dyck

Réductions. Soit le système de réduction $R_\eta = \{a\bar{a} \rightsquigarrow \varepsilon \mid a \in A\}$. Ce dernier supprime dans un mot toute occurrence de facteurs $a\bar{a}$, avec $a \in A$. Un mot est dit R_η*-réduit* s'il ne contient aucune occurrence de facteur $a\bar{a}$ quel que soit $a \in A$. Autrement, il est dit *réductible* par R_η. Le système R_η est dit *confluent*, c'est à dire que quel que soit le facteur par lequel on commence la réduction, on aboutit au même mot R_η-réduit. Tout mot u est alors équivalent (modulo R-réduction) à un unique mot R-réduit que nous notons $\eta_A(u)$. Nous abuserons de cette notation et écrirons $\eta(u)$ au lieu de $\eta_A(u)$.

Exemple 2.14. *Soit* $u = ab\bar{b}\bar{a}aa\bar{a}$, *alors* $\eta(u) = ba$, *comme illustré sur la figure 2.1.*

Réduction bilatérale. Considérons cette fois ci le système de réduction $R_\rho = \{a\bar{a} \rightsquigarrow \varepsilon, a \in \widehat{A}\}$. Le système R_ρ est la version *bilatérale* de R_η ; il réduit à la fois les occurrences $a\bar{a}$ et $\bar{a}a$. De façon similaire, nous désignons par $\rho(u)$ l'unique mot R_ρ-réduction d'un mot u.

Exemple 2.15. *Soit* $u = \bar{a}b\bar{b}c\bar{c}a$, *alors* $\rho(u) = \varepsilon$.

La confluence des réductions R_η et R_ρ donne lieu au lemme suivant.

Lemme 2.16. *Soit un mot* $u \in \widehat{A}^*$ *et* $u = u_1u_2u_3$ *une décomposition de celui-ci. Alors* $\rho(u) = \rho(u_1\rho(u_2)u_3)$. *De même,* $\eta(u) = \eta(u_1\eta(u_2)u_3)$.

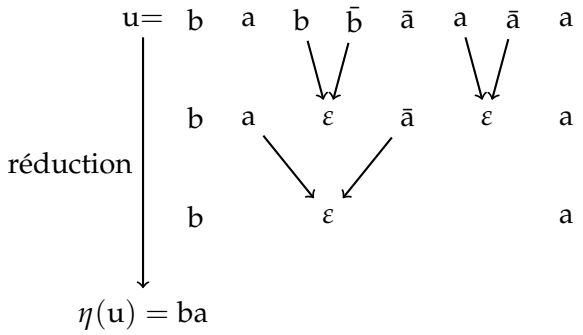


FIGURE 2.1 – Réduction du mot $ab\bar{b}\bar{a}a\bar{a}a$ en ba.

Le lemme ci-dessus exprime le fait que quel que soit le facteur par lequel on commence la réduction, le résultat de celle-ci est le même.

Le langage de Dyck et le langage de Dyck bilatéral se définissent comme les classes d'équivalence de ε par les réductions R_η et R_ρ.

Définition 2.17 (Langage de Dyck)**.** *Soit A un alphabet, le langage de Dyck sur A, dénoté $\mathcal{D}_A$, est*

1. *l'ensemble de mots $u \in \widehat{A}^*$ tels que $\eta(u) = \varepsilon$.*
2. *l'ensemble de mots $u \in \widehat{A}^*$ tels que $\rho(u) = \varepsilon$ et pour tout préfixe $u_0 \preccurlyeq u : |u_0|_a \geq |u_0|_{\bar{a}}$ pour tout $a \in A$.*

Cette deuxième définition exprime le fait que le mot u se ρ-réduise au mot vide ε, mais qu'en plus de cela, les symboles positifs $a \in A$ précèdent toujours les symboles négatifs $\bar{a} \in \overline{A}$ lors de cette ρ-réduction. Cela est alors équivalent à dire que le mot u se η-réduit au mot vide.

Lemme 2.18. *Soit A un alphabet. Pour tout $u \in \mathcal{D}_A$ et pour tout préfixe $u_0 \preccurlyeq u : \eta(u_0) \in A^*$.*

Algébricité. Le langage de Dyck est un langage algébrique. Puisqu'étant la classe d'équivalence de ε par la R_η, il se définit aussi comme le plus petit langage D tel que $\varepsilon \in D$ et pour tous $u, v \in D$ et pour tout $a \in A : uv \in D$ et $au\bar{a} \in D$. Ainsi, il peut être décrit comme la plus petite solution de

$$D = \varepsilon + DD + \sum_{a \in A} aD\bar{a}.$$

L'équation ci-dessus met aussi en évidence ce fait : pour tout mot $u \in \mathcal{D}_A$, de façon non-exclusive, nous avons soit $u \in \{\varepsilon\}$, soit $u \in \mathcal{D}_A\mathcal{D}_A$, soit $u \in \bigcup_{a \in A} \mathcal{D}_A$. Il en suit le lemme de décomposition suivant.

Lemme 2.19. *Soit A un alphabet, tout mot $u \in \mathcal{D}_A$ admet une des factorisations suivantes :*
— *$u = \varepsilon$;*
— *$u = vw$ avec $v, w \neq \varepsilon$ et $v, w \in \mathcal{D}_A$;*
— *$u = av\bar{a}$ avec $a \in A$ et $v \in \mathcal{D}_A$.*

Notations. Nous désignons par $\mathcal{D}_k$ le langage de Dyck sur un alphabet de taille k, et nous écrivons $\mathcal{D}$ si l'alphabet est relatif au contexte et que cela ne soulève aucune ambiguïté.

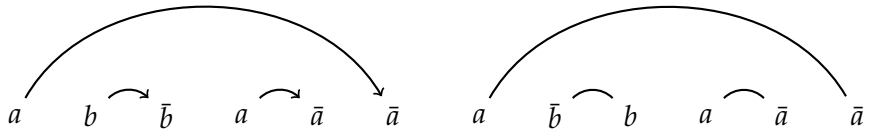


FIGURE 2.2 – À gauche : un mot de Dyck ; à droite : un mot de Dyck bilatéral.

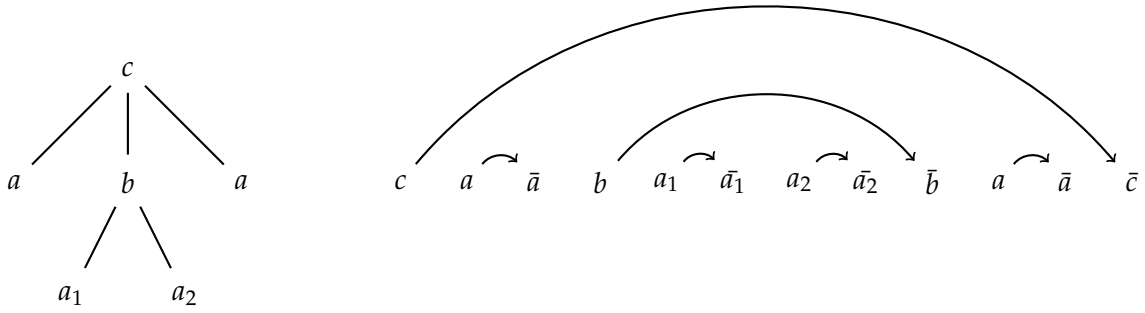


FIGURE 2.3 – A gauche : un arbre ; à droite : sa description sous forme de mot de Dyck.

Langage de Dyck et arbres. Nous n'utiliserons pas de formalisme d'arbres dans ce manuscrit. Nous pouvons toutefois mentionner qu'un des intérêts souvent portés au langage de Dyck est sa capacité à décrire des arbres. En effet, un arbre $t[t_1, t_2, \ldots, t_n]$ peut être décrit comme un mot de Dyck via le codage *cod* tel que

$$cod(t[t_1, t_2, \ldots, t_n]) = t\, cod(t_1) cod(t_2) \cdots cod(t_n)\, \bar{t}$$

(voir la figure 2.3 pour une illustration). Ce même fait explique la présence du langage de Dyck dans les théorèmes de représentation de familles de langages générées par différents types de grammaires ; car ces théorèmes de représentation consistent le plus souvent à décrire les (arbres de) dérivations de ces grammaires.

Nous considérons aussi le langage de Dyck *bilatéral*. Celui-ci est l'ensemble de mots se réduisant au mot vide par la réduction R_ρ (voir la figure 2.2 pour une illustration).

Définition 2.20 (Langage de Dyck bilatéral)**.** *Soit A un alphabet. Le langage de Dyck* bilatéral *sur A, dénoté $\mathcal{T}_A$, est l'ensemble de mots u tels que $\rho(u) = \varepsilon$.*

Le langage de Dyck bilatéral (sur un alphabet A) est bien évidemment lui aussi algébrique et est décrit par l'équation

$$\mathcal{T} = \varepsilon + \mathcal{T}\mathcal{T} + \sum_{a \in \widehat{A}} a\mathcal{T}\bar{a}.$$

De même, $\mathcal{T}_k$ désigne le langage de Dyck bilatéral sur k paires de parenthèses et nous dirons souvent qu'un mot u appartient à $\mathcal{T}$ si l'omission de son alphabet ne soulève aucune ambiguïté. De façon générale, nous dirons qu'un mot u appartient à $\mathcal{D}$ (resp. $\mathcal{T}$) plutôt que $\eta(u) = \varepsilon$ (resp. $\rho(u) = \varepsilon$) ; sauf cas exceptionnels où nous aurons besoin d'argumenter sur la syntaxe de celui-ci.

Lemme 2.21. *Soient A et B deux alphabets disjoints. Alors, $\pi_{\widehat{B}}(\mathcal{D}_{A\cup B}) = \mathcal{D}_B$.*

Intuitivement, ceci signifie que si un mot est bien parenthésé sur l'ensemble de parenthèses $\widehat{A \cup B}$, alors il l'est nécessairement sur l'ensemble de parenthèses $\widehat{B}$.

Démonstration. Étant donné un langage algébrique L et un homomorphisme f, la construction du système d'équations décrivant $f(L)$ consiste à remplacer chaque symbole terminal dans le système d'équation associé à L par son image par f. De cette façon, l'équation décrivant $\pi_{\widehat{B}}(\mathcal{D}_{A\cup B})$ est alors

$$\begin{aligned} S &= \varepsilon + SS + \sum_{a\in A\cup B} \pi_{\widehat{B}}(a) S \pi_{\widehat{B}}(\bar{a}) \\ &= \varepsilon + SS + \sum_{a\in B} aS\bar{a} \end{aligned}$$

dont la plus petite solution est $\mathcal{D}_B$. □

De façon similaire, nous avons aussi le lemme suivant.

Lemme 2.22. *Soient A et B deux alphabets disjoints. Alors, $\pi_{\widehat{B}}(\mathcal{T}_{A\cup B}) = \mathcal{T}_B$.*

2.4 Théorème de Chomsky–Schützenberger

Le théorème de Chomsky–Schützenberger apporte une description de ALG qui sera au cœur de notre étude. Il s'énonce formellement comme suit.

Théorème 2.7 (Théorème de Chomsky–Schützenberger [CS63])**.** *Soit L un langage. L est algébrique si et seulement si il existe un homomorphisme h, un langage régulier (local) R et un entier $k \geq 1$ tels que :*

$$L = h(R \cap \mathcal{D}_k).$$

L'idée du théorème de Chomsky–Schützenberger est que la structure arborescente des dérivations d'une grammaire algébrique peut être décrite par une intersection entre le langage de Dyck et un langage régulier.

Principalité des langages algébriques

Le langage de Dyck sur un nombre arbitraire de parenthèses peut être codé sur deux paires de parenthèses : Pour tout $k \geq 1$, il existe un homomorphisme f tel que $\mathcal{D}_k = f^{-1}(\mathcal{D}_2)$. L'homomorphisme f en question code un alphabet de taille k sur deux symboles :

$$f(a_i) = 01^i0 \text{ et } f(\bar{a}_i) = \overline{f(a_i)} \qquad \forall a_i \in \{a_1, \ldots, a_k\}.$$

On obtient ainsi une forme plus forte du théorème de Chomsky–Schützenberger.

Corollaire 2.23. *Soit L un langage. L est algébrique si, et seulement si, il existe un langage régulier (local) R et deux homomorphismes h et f tels que*

$$L = h(R \cap g^{-1}(\mathcal{D}_2)).$$

Le corollaire ci-dessus implique que $\text{ALG} \subseteq \mathscr{C}(\mathcal{D}_2)$. Puisque $\mathscr{C}(\mathcal{D}_2) \subseteq \text{ALG}$ (le langage de Dyck est algébrique et ALG est une famille agréable), il en découle le corollaire suivant.

Corollaire 2.24. $\text{ALG} = \mathscr{C}(\mathcal{D}_2)$.

Un résultat de Nivat et Boasson particulièrement élégant est le suivant.

Théorème 2.8 (Boasson–Nivat [BN73])**.** *Soit L un langage. Si $\text{ALG} = \mathscr{C}(L)$, alors $\text{ALG} = \mathscr{C}^f(L \cup \{\varepsilon\})$.*

On dit que tout générateur rationnel de ALG est fidèle.

Corollaire 2.25. $\text{ALG} = \mathscr{C}^f(\mathcal{D}_2)$.

Une version « strictement alphabétique » du théorème de Chomsky–Schützenberger a été prouvée par Okhotin comme énoncé ci-dessous.

Théorème 2.9 (Okhotin [Okh12])**.** *Soit L un langage. L est algébrique si et seulement si il existe un langage régulier R, un homomorphisme strictement alphabétique f et un entier k tel que $L = f(R \cap \mathcal{D}_{k,\perp})$.*

Le langage $\mathcal{D}_{k,\perp}$ représente ici le langage de Dyck muni d'un symbole neutre $\perp$: l'ensemble de mots se réduisant à ε par le système $\{a\bar{a} \rightsquigarrow \varepsilon, \perp \rightsquigarrow \varepsilon \mid a \in \{a_1, \ldots, a_k\}\}$.

2.5 Langages indexés

La famille des langages indexés a été introduite sous deux formes distinctes par Aho [Aho67] et Fischer [Fis68] comme une extension de celle des langages algébriques. Celle-ci contient strictement la famille des algébriques et est strictement incluse dans celle des *contextuels*. Le formalisme que nous retenons ici pour leurs présentation est celui des grammaires indexées d'Aho.

Les grammaires indexées se définissent de façon similaire aux grammaires algébriques. La différence est que cette fois, chaque non-terminal présent dans une forme sententielle est muni d'une pile d'*indices* et la dérivation de celui-ci est conditionnée par le sommet de cette pile. Tout au long des dérivations, il est possible de stocker de l'information dans les non-terminaux par le biais de cette pile d'indices et de la copier afin de s'en servir autant de fois que nécessaire.

Définition 2.26 (Grammaires Indexées)**.** *Une grammaire indexée est une structure $\mathcal{I} = (N, T, I, P, S)$ dans laquelle*

- *N est l'ensemble de non-terminaux,*
- *T est l'alphabet terminal,*
- *I est un ensemble fini d'*indices*,*
- *$P \subseteq N^{I \cup \{\varepsilon\}} \times (N^{I^*} \cup T)^*$ est l'ensemble de productions.*
- *$S \in N$ est le non-terminal initial.*

Notations.

- Les éléments de N^{I^*} sont notés X^ω avec $X \in N$ et $\omega \in I^*$;
- les productions (X^α, θ) sont notées $X^\alpha \longrightarrow \theta$;
- les productions de la forme $X^\varepsilon \longrightarrow \theta$ sont notées $X \longrightarrow \theta$;

Dérivations. Les *phrases* (ou *formes sententielles*) sont des mots $\theta \in (N^{I^*} \cup T)^*$, et la relation de dérivation $\underset{\mathcal{I}}{\Longrightarrow}$ se définit comme suit : étant donné deux phrases

$$\begin{aligned} \theta &= \theta_0 X^{x\omega} \theta_1 \\ \text{et } \theta' &= \theta_0 u_1 Y_1^{y_1\omega} u_2 Y_2^{y_2\omega} u_3 \cdots u_n Y_n^{y_n\omega} u_{n+1} \theta_1, \end{aligned}$$

$\theta \underset{\mathcal{I}}{\Longrightarrow} \theta'$ si il existe une production $X^x \longrightarrow u_1 Y_1^{y_1} u_2 Y_2^{y_2} u_3 \cdots u_n Y_n^{y_n} u_{n+1} \in P$. Comme pour les grammaires algébriques, pour toute phrase Θ, $\mathscr{L}(\Theta)$ désigne l'ensemble de mots terminaux dérivables depuis celle-ci :

$$\mathscr{L}(\Theta) = \{u \in T^* \mid \Theta \overset{*}{\underset{\mathcal{I}}{\Longrightarrow}} u\}$$

où $\overset{*}{\underset{\mathcal{I}}{\Longrightarrow}}$ dénote la fermeture réflexive et transitive de $\underset{\mathcal{I}}{\Longrightarrow}$. Le langage généré par la grammaire est $\mathscr{L}(\mathcal{G}) = \mathscr{L}(S)$.

Exemple 2.27. *Soit la grammaire indexée donnée par les productions*

$$\begin{aligned} p_1, p_2 &: S \longrightarrow S^\alpha + aYZ^\beta; & p_3 &: Y^\alpha \longrightarrow a; \\ p_4 &: Z^\beta \longrightarrow b. \end{aligned}$$

Une dérivation possible est alors

$$\begin{aligned} S &\Longrightarrow S^\alpha && (p_1) \\ &\Longrightarrow aY^\alpha Z^{\beta\alpha} && (p_2) \\ &\Longrightarrow aaZ^{\beta\alpha} && (p_3) \\ &\Longrightarrow aab. && (p_4) \end{aligned}$$

Dérivations gauches et droites Une étape de dérivation $\Theta \underset{\mathcal{I}}{\Longrightarrow} \Theta'$ est dite *gauche* et notée $\Theta \underset{g,\mathcal{I}}{\Longrightarrow} \Theta'$ si le non-terminal substitué dans la phrase Θ est celui se trouvant le plus à gauche : Elle est de la forme

$$uX^{\alpha\omega}\Theta_2 \underset{g,\mathcal{I}}{\Longrightarrow} u\Theta_1\Theta_2 \quad \text{avec } u \in T^*, X^{\alpha\omega} \in N^{I^*} \text{ et } X^\alpha \longrightarrow \Theta_1 \in P.$$

De façon similaire, une étape de dérivation est dite *droite* et notée $\Theta \underset{d,\mathcal{I}}{\Longrightarrow} \Theta'$ si le non-terminal substitué est celui se trouvant le plus à droite. Soient $\overset{*}{\underset{g,\mathcal{I}}{\Longrightarrow}}$ et $\overset{*}{\underset{d,\mathcal{I}}{\Longrightarrow}}$ les fermetures réflexives et transitives des relation $\underset{g,\mathcal{I}}{\Longrightarrow}$ et $\underset{d,\mathcal{I}}{\Longrightarrow}$. Pour toute phrase Θ, nous avons :

$$\mathscr{L}(\Theta) \;=\; \{u \in T^* \mid \Theta \overset{*}{\underset{g,\mathcal{I}}{\Longrightarrow}} u\} \;=\; \{u \in T^* \mid \Theta \overset{*}{\underset{d,\mathcal{I}}{\Longrightarrow}} u\}.$$

Notation : Mise en puissance. Étant donné une phrase Θ et une suite d'indices $\omega \in I^*$, nous notons Θ^ω la phrase définie comme suit.

— pour tout symbole terminal $a \in T \cup \{\varepsilon\}$: $a^\omega = a$;
— pour tout $X^{\omega'} \in N^{I^*}$: $(X^{\omega'})^\omega = X^{\omega\omega'}$;

— pour toute paire de phrases $\Theta_1, \Theta_2 \in (N^{I^*} \cup T)^* : (\Theta_1\Theta_2)^\omega = \Theta_1^\omega\Theta_2^\omega$.

Définition 2.28 (Langages indexés)**.** *Un langage L est dit indexé si il existe une grammaire indexée $\mathcal{G}$ telle que $\mathscr{L}(\mathcal{I}) = L$. Nous désignons par* IL *la famille des langages indexés.*

Proposition 2.29. *La famille des langages indexés est une famille agréable de langages.*

Exemple 2.30. *Le langage $L = \{a^n b^n c^n \mid n \geq 0\}$ est indexé. Une grammaire indexée le générant est :*

$$\begin{aligned} p_1 &: S \longrightarrow X^{\$}; & p_2, p_3 &: X \longrightarrow X^{\alpha} + ABC;\\ p_4 &: A^{\alpha} \longrightarrow aA; & p_5 &: A^{\$} \longrightarrow \varepsilon;\\ p_6 &: B^{\alpha} \longrightarrow bB; & p_7 &: B^{\$} \longrightarrow \varepsilon;\\ p_8 &: C^{\alpha} \longrightarrow cC; & p_9 &: C^{\$} \longrightarrow \varepsilon. \end{aligned}$$

Une dérivation pour le mot aabbcc est la suivante :

$$\begin{aligned} S &\Longrightarrow X^{\$} \Longrightarrow X^{\alpha\$} \Longrightarrow X^{\alpha\alpha\$} && (p_1, p_2, p_2)\\ &\Longrightarrow A^{\alpha\alpha\$}B^{\alpha\alpha\$}C^{\alpha\alpha\$} && (p_3)\\ &\Longrightarrow aA^{\alpha\$}B^{\alpha\alpha\$}C^{\alpha\alpha\$} && (p_4)\\ &\Longrightarrow aaA^{\$}B^{\alpha\alpha\$}C^{\alpha\alpha\$} && (p_4)\\ &\Longrightarrow aaB^{\alpha\alpha\$}C^{\alpha\alpha\$} && (p_5)\\ &\overset{*}{\Longrightarrow} aabbC^{\alpha\alpha\$} && (p_6; p_6; p_7)\\ &\overset{*}{\Longrightarrow} aabbcc && (p_8; p_8; p_9) \end{aligned}$$

Formes normales. Nous emploierons les formes normales suivantes.

Théorème 2.10 (Duske–Parchman [PD90])**.** *Pour toute grammaire indexée $\mathcal{I} = (N, T, I, P, S)$, il existe une grammaire indexée équivalente $\mathcal{I}' = (N', T, I, P', S)$ dans laquelle chaque production est sous l'une des formes suivantes :*

$$(1) X \longrightarrow Y^{\alpha}; \quad (2) X^{\alpha} \longrightarrow Y; \quad (3) X \longrightarrow YZ; \quad (4) X \longrightarrow u;$$

avec $X, Y \in N, \alpha \in I, u \in T^*$.

Corollaire 2.31. *Pour toute grammaire indexée $\mathcal{I} = (N, T, I, P, S)$, il existe une grammaire indexée équivalente $\mathcal{I}' = (N', T, I, P', S)$ dans laquelle chaque production est sous l'une des formes suivantes :*

$$(1) X \longrightarrow u\Theta^{\alpha}v; \quad (2) X^{\alpha} \longrightarrow u\Theta v; \qquad \text{avec } u, v \in T^*, \Theta \in N^*, \alpha \in I \cup \{\varepsilon\}.$$

Première partie

Caractérisations des Langages Indexés

Résumé

Dans cette partie, nous étudions la famille des langages algébriques ε-sûrs. Ceux-ci se caractérisent comme des langages générés par des grammaires algébriques pour lesquelles chaque mot dérivable depuis un non-terminal se ρ-réduit au mot vide. Celle-ci est une sous-classe stricte des langages algébriques se ρ-réduisant au mot vide. Nous dérivons de cette famille la classe de transductions algébriques ε-sûres. Celles-ci sont des transductions algébriques pour lesquelles le domaine est ε-sûr. Notre intérêt vis-à-vis de ces objets est que nous en tirons des théorèmes de représentations des langages indexés. En particulier, nous caractérisons la famille des langages indexés comme l'ensemble d'images du langage de Dyck par des transductions algébriques ε-sûres. Ces caractérisations nous offrent de nouvelles approches à d'autres problèmes de caractérisation (des langages indexés). Nous prouvons grâce à ces nouvelles présentations des caractérisations logiques des langages indexés.

Plan et Contenu de la partie

Cette partie de contributions est divisée en trois chapitres.

— Dans le chapitre 3, nous étudions la famille des langages et transductions algébriques ε-sûrs : syntaxe des grammaires, propriétés de clôture, théorèmes de représentations, problèmes indécidables.

— Dans le chapitre 4, nous présentons plusieurs théorèmes de représentation des langages indexés. Nous y prouvons en plusieurs variantes les caractérisations des langages indexés comme images du langages de Dyck par transductions algébriques ε-sûres. De ces caractérisations et des théorèmes de représentation des transductions algébriques ε-sûres, nous prouvons la principalité du cône qu'est la famille des langages indexés par une généralisation du langage de Dyck.

— Finalement, dans le chapitre 5, nous étudions la définissabilité des langages indexés dans des fragments sémantique du second ordre existentiel. Nous nous intéressons aussi aux liens entre certains fragments sémantiques du second ordre existentiel et cônes (croissants) principaux.

Chapitre 3

Langages algébriques ε-sûrs

3.1 Résumé

McNaughton [McN67] et Knuth [Knu67] ont défini dans les années 60 la notion de grammaires à parenthèses. Celles-ci sont des grammaires algébriques dans lesquelles chaque production est de la forme $X \longrightarrow (\Theta)$, où Θ est une phrase ne contenant pas les parenthèses $(,)$. Cette restriction octroient à ces grammaires de meilleurs propriétés. En particulier, le problème de l'équivalence est décidable pour les langages générés par ces grammaires [Knu67]. Il existe depuis lors, plusieurs classes de grammaires généralisant ce concept (voir [AM04, GH67, DPS79]). Des plus récentes, nous pouvons citer les grammaires purement *équilibrées* de Berstel et Boasson [BB02]. Celles-ci ont comme alphabet terminal une union disjointe d'alphabets $A \cup \overline{A}$ et leurs productions sont de la forme $X \longrightarrow a\Theta\bar{a}$ avec $a \in A$ et Θ est une expression régulière sur l'alphabet des non-terminaux. Une classe restreinte de celles-ci, appelée *grammaires XML*, est définie et étudiée dans [BB00] par les mêmes auteurs.

Nous étudions ici une classe de grammaires algébriques pour lesquelles l'alphabet terminal est une union disjointe $A \cup \overline{A}$, mais cette fois chaque production est de la forme $X \longrightarrow \omega\Theta\overline{\omega}$ avec $\omega \in \widehat{A}^*$ et Θ est une suite de non-terminaux. Nous appelons ces grammaires des grammaires algébriques ε-sûres. Le choix de ce nom vient du fait que cette restriction des productions est équivalente à ce que le langage généré par chaque non-terminal se ρ-réduit au mot vide. Dans ce chapitre, nous étudions ces grammaires et langages : propriétés des grammaires, propriétés de clôture, problèmes indécidables et théorème de représentation. Nous dérivons de ces objets une classe de transductions que nous appelons transductions algébriques ε-sûres. Nous prouvons plusieurs théorèmes de représentation pour cette classe de transductions ; ce qui nous permet d'obtenir des caractérisations homomorphiques pour toute famille de langages qui peut être générée par ces transformations.

Des grammaires similaires sont introduites par Duske,Parchmann et Specht dans [DPS79], grâce auxquelles les auteurs proposent une caractérisation homomorphique des langages indexés (de ce fait aussi similaire à l'une de celles que nous présenterons). Nous menons toutefois, dans ce chapitre et dans cette partie une étude plus en profondeur de ces langages et de ces caractérisations.

3.2 Langages ε-sûrs

Dans cette section, nous présentons les langages et grammaires algébriques ε-sûrs. Nous introduisons les grammaires algébriques ε-sûres, nous étudions leurs propriétés, puis nous étu-

dions les propriétés de clôture de ces langages et nous terminons la section par un théorème de représentation de ces langages.

Considérons premièrement la classe d'homomorphismes définie ci-dessus. Celle-ci jouera un rôle important tout au long de ce chapitre.

3.2.1 Homomorphismes ε-sûrs

Définition 3.1 (homomorphismes ε-sûrs)**.** *Un homomorphisme $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ est dit*
- symétrique *si* $\forall a \in \widehat{A} : f(\bar{a}) = \overline{f(a)}$;
- ε-sûr *si* $\forall u \in \mathcal{T}_A : f(u) \in \mathcal{T}_B$;

Remarque 3.2.
1. *Tout homomorphisme symétrique est ε-sûr ;*
2. *chacune de ces propriétés est préservée par composition ;*

Lemme 3.3. *La propriété « ε-sûr » est équivalente à « $\forall a \in \widehat{A} : f(a\bar{a}) \in \mathcal{T}_B$ ».*

Démonstration. En effet, si un homomorphisme $f : \widehat{A} \longrightarrow \widehat{B}^*$ est ε-sûr, alors en particulier pour tout $a \in \widehat{A} : f(a\bar{a}) \in \mathcal{T}_B$ puisque $a\bar{a} \in \mathcal{T}_A$. Réciproquement, si $f(a\bar{a}) \in \mathcal{T}_B$ pour tout $a \in \widehat{A}$, alors pour tout mot $u \in \mathcal{T}_A$, $f(u) \in \mathcal{T}_B$ puisque tout facteur réductible $a\bar{a} \in \mathcal{T}_A$ est substitué par $f(a)f(\bar{a}) \in \mathcal{T}_B$. □

Exemple 3.4. *L'homomorphisme $f : \widehat{\{a,b\}}^* \longrightarrow \widehat{\{c,d\}}^*$ tel que*

$$f(a) = cd; \quad f(\bar{a}) = \bar{d}\bar{c}; \quad f(b) = cc; \quad ; f(\bar{b}) = \bar{c}\bar{d}d\bar{c}$$

est ε-sûr puisque pour tout $a \in \widehat{A} : f(a\bar{a}) \in \mathcal{T}_B$.

3.2.2 Grammaires algébriques ε-sûres

Nous introduisons maintenant les grammaires et langages algébriques ε-sûrs.

Définition 3.5 (Grammaires algébriques ε-sûres)**.** *Une grammaire algébrique ε-sûre est une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ avec $T = \widehat{A}$ pour un certain alphabet A et dans laquelle toute production est de la forme*

$$X \longrightarrow \omega\Theta\overline{\omega} \qquad \text{avec } \omega \in \widehat{A}^*, \Theta \in N^*.$$

Un langage algébrique est dit ε-sûr si il peut être généré par une grammaire algébrique ε-sûre. Nous désignons par ε-ALG *la famille de langages algébriques ε-sûrs.*

Exemple 3.6. *Le langage de Dyck (resp. Dyck bilatéral) est bien évidemment ε-sûr car il est généré par la grammaire*

$$S \longrightarrow \varepsilon + \sum_{a \in A} aS\bar{a} + SS \quad (\text{resp. } S \longrightarrow \varepsilon + \sum_{a \in \widehat{A}} aS\bar{a} + SS).$$

De par la forme des productions des grammaires algébriques ε-sûres, il est déjà facile d'observer que dans une grammaire algébrique ε-sûre $\mathcal{G} = (N, T = \widehat{A}, P, S)$, pour tout non-terminal $X \in N : \mathscr{L}(X) \subseteq \mathcal{T}_A$. Nous montrons dans ce qui suit que cette propriété est équivalente à dire qu'une grammaire algébrique génère un langage ε-sûr.

Proposition 3.7. *Soit $\mathcal{G} = (N, \widehat{A}, P, S)$ une grammaire algébrique. Si pour toute production $X \longrightarrow \Theta \in P$: nous avons $\pi_{\widehat{A}}(\Theta) \in \mathcal{T}_A$, alors $\mathscr{L}(\mathcal{G})$ est ε-sûr.*

Démonstration. Il s'agit ici de montrer que le membre droit de toute production dans une telle grammaire peut être décomposé de façon à obtenir des productions de la forme

$$X \longrightarrow \omega\Theta\overline{\omega}, \qquad \text{avec } \omega \in \widehat{A}^*, \Theta \in N^*. \tag{3.1}$$

Considérons une production $p : X \longrightarrow \Theta \in P$ telle que le mot $v = \pi_{\widehat{A}}(\Theta) \in \mathcal{T}_A$. Puisque $v \in \mathcal{T}_A$, il peut être factorisé de trois façons différentes : soit $v = \varepsilon$; soit $v = v_1v_2$ avec $v_1, v_2 \neq \varepsilon$ et $v_1, v_2 \in \mathcal{T}_A$; soit $v = a\omega\bar{a}$ avec $a \in \widehat{A}$ et $\omega \in \mathcal{T}_A$. La production p peut être décomposée selon ces trois cas :

— Cas 1 : $v = \varepsilon$. Laisser la production p telle quelle, elle est de la forme (3.1).

— Cas 2 : $v = v_1v_2$ avec $v_1, v_2 \neq \varepsilon$ et $v_1, v_2 \in \mathcal{T}_A$.
Remplacer la production p par $X \longrightarrow Y_pZ_p$ et ajouter deux quelconques productions $Y_p \longrightarrow \Theta_1$ et $Y_p \longrightarrow \Theta_2$ telles que

$$\Theta_1\Theta_2 = \Theta; \qquad \pi_{\widehat{A}}(\Theta_1) = v_1 \quad \text{et} \quad \pi_{\widehat{A}}(\Theta_2) = v_2.$$

— Cas 3 : $v = a\omega\bar{a}$ avec $a \in \widehat{A}, \omega \in \mathcal{T}_A$. Alors il existe une unique décomposition

$$\Theta = \Theta_1 a \Theta_2 \bar{a} \Theta_3; \qquad \pi_{\widehat{A}}(\Theta_2) = \omega; \quad \Theta_1, \Theta_3 \in N^*.$$

Il nous suffit de remplacer p par $X \longrightarrow X_pY_pZ_p$ et d'ajouter les productions

$$X_p \longrightarrow \Theta_1; \quad Y_p \longrightarrow aY'_p\bar{a}; \quad Y'_p \longrightarrow \Theta_2; \quad \text{et} \quad Z_p \longrightarrow \Theta_3.$$

La construction de la grammaire ε-sûre consiste à itérer ce processus jusqu'à ce que toute les productions soient de la forme (3.1). Cette procédure termine et est valide parce que toutes les productions obtenues lors d'une décomposition sont soit de la forme (3.1) ; soit de la forme $X \longrightarrow \Theta$ avec $\pi_{\widehat{A}}(\Theta) \in \mathcal{T}_A$ avec strictement moins de symboles terminaux que la production décomposée. Puisqu'une production n'ayant pas de symbole terminal est de (3.1), toutes les productions obtenues à la fin de la procédure sont de la forme (3.1). □

Corollaire 3.8. *Un langage algébrique L est ε-sûr si, et seulement si, il peut être généré par une grammaire algébrique* $(N, T = \widehat{A}, P, S)$ *dans laquelle pour tout production* $X \longrightarrow \Theta : \pi_T(\Theta) \in \mathcal{T}_A$.

Démonstration. En effet, par définition des grammaires algébriques ε-sûres, pour toute production $X \longrightarrow \Theta$ d'une grammaire algébrique ε-sûr, nous avons $\pi_T(\Theta) \in \mathcal{T}$. La proposition 3.7 énonce la réciproque. □

Exemple 3.9. *Le langage généré par la grammaire ci-dessous est ε-sûr puisque la suite de terminaux apparaissant dans chaque production appartient à* $\mathcal{T}$.

$$S \longrightarrow aY\bar{a}Z + ab\bar{b}\bar{a}; \qquad Y \longrightarrow b\bar{b};$$
$$Z \longrightarrow YS;$$

Proposition 3.10. *Soit* $\mathcal{G} = (N, T = \widehat{A}, P, S)$ *une grammaire algébrique. Les propriétés suivantes sont équivalentes.*

1. *Pour toute production* $X \longrightarrow \Theta : \pi_T(\Theta) \in \mathcal{T}_A$.
2. *Pour tout non-terminal* $X \in N : \mathscr{L}(X) \subseteq \mathcal{T}_A$.

Démonstration. Le sens $(1 \Rightarrow 2)$ est trivialement vrai. En effet, puisque pour toute productions $X \longrightarrow \Theta$, nous avons $\pi_T(\Theta) \in \mathcal{T}_A$, alors pour toute dérivation $X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} \Theta$, nous avons aussi $\pi_T(\Theta) \in \mathcal{T}$. Le sens $(2 \Rightarrow 1)$ se montre comme suit. Supposons l'hypothèse (2). Soit une production $X \longrightarrow \omega_1 Y_1 \omega_2 \cdots Y_n \omega_{n+1}$. Puisque $\omega_1 \mathscr{L}(Y_1)\omega_2 \cdots \mathscr{L}(Y_n)\omega_{n+1} \subseteq \mathscr{L}(X)$ et que $\rho(\mathscr{L}(X)) = \{\varepsilon\}$, on a

$$\rho(\omega_1 \mathscr{L}(Y_1)\omega_2 \cdots \mathscr{L}(Y_n)\omega_{n+1}) = \rho(\omega_1 \rho(\mathscr{L}(Y_1))\omega_2 \cdots \rho(\mathscr{L}(Y_n))\omega_{n+1}) = \{\varepsilon\}.$$

Par hypothèse, $\rho(\mathscr{L}(Y_i)) = \{\varepsilon\}$ pour tout i, et par conséquent $\rho(\omega_1\omega_2 \cdots \omega_{n+1}) = \varepsilon$. □

Corollaire 3.11. *Un langage algébrique L est ε-sûr si, et seulement si, il peut être généré par une grammaire algébrique telle que pour tout non-terminal $X : \mathscr{L}(X) \subseteq \mathcal{T}$.*

Lemme 3.12. *Un langage algébrique $L \subseteq \widehat{A}^*$ est ε-sûr si, et seulement si, il peut être généré par une grammaire algébrique (N, T, P, S) avec $T = \widehat{A}$ et dans laquelle toute production est sous l'une des formes suivantes*

$$X \longrightarrow YZ; \quad X \longrightarrow \alpha Y \overline{\alpha}, \alpha \in T; \quad X \longrightarrow \omega, \omega \in \mathcal{T}_A.$$

Démonstration. D'après le corollaire 3.11, le langage généré par toute grammaire sous cette forme est ε-sûr. Réciproquement, tout langage algébrique ε-sûr peut par définition être généré par une grammaire dans laquelle toute production est de la forme $X \longrightarrow \omega\Theta\overline{\omega}$ avec $\omega \in T^*$ et $\Theta \in N^*$. Nous pouvons par substitutions mettre cette grammaire sous la forme désirée : toute production $p : X \longrightarrow \omega\Theta\overline{\omega}$ sera remplacée par $X \longrightarrow \omega Y_0 \overline{\omega}$, en introduisant une suite de productions quadratiques permettant de générer Θ depuis Y_0. Les productions $X \longrightarrow \omega Y \overline{\omega}$ avec $\omega = \alpha_1 \cdots \alpha_{|\omega|}$ seront elles même remplacées par des suites de productions linéaires $X_i \longrightarrow \alpha_i X_{i+1} \bar{\alpha}_i$ avec $i \in \{0, \ldots, |\omega| - 1\}$, $X_0 = X$ et $X_{|\omega|} = Y$. □

3.2.3 Propriétés de clôture de ε-ALG

Nous prouvons maintenant des propriétés de clôture de ε-ALG. Nous montrons en particulier que cette famille est fermée par union, concaténation, étoile de Kleene, homomorphismes ε-sûrs, intersection avec des réguliers, substitutions et quotient par des mots de $\mathcal{T}$.

Proposition 3.13. *La famille ε-ALG est fermée par union, concaténation et étoile de Kleene.*

Démonstration. La construction de l'union, concaténation et étoile de Kleene par des grammaires algébriques préservent la propriété « $\mathscr{L}(X) \subseteq \mathcal{T}$, pour tout non-terminal X » : soient $\mathcal{G}_1 = (N_1, T, P_1, S_1)$ et $\mathcal{G}_2 = (N_2, T, P_2, S_2)$ deux grammaires algébriques ε-sûres. Nous pouvons supposer sans perte de généralité que N_1 et N_2 sont disjoints ;

- le langage $\mathscr{L}(\mathcal{G}_1) \cup \mathscr{L}(\mathcal{G}_2)$ est généré par la grammaire $\mathcal{G}_\cup = (N_1 \cup N_2 \cup \{S\}, T, P, S)$ avec $P = P_1 \cup P_2 \cup \{S \longrightarrow S_1 + S_2\}$;
- le langage $\mathscr{L}(\mathcal{G}_1)\mathscr{L}(\mathcal{G}_2)$ est généré par la grammaire $\mathcal{G}_\cdot = (N_1 \cup N_2 \cup \{S\}, T, P, S)$ avec $P = P_1 \cup P_2 \cup \{S \longrightarrow S_1 S_2\}$;
- le langage $\mathscr{L}(\mathcal{G}_1)^*$ est quant à lui généré par la grammaire $\mathcal{G}_* = (N \cup \{S\}, T, P', S)$ avec $P' = P \cup \{S \longrightarrow \varepsilon + S_1 S\}$.

Chacune de ces trois constructions introduit une unique nouvelle variable dont le langage généré est inclus dans $\mathcal{T}$ puisque celui-ci est soit l'union, soit la concaténation, soit l'étoile de Kleene de langages inclus dans $\mathcal{T}$. D'après le corollaire 3.11, les grammaires construites sont donc ε-sûres. □

Proposition 3.14. *La famille ε-ALG est fermée par homomorphismes ε-sûrs.*

Démonstration. Étant donnée une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ et un homomorphisme f, la construction d'une grammaire $\mathcal{G}' = (N, T, P', S)$ telle que $\mathscr{L}(\mathcal{G}') = f(\mathscr{L}(\mathcal{G}))$ consiste à remplacer, tout symbole terminal a par $f(a)$ dans toute production. Pour tout non-terminal X, le langage généré par celui-ci dans la grammaire $\mathcal{G}'$, notons le $\mathscr{L}_{\mathcal{G}'}(X)$, sera alors $\mathscr{L}_{\mathcal{G}'}(X) = f(\mathscr{L}_{\mathcal{G}}(X))$. Du fait que f soit ε-sûr, si $\mathscr{L}_{\mathcal{G}}(X)$ est inclus dans $\mathcal{T}$, alors $\mathscr{L}_{\mathcal{G}'}(X)$ aussi. Par conséquent d'après le corollaire 3.11, si la grammaire $\mathcal{G}$ est ε-sûre, alors $\mathcal{G}'$ aussi. □

Lemme 3.15. *La famille ε-ALG est fermée par substitution.*

Démonstration. Soit $L \in$ ε-ALG et $\mathcal{G} = (N, T, P, S)$ une grammaire algébrique ε-sûre générant L avec $T = \widehat{A}$ pour un certain alphabet A. Soit $\nu : \widehat{A} \longrightarrow 2^{\widehat{B}^*}$ une substitution telle que pour tout $a \in \widehat{A} : \nu(a) \in$ ε-ALG. Pour tout $a \in \widehat{A}$, soit alors $\mathcal{G}_a = (N_a, T', P_a, S_a)$ une grammaire algébrique ε-sûre générant $\nu(a)$ avec $T' = \widehat{B}$. La grammaire algébrique ε-sûre $\mathcal{G}' = (N_\nu, T', P_\eta, S)$ telle que $\mathscr{L}(\mathcal{G}') = \nu(\mathscr{L}(\mathcal{G}))$ est définie comme suit :

- $N_\nu = N \cup \bigcup_{a \in A} N_a$;
- $P_\nu = \bigcup_{a \in A} P_a \cup P'$ et P' est obtenu en remplaçant chaque symbole terminal a dans une production de P par le non-terminal S_a.

Le langage généré est clairement $\nu(L)$ puisque par définition, chaque non-terminal S_a génère $\nu(a)$ et que P' est obtenu en remplaçant chaque terminal a par le non-terminal S_a. De plus, celui-ci est ε-sûr d'après le corollaire 3.11 puisque le langage associé à chaque non-terminal est inclus dans $\mathcal{T}$. □

Lemme 3.16. *Soit $L \subseteq A^*$ un langage algébrique et $\nu : A^* \longrightarrow \widehat{2}^{B^*}$ une substitution telle que pour tout $a \in A : \nu(a) \subseteq \mathcal{T}$. Le langage $\nu(L)$ est ε-sûr.*

Démonstration. Le langage généré par la grammaire obtenue en appliquant la construction de la preuve du lemme 3.15 est ε-sûre même si la grammaire initiale ne l'est pas. Le langage associé à chaque non-terminal dans la grammaire obtenue est inclus dans $\mathcal{T}$, ce qui d'après le corollaire 3.11 implique que le langage $\nu(L)$ est ε-sûr. □

Proposition 3.17. *La famille ε-ALG est fermée par intersection avec des langages réguliers.*

Démonstration. Soit $\mathcal{G} = (N, T = \widehat{A}, P, S)$ une grammaire algébrique ε-sûre et $R \subseteq \widehat{A}^*$ un langage régulier. D'après le théorème 1.2, il existe un monoïde fini M, un ensemble $M_0 \subseteq M$ et un morphisme de monoïde $\mu : \widehat{A}^* \longrightarrow M$ tel que $R = \mu^{-1}(M_0)$. Le langage $\mathscr{L}(\mathcal{G}) \cap R$ se définit alors comme

$$\mathscr{L}(\mathcal{G}) \cap R = \{\omega \in \widehat{A}^* \mid S \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} \omega \text{ et } \mu(\omega) \in M_0\}.$$

Nous construisons une grammaire $\mathcal{G}'$ générant $\mathscr{L}(\mathcal{G}) \cap R$ en étiquetant les non-terminaux $X \in N$ par les éléments $m \in M$ de sorte que pour tout non-terminal $X \in N$, $m \in M$ et $\omega \in T^*$:

$$\omega \in \mathscr{L}_{\mathcal{G}'}(X_m) \quad \Leftrightarrow \quad \omega \in \mathscr{L}_{\mathcal{G}}(X) \text{ et } \mu(\omega) = m.$$

Le langage $\mathscr{L}(\mathcal{G}) \cap R$ sera alors l'union finie

$$\mathscr{L}(\mathcal{G}) \cap R = \bigcup_{m \in M_0} \mathscr{L}_{G'}(S_m).$$

La grammaire $\mathcal{G}' = (N', T, P', S')$ désirée se définit comme suit : l'ensemble de non-terminaux est $N' = \{X_m \mid m \in M\}$ et P' est composé des productions

$$\begin{aligned} S' &\longrightarrow S_m \text{ telles que } m \in M_0, \\ X_m \longrightarrow \omega X_{1,m_1} \cdots X_{n,m_n} \overline{\omega} &\text{ telles que } X \longrightarrow \omega X_1 \cdots X_n \overline{\omega} \in P \\ &\text{et } \mu(\omega) m_1 \cdots m_n \mu(\overline{\omega}) = m \end{aligned}$$

Nous pouvons affirmer que pour tout $\omega \in T^*, X \in N, m \in M_0$:

$$X_m \underset{\mathcal{G}'}{\overset{*}{\Longrightarrow}} \omega \text{ ssi } X \underset{\mathcal{G}}{\overset{*}{\Longrightarrow}} \omega \text{ et } \mu(\omega) = m.$$

Par construction, cette propriété est vraie pour toute production terminale $X_m \longrightarrow \omega$, avec $\omega \in T^*$. Inductivement, si on la considère vraie pour toute dérivation terminale de taille au plus n, elle l'est aussi pour toute dérivation terminale $X_m \Rightarrow \omega X_{1,m_1} \cdots X_{n,m_n} \overline{\omega} \overset{*}{\Longrightarrow} \omega'$ telle que la dérivation de chaque X_{i,m_i} est de taille au plus n. De plus, par construction, nous avons

$$\mathscr{L}(\mathcal{G}') = \mathscr{L}(S') = \bigcup_{m \in M_0} \mathscr{L}(S_m).$$

Ceci conclut la preuve de la proposition ci-dessus. □

Puisque tout langage régulier R inclus dans $\mathcal{T}$ peut s'exprimer comme $R \cap \mathcal{T}$, la proposition 3.17 a pour corollaire le suivant :

Corollaire 3.18. *Tout langage régulier inclus dans $\mathcal{T}$ est ε-sûr.*

Proposition 3.19. *Soit $L \subseteq \widehat{A}^*$ un langage algébrique ε-sûr, et $\omega \in \mathcal{T}_A$. Les langages $\omega^{-1} \cdot L$ et $L \cdot \omega^{-1}$ sont algébriques ε-sûrs.*

Démonstration. Nous montrons que le langage $\omega^{-1} \cdot L$ est ε-sûr, la preuve pour $L \cdot \omega^{-1}$ est complètement symétrique.

Soit $\mathcal{G} = (N, T, P, S)$ une grammaire algébrique ε-sûre générant L avec $T = \widehat{A}$, et R_ω l'ensemble de phrases $\Theta \in (N \cup T)^*$ telles qu'il existe une dérivation gauche $S \underset{g,\mathcal{G}}{\overset{*}{\Longrightarrow}} \Theta_0 \underset{g,\mathcal{G}}{\Longrightarrow} \omega\Theta$ et telles que ω n'est pas préfixe de Θ_0. Le langage $\omega^{-1} \cdot L$ se définit alors comme

$$\omega^{-1} \cdot L = \bigcup_{\Theta \in R_\omega} \mathscr{L}_{\mathcal{G}}(\Theta).$$

Remarque 3.20.
1. *Puisque $\mathcal{G}$ est ε-sûre et que $\omega \in \mathcal{T}_A$, pour tout $\Theta \in R_\omega$: nous avons $\pi_T(\Theta) \in \mathcal{T}_A$ (Proposition 3.7).*
2. *D'après le théorème 2.2, le langage R_ω est un langage régulier sur l'alphabet $N \cup T$.*

Considérons l'homomorphisme $f : (N \cup T)^* \longrightarrow (\widehat{N} \cup T)^*$ tel que $f(\alpha) = \alpha$ pour $\alpha \in T$ et $f(\alpha) = \alpha\bar{\alpha}$ pour $\alpha \in N$. Le langage $f(R_\omega)$ est alors inclus dans $\mathcal{T}_{A \cup N}$ et d'après le corollaire 3.18, il est ε-sûr. Celui-ci peut donc être généré par une grammaire algébrique ε-sûre. Nous pouvons émettre l'affirmation suivante.

Affirmation 1. $\omega^{-1} \cdot L = \nu(f(R_\omega))$ *avec* $\nu : (\widehat{N} \cup T)^* \longrightarrow 2^{T^*}$ *tel que*

$$\forall a \in T : \nu(a) = \{a\}; \quad \forall X \in N : \nu(X) = \mathscr{L}_{\mathcal{G}}(X) \text{ et } \nu(\overline{X}) = \{\varepsilon\}.$$

Soit alors $\mathcal{G}_\omega = (N_\omega, T \cup \widehat{N}, P_\omega, S_\omega)$ une grammaire ε-sûre générant le langage $f(R_\omega)$. La grammaire générant $\omega^{-1} \cdot L$ est alors $\mathcal{G}' = (\widehat{N} \cup N_\omega, T, P', S_\omega)$ avec

$$P' = P \cup P_\omega \cup \{\overline{X} \longrightarrow \varepsilon \mid X \in N\}.$$

La grammaire $\mathcal{G}'$ est de plus ε-sûre. En effet, puisque $\mathcal{G}$ et $\mathcal{G}_\omega$ sont ε-sûres, pour tout $X \longrightarrow \Theta \in P' : \pi_T(\Theta) \in \mathcal{T}_A$. Ceci conclut la preuve du lemme ci-dessus. □

Exemple 3.21. *Soit la grammaire*

$$\begin{aligned} &S \longrightarrow SX + YX; && Y \longrightarrow aY\bar{a} + \bar{a}Xa;\\ &X \longrightarrow b\bar{b}. \end{aligned}$$

Prenons ici $\omega = a\bar{a}$. Les dérivations gauches produisant $a\bar{a}$ sont de la forme

$$S \underset{\mathcal{G}'}{\overset{*}{\Longrightarrow}} YX^n \Rightarrow aY\bar{a}X^n \Rightarrow a\bar{a}Xa\bar{a}X^n \quad \text{avec } n \geq 1.$$

Le langage $R_\omega \subseteq (N \cup T)^*$ est alors $Xa\bar{a}X^+$ et le langage $\omega^{-1} \cdot L$ est

$$\omega^{-1} \cdot L \;=\; \mathscr{L}(X)a\bar{a}\mathscr{L}(X)^+ \;=\; b\bar{b}a\bar{a}(b\bar{b})^+.$$

Une grammaire $\mathcal{G}_\omega = (N_\omega, \widehat{N} \cup T, P_\omega, S_\omega)$ générant $f(R_\omega) = X\overline{X}a\bar{a}(X\overline{X})^+$ est

$$\begin{aligned} &S_\omega \longrightarrow Z_1Z_2Z_3; && Z_1 \longrightarrow X\overline{X};\\ &Z_2 \longrightarrow a\bar{a}; && Z_3 \longrightarrow X\overline{X} + X\overline{X}Z_3. \end{aligned}$$

La grammaire $\mathcal{G}' = (\widehat{N} \cup N_\omega, T, P', S_\omega)$ générant $\omega^{-1} \cdot L$ est alors

$$\begin{aligned} &S_\omega \longrightarrow Z_1Z_2Z_3; && Z_1 \longrightarrow X\overline{X};\\ &Z_2 \longrightarrow a\bar{a}; && Z_3 \longrightarrow X\overline{X} + X\overline{X}Z_3\\ &S \longrightarrow SX + YX; && Y \longrightarrow aY\bar{a} + \bar{a}Xa;\\ &X \longrightarrow b\bar{b}; && \overline{X} \longrightarrow \varepsilon. \end{aligned}$$

Proposition 3.22. *La famille ε-ALG n'est pas fermée par intersection.*

Démonstration. Soient $L_1, L_2 \subseteq A^*$ deux langages algébriques tels que $L = L_1 \cap L_2$ n'est pas algébrique. De tels langages existent puisque ALG n'est pas fermée par intersection (voir [ABB97]). Considérons aussi $f : A^* \longrightarrow \widehat{A}^*$ tel que $f(a) = a\bar{a}$ pour tout $a \in A$. L'homomorphisme f est injectif et la composition $\pi_A \circ f$ est l'identité. Le langage L peut alors s'écrire comme

$$\begin{aligned} L &= \pi_A \circ f(L_1 \cap L_2)\\ &= \pi_A(f(L_1) \cap f(L_2)) \quad \text{(par injectivité de } f\text{)} \end{aligned}$$

D'après le lemme 3.16, les langages $f(L_1)$ et $f(L_2)$ sont chacun algébriques et ε-sûrs puisque pour tout $a \in A : f(a) \in \mathcal{T}$; pourtant le langage $f(L_1) \cap f(L_2)$ n'est pas algébrique car autrement L le serait. La famille ε-ALG n'est donc pas fermée par intersection. □

Corollaire 3.23. *La famille ε-ALG n'est pas fermée par complément (dans $\mathcal{T}$).*

Démonstration. La famille ε-ALG est fermée par union ; si elle était fermée par complément, elle le serait par intersection. □

3.2.4 Théorème de représentation de ε-ALG

Nous terminons cette présentation de ε-ALG par une caractérisation homomorphique. Nous montrons qu'un langage est algébrique ε-sûr si et seulement si il peut s'exprimer comme $f(R \cap \mathcal{D}_k)$ où R est un langage régulier, f un homomorphisme ε-sûr (symmétrique) et k un entier. Les différentes preuves du théorème de Chomsky–Schützenberger trouvées dans la littératures [ABB97, CS63, Okh12] consistent à décrire l'ensemble de dérivations d'une grammaire comme un langage sur l'alphabet de ses productions. C'est ce dernier qui correspond à $R \cap \mathcal{D}_k$. L'homomorphisme f associe à chacune de ces productions le mot généré. En particulier, dans notre cas, cet homomorphisme pourra être symétrique du fait de la forme des productions des grammaires algébriques ε-sûres.

Théorème 3.1. *Soit L un langage. Le langage $L \in$ ε-ALG si et seulement si il existe un homomorphisme ε-sûr (symétrique) f, un langage régulier (une union finie de langages locaux) R et un entier k tels que $L = f(R \cap \mathcal{D}_k)$.*

La preuve qui suit est une adaptation de la version non-effaçante du théorème de Chomsky–Schützenberger prouvée dans [Okh12]. Nous ne présentons que la construction.

Démonstration. Le sens ($\Leftarrow$) découle des propriétés de clôture de ε-ALG. Il nous suffit de montrer le sens ($\Rightarrow$).

Soit $\mathcal{G} = (N, \widehat{A}, P, S)$ une grammaire algébrique ε-sûre générant L et dans laquelle toute production est de la forme

$$X \longrightarrow \omega\Theta\overline{\omega} \text{ avec } \omega \in \widehat{A}^*, \Theta \in N^*.$$

Nous construisons à partir de $\mathcal{G}$ la grammaire algébrique ε-sûre $\mathcal{G}' = (N', \widehat{\Sigma}, P', S')$ définie comme suit :

- l'ensemble de non-terminaux est $N' = \{X_p \mid X \in N, p \in P\}$,
- l'alphabet $\Sigma \subseteq (P \cup \{\$\}) \times P$ est l'ensemble des $(\$, p)$ tels que le membre gauche de p est S et des (p, p') tels que p est de la forme $p = X \longrightarrow \Theta_1 Y \Theta_2$ et le membre gauche de p' est Y ($(p, p') \in \Sigma$ *si la production p peut être « suivie » par la production p'*) ;
- l'ensemble P' est constitué de toutes les productions $X_{p_0} \longrightarrow (p_0, p_1) X_{1,p_1} \cdots X_{n,p_1} \overline{(p_0, p_1)}$ pour tout $(p_0, p_1) \in \Sigma$ et $p_1 = X \longrightarrow \omega X_1 \cdots X_n \overline{\omega} \in P$.
- le symbole initial est $S_\$$.

Soit alors l'homomorphisme symétrique $h : \widehat{\Sigma}^* \longrightarrow \widehat{A}^*$ tel que pour tout $(p, p') \in \Sigma$, $h((p, p')) = \omega$ et $h(\overline{(p, p')}) = \overline{\omega}$ si p' est de la forme

$$p' = X \longrightarrow \omega\Theta\overline{\omega} \text{ avec } \omega \in \widehat{A}^*, \Theta \in N^*.$$

Il en va alors de soi que $\mathscr{L}(\mathcal{G}) = h(\mathscr{L}(\mathcal{G}'))$.

Le langage $\mathscr{L}(\mathcal{G}')$ peut s'exprimer comme $R \cap \mathcal{D}_\Sigma$ [Okh12] où R est une union finie de langages locaux défini comme suit : Soit R_0 l'ensemble de mots de $\widehat{\Sigma}^*$ tels que tout facteur de taille 2 est de l'une des formes suivantes

- $(p_1, p_2)(p_2, p_3)$ où p_2 est de la forme $X \longrightarrow \omega Y_1 \cdots Y_n \overline{\omega}$ et le membre gauche de p_3 est Y_1 ;
- $\overline{(p_1, p_2)}(p_1, p_3)$ où p_1 est de la forme $X \longrightarrow \omega Y_1 \cdots Y_n \overline{\omega}$, le membre gauche p_2 est Y_i, pour un certain i, et celui de p_3 est Y_{i+1} ;
- $\overline{(p_1, p_2)}\,\overline{(p_3, p_1)}$ où p_1 est de la forme $X \longrightarrow \omega Y_1 \cdots Y_n \overline{\omega}$ et le membre gauche de p_2 est Y_n.

— $(p, p')\overline{(p, p')}$ si la production p' est terminale.
Le langage R est alors

$$R = R_0 \cap \bigcup_{p=S \longrightarrow \Theta \in P} (\$, p)\widehat{\Sigma}^* \overline{(\$, p)}.$$

□

Corollaire 3.24. *ε-ALG est la plus petite famille de langages fermée par homomorphismes ε-sûrs, intersection avec des langages réguliers, union, concaténation, étoile de Kleene, et qui contient le langage de Dyck.*

3.3 Comparaison avec l'inclusion dans le langage de Dyck bilatéral

Il est naturel de se demander si tout langage algébrique inclus dans $\mathcal{T}$ est ε-sûr. Dans cette section, nous montrons que la famille ε-ALG est strictement incluse dans la famille des langages algébriques inclus dans $\mathcal{T}$ et (de ce même fait) qu'il est indécidable de savoir si un langage algébrique inclus dans $\mathcal{T}$ est ε-sûr.

Par besoin de nomenclature, nous appelons $\mathcal{T}$-ALG la famille de langages algébriques inclus dans $\mathcal{T}$.

Définition 3.25. *Soit $\mathcal{T}$-ALG la famille de langages algébriques inclus dans $\mathcal{T}$.*

$$\mathcal{T}\text{-ALG} = \{L \in \text{ALG} \mid L \subseteq \mathcal{T}\}.$$

Proposition 3.26. *$\mathcal{T}$-ALG est fermée par union, concaténation, étoile de Kleene, homomorphismes ε-sûrs, intersection avec des réguliers et quotient (gauche et droit) par des mots de $\mathcal{T}$.*

Démonstration. La famille des langages algébriques est fermée par toutes ces opérations. De plus, celles-ci préservent l'inclusion dans $\mathcal{T}$. □

Nous allons maintenant montrer que les familles ε-ALG et $\mathcal{T}$-ALG sont différentes. Pour ce faire, nous allons établir une adaptation de la version du lemme de pompage des langages algébriques pour ε-ALG. Enfin nous donnerons un exemple de langages algébrique inclus dans $\mathcal{T}$ qui n'est pas ε-sûr.

Théorème 3.2 (Lemme de pompage [BHPS60])**.** *Soit L un langage. Si L est algébrique, alors il existe un entier k tel que tout mot $u \in L$ de taille supérieure à k peut s'écrire comme $u = vwxyz$ avec*

1. *$|wxy| < k$,*
2. *$wy \neq \varepsilon$,*
3. *$vw^i xy^i z \in L$ pour tout $i \geq 0$.*

Ce résultat s'explique brièvement comme suit. Considérons une grammaire algébrique $\mathcal{G} = (N, T, P, S)$. Disons qu'un non-terminal peut s'autogénérer s'il existe une dérivation de la forme $X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} \Theta_1 X \Theta_2$. Puisque l'ensemble de productions P est fini, le nombre de dérivations au cours desquelles aucun non-terminal ne s'est autogénéré est fini. Du fait de cette finitude, la taille des phrases obtenues par des dérivations au cours desquelles aucun non-terminal ne s'est autogénéré peut être bornée, *ipso facto*, si un mot $u \in \mathscr{L}(\mathcal{G})$ est suffisamment grand, c'est qu'il a été obtenu par une dérivation au cours de laquelle un non-terminal s'est auto-généré : une dérivation de la forme

$$S \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} vXz \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} vwXyz \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} u = vwxyz$$

où vxz est de taille au plus trois la longueur maximum des dérivations sans non-terminal auto-généré et $wy \neq \varepsilon$ étant le facteur rendant le u suffisamment grand. Puisque $X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} wXy$, alors pour tout $i \geq 0 : X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} w^i X y^i$. Par conséquent le mot $vw^i xy^i z \in L$ pour tout $i \geq 0$.

Nous pouvons adapter ce résultat à ε-ALG comme suit.

Théorème 3.3. *Soit L un langage, si $L \in$ ε-ALG alors il existe un entier k tel que tout mot $u \in L$ de taille supérieur à k peut s'écrire comme $u = vwxyz$ avec*

1. $|wxy| < k$,
2. $wy \neq \varepsilon$,
3. $vz, wy, x \in \mathcal{T}$,
4. $vw^i xy^i z \in L$ *pour tout* $i \geq 0$.

Démonstration. La condition supplémentaire « $vz, wy, x \in \mathcal{T}$ » vient du fait que pour toute dérivation $X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} \Theta$ d'une grammaire algébrique ε-sûre, $\pi_T(\Theta) \in \mathcal{T}$. Par conséquent, lors d'une dérivation $S \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} vXz \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} vwXyz \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} vwxyz$, nous avons $vz, wy, x \in \mathcal{T}$. □

Proposition 3.27. *Il existe un langage $L \subseteq \mathcal{T}$ tel que $L \notin$ ε-ALG.*

Démonstration. Considérons le langage $L = \{(a\bar{a})^n b (a\bar{a})^n \bar{b}, n \geq 0\}$. Soit k un entier non-nul et $u = (a\bar{a})^k b (a\bar{a})^k \bar{b}$. Les seuls facteurs itérables dans le mot u sont les blocs de $a\bar{a}$. Pour remplir la dernière condition de l'énoncé du théorème 3.3 , il est nécessaire à toute décomposition $u = vwxyz$ que w et y soient des sous-mots des facteurs $(a\bar{a})^k$ situés respectivement à gauche et à droite du symbole b. Mais alors le facteur x contient le symbole b et z le symbole $\bar{b}$. Par conséquent, $x, vz \notin \mathcal{T}$. Il n'existe donc aucune factorisation satisfaisant simultanément les quatre contraintes. □

3.3.1 Problème de décision

La décision d'une propriété P pour une famille $\mathcal{F}$ se formule ainsi

ENTRÉE : $L \in \mathcal{F}$
SORTIE : Est-ce que L satisfait P ?

Un problème de décision est dit *décidable* si il existe un algorithme qui répond oui ou non à la question formulée en un nombre fini d'étapes de calcul. S'il n'existe pas de tel algorithme, le problème est dit *indécidable*. Le problème de l'inclusion « $L \subseteq M$? » est connu pour être indécidable pour $L, M \in$ ALG. Le problème devient toutefois décidable si le langage M est *super-déterministe* [GF80, Hop69]. Il est donc décidable de savoir si un langage algébrique L est inclus dans le langage de Dyck. Ce dernier résultat est aussi prouvé dans [BB00]. Dans [BCR09], les auteurs montrent que le problème reste décidable si le susmentionné langage M est un ensemble d'éléments qui se réduisent à un élément neutre par une réduction *simplifiable* (par exemple, la réduction $\bigcup_{a \in \widehat{A}} (a\bar{a}, \varepsilon)$). Il est donc décidable de savoir si un langage algébrique est inclus dans $\mathcal{T}$. Qu'en est-il de savoir si un langage algébrique ou $\mathcal{T}$-algébrique appartient à ε-ALG ? Pour ce qu'il est de savoir si un langage algébrique appartient à ε-ALG, il est montré dans [BB00] qu'il est indécidable de savoir si un langage algébrique peut être généré par une

grammaire XML[2]. Cette famille de langage est visiblement incluse dans ε-ALG, il en va de soit qu'il est indécidable de savoir si un langage algébrique appartient à ε-ALG. Nous étudions ici la question de savoir si un langage $\mathcal{T}$-algébrique appartient à ε-ALG. Nous montrons que ce problème est indécidable. À cette fin, nous montrons aussi qu'il est indécidable de savoir si un langage algébrique ε-sûr $L \subseteq \widehat{A}^*$ est égal à $\mathcal{T}_A$.

Notation. Dans toute cette sous-section, μ désigne l'homomorphisme $\mu : A^* \longrightarrow \widehat{A}^*$ tel que $\mu(a) = a\bar{a}$ pour tout $a \in A$.

Définition 3.28 (K-décomposition)**.** *Soit $K \subseteq A^*$ un langage. Un mot $w \in \mathcal{T}_A$ est dit K-décomposable si soit $w = \varepsilon$ et $\varepsilon \in K$, soit il admet une (unique) décomposition*

$$w = w_1\mu(v_1)w_2\mu(v_2)\cdots w_{n-1}\mu(v_{n-1})w_n$$

telle que
- *$n > 1$,*
- *chaque $w_i \in \widehat{A}^*$ et ne contient aucune occurrence de $a\bar{a}$ quel que soit $a \in A$,*
- *pour tout $i \in \{2, \dots, n-1\}$, $w_i \neq \varepsilon$,*
- *et chaque $v_i \neq \varepsilon$ et $v_i \in K$.*

L'unicité de la factorisation vient du fait qu'aucun facteur w_i ne contienne un facteur de la forme $\mu(x)$.

Lemme 3.29. *Soit $K \subseteq A^*$ un langage algébrique. L'ensemble L_K de mots K décomposables est ε-sûr.*

Démonstration. Soit R l'ensemble de mots sur l'alphabet $\widehat{A} \cup \{\$\}$ tels qu'aucun facteur de taille de 2 n'est de la forme $a\bar{a}$ quel que soit $a \in A$; et R' le langage

$$R' = \widehat{A}^*(\$\bar{\$}\widehat{A}^+)^*\$\bar{\$}\widehat{A}^*.$$

Le langage $M = \mathcal{T}_{A\cup\{\$\}} \cap R \cap R'$ est alors l'ensemble de mots $w \in \mathcal{T}_A$ de la forme

$$w = w_1\$\bar{\$}w_2\$\bar{\$}\cdots w_{n-1}\$\bar{\$}w_n$$

tels que
- $n > 1$,
- chaque $w_i \in \widehat{A}^*$ et ne contient aucune occurrence de $a\bar{a}$ quel que soit $a \in A$,
- pour tout $i \in \{2, \dots, n-1\}$, $w_i \neq \varepsilon$.

Puisque $\mathcal{T}_A \in \varepsilon$-ALG et que ε-ALG est fermé par intersection avec des langages réguliers (Proposition 3.17), le langage M est ε-sûr. De plus, pour tout langage algébrique $K \subseteq A^*$, le langage L_K peut s'exprimer comme $\nu(M)$ où ν est la substitution de $(\widehat{A} \cup \$)^*$ vers $2^{\widehat{A}^*}$ telle que

$$\nu(a) = \{a\} \ \ \forall a \in \widehat{A}; \qquad \nu(\$) = \mu(K) \cap \widehat{A}^+ \text{ et } \nu(\bar{\$}) = \{\varepsilon\}.$$

Rappelons que d'après le lemme 3.16, le langage $\mu(K)$ est algébrique ε-sûr, et puisque ε-ALG est fermé par intersection avec des langages réguliers, $\nu(\$) = \mu(K) \cap \widehat{A}^+$ est algébrique ε-sûr. Il s'en suit que le langage $L_K = \nu(M)$ est algébrique ε-sûr : pour construire une grammaire le générant, il suffit de substituer dans chaque production de la grammaire générant $\nu(M)$, le symbole terminal $\bar{\$}$ par ε et $\$$ par un non-terminal générant $\nu(\$)$ (et d'ajouter les productions permettant de générer $\nu(\$)$ depuis ce non-terminal). □

2. Les productions d'une grammaire XML sont de la forme $X \longrightarrow a\Theta\bar{a}$ avec $a \in A$ et Θ est une expression régulière sur l'alphabet des non-terminaux. De plus, un symbole $a \in A$ ne peut être généré que par une unique production.

Proposition 3.30. *Le problème « $L = \mathcal{T}_A$? » est indécidable pour $L \in \varepsilon$-ALG.*

Démonstration. Soit $K \subseteq A^*$ un langage algébrique et L_K l'ensemble de mots K-décomposables. Nous pouvons affirmer que « $K = A^*$ si et seulement si $L_K = \mathcal{T}_A$ ».

En effet, si $K = A^*$ alors L_K est tout simplement $\mathcal{T}_A$. Réciproquement, si $K \neq A^*$, alors il existe un mot $u \in A^* : u \notin K$; par conséquent, $\mu(u) \notin L_K$. C'est à dire $L_K \neq \mathcal{T}_A$.

Puisque le problème de l'universalité est indécidable pour les langages algébriques (voir [HU79]), l'affirmation ci-dessus implique alors qu'il est indécidable de savoir si un langage $L \in \varepsilon$-ALG satisfait $L = \mathcal{T}$. □

Théorème 3.4. *Le problème « $L \in \varepsilon$-ALG? » est indécidable pour $L \in \mathcal{T}$-ALG.*

Démonstration. Soit $L \subseteq \widehat{A}^*$ un langage de ε-ALG, et $L_0 \subseteq \widehat{A}^*$ un langage $\mathcal{T}$-ALG qui n'appartient pas à ε-ALG. Nous avons déjà montré (Proposition 3.27) que la famille ε-ALG est strictement incluse dans $\mathcal{T}$-ALG ; un tel langage L_0 existe donc. Considérons le langage

$$\varphi(L) = L_0 \# \bar{\#} \mathcal{T}_A \cup \mathcal{T}_A \# \bar{\#} L.$$

D'après les propriétés de clôture de $\mathcal{T}$-ALG(proposition 3.26), le langage $\varphi(L) \in \mathcal{T}$-ALG. Nous émettons l'affirmation suivante :

Affirmation 1. $L = \mathcal{T}_A$ *si et seulement si* $\varphi(L) \in \varepsilon$-ALG.

Preuve de l'affirmation 1. En effet, si $L = \mathcal{T}_A$, alors

$$\varphi(L) = L_0 \# \bar{\#} \mathcal{T}_A \cup \mathcal{T}_A \# \bar{\#} \mathcal{T}_A = \mathcal{T}_A \# \bar{\#} \mathcal{T}_A$$

qui par les propriétés de clôture de ε-ALG appartient à ε-ALG. Réciproquement, si $L \neq \mathcal{T}_A$, alors il existe un mot $u \in \mathcal{T}_A$ tel que $u \notin L$. Par conséquent,

$$\varphi(L).(\# \bar{\#} u)^{-1} = L_0.$$

Du fait que ε-ALG est fermée par quotient gauche et droit par $\mathcal{T}$ (Lemme 3.19), le langage $\varphi(L)$ n'appartient donc pas à ε-ALG parce que sinon, L_0 appartiendrait à ε-ALG. □

Puisqu'il est indécidable de savoir si un langage $L \in \varepsilon$-ALG satisfait $L = \mathcal{T}$ (Proposition 3.30), il est donc indécidable de savoir si un langage $L \in \mathcal{T}$ appartient à ε-ALG. □

Corollaire 3.31. *Le problème « $L \in \varepsilon$-ALG? » est indécidable pour $L \in$ ALG.*

3.4 Transductions algébriques ε-sûres

Dans cette dernière section, nous étendons la notions d'« algébrique ε-sûr » aux transductions. Nous définissons ici les transductions algébriques ε-sûres et nous prouvons différentes caractérisations de celles-ci.

Définition 3.32 (Transducteur Algébrique ε-sûr)**.** *Un transducteur algébrique (N, A, B, P, S) est dit ε-sûr si A est une union d'alphabets opposés : $A = \widehat{A_0}$ pour un certain alphabet A_0 et si toutes ses productions sont de la forme*

$$X \longrightarrow (\omega, u)\Theta(\overline{\omega}, v), \quad \textit{avec } \omega \in A^*, u, v \in B^*, \Theta \in N^*$$

Une transduction algébrique est dite ε-sûre si elle peut être générée par un transducteur algébrique ε-sûr.

De façon similaire aux langages ε-sûrs, cette restriction sur les formes des productions est sémantiquement équivalente à dire que pour tout non-terminal X, l'ensemble de couples dérivables depuis X est inclus dans $\mathcal{T}_{A_0} \times B^*$. La preuve peut en être faite en adaptant celle des langages aux transductions. Admettons, la proposition suivante.

Proposition 3.33. *Une transduction algébrique $\tau \subseteq \widehat{A}_0^* \times B^*$ est ε-sûre si, et seulement si, elle peut être générée par un transducteur algébrique $\mathcal{T} = (N, \widehat{A}_0, B, P, S)$ tel que pour tout non-terminal X : $\mathcal{R}(X) \subseteq \mathcal{T}_{A_0} \times B^*$.*

Lemme 3.34. *Toute transduction algébrique ε-sûre peut être générée par un transducteur algébrique $(N, A = \widehat{A}_0, B, P, S)$ dans lequel toutes les productions sont de la forme*

$$X \longrightarrow (\alpha, u)\Theta(\bar{\alpha}, v), \text{ avec } \alpha \in A \cup \{\varepsilon\}, u, v \in B^*, \Theta \in N^*$$

Démonstration. Il suffit pour ce faire de décomposer toute production $X \longrightarrow (\omega, u)\Theta(\overline{\omega}, v)$ en une suite de quelconque de productions $X_i \longrightarrow (\alpha_i, x_i)\Theta_{i+1}(\bar{\alpha}_i, y_i)$ avec $1 \leq i \leq l = \max\{|\omega|, |u|, |v|\}$, et telle que $\alpha_1 \cdots \alpha_l = \omega, x_1 \cdots x_l = u, y_1 \cdots y_l = v$, $X_1 = X, \Theta_l = \Theta$ et pour $1 < i < l$, Θ_i est un nouveau non-terminal X_i. □

Théorème 3.5. *Soit $\tau \subseteq \widehat{A}^* \times B^*$ une transduction. Les affirmations suivantes sont équivalentes*

1. *La transduction τ est algébrique ε-sûre.*
2. *Il existe un homomorphisme ε-sûr (symétrique et alphabétique) g, un homomorphisme f et un langage $L \in$ ε-ALG tels que*
$$\tau = \{(g(u), h(u)) \mid u \in L\}.$$
3. *Il existe un homomorphisme ε-sûr (symétrique)g, un homomorphisme f, un langage régulier R et $k \geq 1$ tels que*
$$\tau = \{(g(u), h(u)) \mid u \in R \cap \mathcal{D}_k\}.$$

Démonstration. $(1 \Rightarrow 2)$ Soit $\tau \subseteq \widehat{A}^* \times B$ une transduction algébrique ε-sûre. D'après le lemme 3.34, il existe un transducteur algébrique ε-sûr $\mathcal{T} = (N, \widehat{A}, B, P, S)$ tel que $\tau = \mathcal{R}(\mathcal{T})$ et dans lequel toute production est de la forme

$$X \longrightarrow (\alpha, u)\Theta(\bar{\alpha}, v) \quad \text{avec } \alpha \in \widehat{A} \cup \{\varepsilon\}, u, v \in B^*, \Theta \in N^*.$$

Considérons la grammaire algébrique ε-sûre $\mathcal{G} = (N, \widehat{P}, P', S)$ décrivant le langage d'arborescence des productions de $\mathcal{T}$: celle-ci est obtenue en remplaçant toute production $p = X \longrightarrow (\alpha, u)\Theta(\bar{\alpha}, v)$ avec $\alpha \in \widehat{A}^*, u, v \in B^*, \Theta \in N^*$ par la production $X \longrightarrow p\Theta\bar{p}$. Soit aussi l'homomorphisme $h : \widehat{P}^* \longrightarrow B^*$ et l'homomorphisme symétrique $g : \widehat{P}^* \longrightarrow \widehat{A}^*$ tels que pour toute production $p : X \longrightarrow (\alpha, u)\Theta(\bar{\alpha}, v) \in P$:

$$g(p) = \omega, g(\bar{p}) = \bar{\alpha}, \quad h(p) = u \text{ et } h(\bar{p}) = v.$$

L'ensemble de productions de $\mathcal{T}$ peut alors se définir comme

$$P = \left\{X \longrightarrow (g(p), h(p))\Theta(g(\bar{p}), h(\bar{p})) \mid X \longrightarrow p\Theta\bar{p} \in P'\right\}.$$

La transduction $\mathcal{R}(\mathcal{T})$ peut par conséquent se définir comme :

$$\mathcal{R}(\mathcal{T}) = \{(g(u), h(u)) \mid u \in \mathscr{L}(\mathcal{G})\}.$$

$(2 \Rightarrow 3)$ D'après le théorème 3.1, tout langage $L \in \varepsilon\text{-ALG}$ peut se décrire comme $f(R \cap \mathcal{D}_k)$ où f est un homomorphisme ε-sûr, R un régulier et k un entier. La transduction

$$\tau = \{(g(u), h(u)) \mid u \in L\}, \text{ avec } L \in \varepsilon\text{-ALG}$$

peut alors se réécrire comme

$$\begin{aligned}\tau &= \{(g(u), h(u)) \mid u \in f(R \cap \mathcal{D}_k)\} \text{ avec } f \text{ } \varepsilon\text{-sûr, } R \text{ régulier} \\ &= \{(g \circ f(v), h \circ f(v)) \mid v \in R \cap \mathcal{D}_k\}.\end{aligned}$$

Les homomorphismes f et g étant ε-sûrs, l'homomorphisme $g \circ f$ l'est aussi.

$(3 \Rightarrow 1)$ Il s'agit d'une construction réciproque à celle de l'implication $(1 \Rightarrow 2)$. Le langage de Dyck est ε-sûr, ainsi d'après les propriétés de clôture de ε-ALG, tout langage de la forme $R \cap \mathcal{D}_k$ avec R régulier, est ε-sûr ; et est par conséquent généré par une grammaire algébrique ε-sûre $\mathcal{G}$. La relation $\tau = \{(g(u), h(u)) \mid u \in R \cap \mathcal{D}_k\}$ peut alors être générée par le transducteur $\mathcal{T}$ obtenu en remplaçant tout symbole terminal a dans les productions de $\mathcal{G}$ par le couple $(g(a), f(a))$. Puisque l'homomorphisme g est ε-sûr et que pour tout non-terminal X de $\mathcal{G}$ on a $\mathscr{L}(X) \subseteq \mathcal{T}_k$; alors pour chaque non-terminal X dans $\mathcal{T}$ on aura alors $\mathcal{R}(X) \subseteq \mathcal{T}_k \times B^*$. La relation τ est donc ε-sûre. □

3.4.1 Images par transductions ε-sûres

Proposition 3.35. ALG $= \{\tau(R) \mid R \textit{ est régulier}, \tau \textit{ algébrique et } \varepsilon\textit{-sûre}\}$.

Démonstration. D'après le théorème de Chomsky–Schützenberger, tout langage algébrique peut s'exprimer comme $L = f(R \cap \mathcal{D}_k)$ c'est-à-dire comme $L = \tau(\varepsilon)$ où τ est la transduction algébrique $\{(\varepsilon, f(u)) \mid u \in R \cap \mathcal{D}_k\}$ qui d'après le théorème 3.5 est ε-sûre.

Réciproquement, d'après le théorème 3.5, pour tout langage régulier R et pour toute transduction algébrique ε-sûre τ, il existe un homomorphisme f, un homomorphisme ε-sûr g un langage régulier R' et un entier k tels que

$$\tau(R) \quad = f(R' \cap \mathcal{D}_k \cap g^{-1}(R)) \quad = f(R'' \cap \mathcal{D}_k) \quad \text{avec } R'' = R' \cap g^{-1}(R);$$

qui d'après le théorème 3.1 implique que $\tau(R) \in \varepsilon\text{-ALG}$. □

On pourrait naturellement se demander à quelle famille de langage appartiendrait un langage $\tau(L)$ si L est algébrique ε-sûr, et τ algébrique ε-sûre.

Proposition 3.36. *La famille des récursivement énumérables peut se décrire comme*

$$\text{RE} = \{\tau(L) \mid \tau \textit{ est algébrique } \varepsilon\textit{-sûr}, L \textit{ algébrique } \varepsilon\textit{-sûr}\}.$$

Démonstration. $(\Rightarrow)$ D'après [HOY85, HN81], pour tout langage récursivement énumérable $E \subseteq A^*$, il existe un langage algébrique $K \subseteq \widehat{B}^*$ et un homomorphisme $f : \widehat{B}^* \longrightarrow A^*$ tel que $E = f(K \cap \mathcal{D}_B)$. Soit alors l'homomorphisme $\mu : \widehat{B}^* \longrightarrow \widehat{C}^*$, avec $|C| = |B|$ tel que

$$\mu(b_i) = c_i\bar{c_i} \text{ et } \mu(\bar{b_i}) = \bar{c_i}c_i \quad \text{pour tout } b_i \in B.$$

L'homomorphisme μ est alors injectif et ε-sûr. De plus, pour tout $u \in \widehat{B}^*$: $\rho(\mu(u)) = \varepsilon$. D'après le lemme 3.16, le langage $\mu(K) \in \varepsilon$-ALG et par injectivité, $\mu^{-1}(\mu(K)) = K$. Nous obtenons alors

$$E \;=\; f(K \cap \mathcal{D}_B) \;=\; f(\mu^{-1}(\mu(K)) \cap \mathcal{D}_B) \;=\; \tau(L)$$

avec $L = \mu(K) \in \varepsilon$-ALG et $\tau = \{(\mu(u), f(u)) \mid u \in \mathcal{D}_B\}$.

Le sens ($\Leftarrow$) découle trivialement des propriétés de clôture de RE. □

Chapitre 4

Théorèmes de représentation des langages indexés

4.1 Introduction

Le théorème de Chomsky–Schützenberger [CS63] a connu plusieurs généralisations à diverses familles de langages. Il est prouvé dans [HOY85, HN81], qu'un langage est récursivement énumérable si et seulement si il est l'image par un homomorphisme de l'intersection entre un langage algébrique (linéaire, simplement linéaire) et le langage de Dyck. En particulier, Hirose et Nasu mettent en relation le fait qu'une famille admette une grammaire universelle et le fait qu'elle admette un certain type de théorème de représentation. Il existe dans la littérature diverses généralisations du théorème de Chomsky–Schützenberger pour la famille des langages indexés et ces sous-classes. En particulier, Arnold et Dauchet [AD77] établissent un théorème de représentation pour les forêts algébriques [3] ; Duske, Parchamann et Secht prouvent [DPS79] qu'un langage est indexé si et seulement si il est l'image par un homomorphisme de l'intersection entre un langage algébrique « spécial » [4] et le langage de Dyck.

Deux résultats qui retiennent particulièrement notre attention ici sont la généralisations de Weir [Wei88] pour les langages indexés *linéaires* et celles de Kanazawa et Sorokin [Kan14b, Sor14] pour les pour les frontières d'arbres algébriques *simples*. Weir montre qu'un langage est linéaire indexé si et seulement si il est l'image par transduction rationnelle d'un langage de la forme $\mathcal{D}_4 \cap \mathrm{w}^{-1}(\mathcal{D}_4)$ où w est un homomorphisme précis. Cette double restriction au langage de Dyck permet de décrire des dépendances syntaxiques plus fortes car elle impose que non-seulement un mot soit bien parenthésé, mais aussi que son image par l'homomorphisme w soit bien parenthésée. Il est alors possible d'associer une même parenthèse à deux autres de cette façon. Ce résultat a ensuite été généralisé par Kanazawa et Sorokin aux frontières d'arbres algébriques *simples* via une généralisation de l'homomorphisme de Weir.

Nous généralisons ici ces résultats aux langages indexés. Nous prouvons (1) qu'un langage est indexé si et seulement il est l'image du langage de Dyck par une transduction algébrique ε-sûre. Nous montrons pour cela qu'une dérivation d'une grammaire indexée peut être associée à une dérivation d'un transducteur algébrique dans laquelle la suite d'actions d'indices de la grammaire indexée est codée comme un mot de Dyck appartenant au domaine de la transduction ε-sûre. De façon similaire, nous montrons aussi qu'un langage est indexé si et seulement

3. La frontière d'une forêt algébrique est un langage indexé [Gue83].

4. Les « special type of context-free languages » mentionnés dans [DPS79] s'avèrent être très similaires aux langages algébriques ε-sûrs que nous avons étudié. Je tiens à remiercier Johannes Osterholzer pour cette référence.

si il est l'image par homomorphisme de l'intersection entre un langage algébrique ε-sûr et le langage de Dyck. Puis, nous montrons (2) que la contrainte « ε-sûr » peut être relaxée. Nous montrons qu'un langage est indexé si et seulement il est l'image du langage de Dyck par une transduction $\mathcal{T}$-algébrique. Nous disons qu'une transduction est $\mathcal{T}$-algébrique si celle-ci est algébrique et si son domaine est inclus dans $\mathcal{T}$. Finalement, nous montrons (3) qu'un langage est indexé si et seulement si il est l'image du langage $\mathcal{D}_6 \cap \sigma^{-1}(\mathcal{D}_2)$ par une transduction rationnelle ; où l'homomorphisme σ est un homomorphisme symétrique et alphabétique que nous définissons. La famille des langages indexés est donc le cône principal généré par le langage susmentionné.

Plan du chapitre Ce chapitre est sectionné comme suit : dans la section 4.2, nous prouvons les caractérisations par langages et transductions algébriques ε-sûrs. La section 4.3 est dédiée à la preuve de la caractérisation par transductions $\mathcal{T}$-algébriques. La section 4.4 est dédiée à la preuve de la principalité de la famille des langages indexés.

4.2 Représentation par transductions algébriques ε-sûres

Nous montrons dans cette section qu'un langage est indexé si et seulement si il peut se décrire comme $\tau(\mathcal{D}_k)$ ou encore $\tau(\mathcal{D}_2)$ où la transduction τ est algébrique ε-sûre et k est un entier positif. Nous en dérivons deux autres caractérisations : un langage est indexé si et seulement si il peut s'exprimer comme $f(K \cap \mathcal{D}_k)$ où le langage K est algébrique ε-sûr, k est un entier positif et f un homomorphisme ; un langage est indexé si et seulement si il peut s'exprimer comme $f(R \cap \mathcal{D}_k \cap g^{-1}(\mathcal{D}_q))$ où R est un langage régulier, k et q sont deux entiers positifs et f est un homomorphisme et g est un homomorphisme ε-sûr.

Rappelons (Lemme 3.34) que toute transduction algébrique ε-sûre peut-être décrite par un transducteur $(N, \widehat{A}, B, P, S)$ dans lequel toute production est de la forme

$$X \longrightarrow (\alpha, u)\Theta(\bar{\alpha}, v) \quad \text{avec } \alpha \in \widehat{A} \cup \{\varepsilon\}, u, v \in B^*, \Theta \in N^*.$$

De même (Corollaire 2.31), tout langage indexé peut être généré par une grammaire indexée (N, T, I, P, S) dans laquelle chaque production est sous l'une des formes suivantes :

$$X \longrightarrow u\Theta^{\alpha}v; \quad X^{\alpha} \longrightarrow u\Theta v; \qquad \text{avec } \Theta \in N^*, u, v \in T^*, \alpha \in I \cup \{\varepsilon\}.$$

Par besoin de nomenclature, qualifions ces formes de *formes adéquates*. Nous pouvons alors associer à chaque grammaire indexée en forme adéquate un transducteur algébrique ε-sûr en forme adéquate par la transformation suivante.

Définition 4.1. *Soit* IG_ET *la transformation qui à toute grammaire indexée en forme adéquate* $\mathcal{I} = (N, T, I, P, S)$ *associe le transducteur algébrique ε-sûr en forme adéquate* $\text{IG\_ET}(\mathcal{I}) = (N, \widehat{I}, T, P', S)$ *comme suit :*

— *pour tout* $p = X \longrightarrow u\Theta^{\alpha}v \in P$ *avec* $\Theta \in N^*, \alpha \in I \cup \{\varepsilon\}, u, v \in T^*$ *:*

$$\text{IG\_ET}(p) = X \longrightarrow (\alpha, u)\Theta(\bar{\alpha}, v) \in P';$$

— *pour tout* $p = X^{\alpha} \longrightarrow u\Theta v \in P$, *avec* $\Theta \in N^*, \alpha \in I, u, v \in T^*$ *:*

$$\text{IG\_ET}(p) = X \longrightarrow (\bar{\alpha}, u)\Theta(\alpha, v) \in P'.$$

Remarque 4.2. *La transformation* IG_ET *est une bijection entre l'ensemble de grammaires indexées en forme adéquate et l'ensemble des transducteurs algébriques ε-sûrs en forme adéquate.*

Définition 4.3. *Pour toute grammaire indexée $\mathcal{I}$ en forme adéquate, nous appelons* IG_ET$(\mathcal{I})$ *le* transducteur associé *à $\mathcal{I}$.*

Prenons pour la suite cet exemple simple.

Exemple 4.4. *Soit la grammaire indexée décrite par l'ensemble de productions suivant :*

$$S \longrightarrow {S_0}^{\$}; \qquad S_0 \longrightarrow a{S_0}^{\alpha}d + X;$$
$$X^{\$} \longrightarrow \varepsilon; \qquad X^{\alpha} \longrightarrow bXc.$$

Le langage généré par cette grammaire est $\{a^n b^n c^n d^n \mid n \geq 0\}$. Le transducteur IG_ET$(\mathcal{I})$ *associé est décrit par l'ensemble de productions suivant :*

$$S \longrightarrow (\$, \varepsilon)S_0(\bar{\$}, \varepsilon); \qquad S_0 \longrightarrow (\alpha, a)S_0(\bar{\alpha}, d) + X;$$
$$X \longrightarrow (\bar{\$}\$, \varepsilon); \qquad X \longrightarrow (\bar{\alpha}, b)X(\alpha, c).$$

La relation générée par ce transducteur est $\{(\$\alpha^n\bar{\alpha}^m\bar{\$}\$\alpha^m\bar{\alpha}^n\bar{\$}, a^n b^m c^m d^n) \mid n, m \geq 0\}$.

Lemme 4.5. *Soit $\mathcal{I} = (N, T, I, P, S)$ une grammaire indexée en forme adéquate, et* IG_ET$(\mathcal{I}) = (N, \widehat{I}, T, P', S)$ *le transducteur algébrique ε-sûr associé. Pour tout $X^{\delta_0} \in N^{I^*}$, soit $\delta = \delta_0^R$. Il existe une dérivation*

$$X^{\delta_0} \underset{\mathcal{I}}{\overset{*}{\Longrightarrow}} u_1 Y_1^{x_1} u_2 Y_2^{x_2} u_3 \cdots u_n Y_n^{x_n} u_{n+1} \text{ dans } \mathcal{I}$$

si et seulement si il existe une dérivation

$$X \underset{\text{IG\_ET}(\mathcal{I})}{\overset{*}{\Longrightarrow}} (\omega_1, u_1)Y_1(\omega_2, u_2)Y_2(\omega_3, u_3)\cdots(\omega_n, u_n)Y_n(\omega_{n+1}, u_{n+1}) \text{ dans IG\_ET}(\mathcal{I})$$

telle que $\delta\omega_1\cdots\omega_{n+1}\bar{\delta} \in \mathcal{D}_I$ et $x_i = \rho(\delta\omega_1\cdots\omega_i)^R$ pour tout $i \in \{1, \ldots, n\}$.

Démonstration. Nous montrons le lemme ci-dessus par induction sur la taille des dérivations. L'affirmation est trivialement vraie pour toute dérivation de taille 1 conformément aux formes des productions de $\mathcal{I}$ et IG_ET$(\mathcal{I})$. Il nous suffit maintenant de prouver le pas d'induction. Considérons deux dérivations de l'énoncé du lemme et montrons que l'énoncé reste vrai pour toute dérivation qui en découle.

Considérons sans perte de généralité une dérivation depuis la variable Y_2. Nous considérons deux cas en fonction de s'il s'agit d'une production d'empilement ou de dépilement.

Empilement. Par construction, il existe une productions $Y_2 \longrightarrow vZ_1^{\alpha}\cdots Z_m^{\alpha}v' \in P$ si et seulement si il existe une production $Y_2 \longrightarrow (\alpha, v)Z_1\cdots Z_m(\bar{\alpha}, v) \in$ IG_ET(P). En appliquant ces deux productions, nous obtenons les dérivations

$$X^{\delta_0} \underset{\mathcal{I}}{\overset{*}{\Longrightarrow}} u_1 Y_1^{x_1} u_2 v Z_1^{\alpha x_2}\cdots Z_m^{\alpha x_2} v' u_3 \cdots u_n Y_n^{x_n} u_{n+1}$$
$$\text{et } X \underset{\text{IG\_ET}(\mathcal{I})}{\overset{*}{\Longrightarrow}} (\omega_1, u_1)Y_1(\omega_2\alpha, u_2 v)Z_1\cdots Z_m(\bar{\alpha}\omega_3, v'u_3)\cdots(\omega_n, u_n)Y_n(\omega_{n+1}, u_{n+1}).$$

De plus, le mot $\delta\omega_1\omega_2\alpha\bar{\alpha}\omega_3\cdots\omega_n\omega_{n+1}\bar{\delta}$ appartient à $\mathcal{D}_I$. Aussi, $\rho(\delta\omega_1\omega_2\alpha)^R = \alpha x_2$ puisque $x_2 = \rho(\delta\omega_1\omega_2)^R$ et $\alpha \in I$; et $\rho(\delta\omega_1\omega_2\alpha\bar{\alpha}\omega_3)^R = \rho(\delta\omega_1\omega_2\omega_3)^R = x_3$.

Dépilement. Il existe une production $p = Y_2^{\alpha} \longrightarrow vZ_1 \cdots Z_m v' \in P$ si et seulement si il existe une production $\text{IG\_ET}(p) = Y_2 \longrightarrow (\bar{\alpha}, v)Z_1 \cdots Z_m(\alpha, v') \in \text{IG\_ET}(P)$. La phrase $Y_2^{x_2}$ ne peut être dérivée en appliquant p que si *premier*$(x_2) = \alpha$. Nous avons donc cette fois ci deux cas à considérer en fonction de l'applicabilité de la production p. Nous montrons que p est applicable si et seulement si l'affirmation du lemme est vraie.

— Si p peut être appliquée, alors $x_2 = \alpha x_0$ avec $x_0 \in I^*$. En appliquant p et $\text{IG\_ET}(p)$, on obtient les dérivations

$$X^{\delta_0} \underset{\mathcal{I}}{\overset{*}{\Longrightarrow}} u_1 Y_1^{x_1} u_2 v Z_1^{x_0} \cdots Z_m^{x_0} v' u_3 \cdots u_n Y_n^{x_n} u_{n+1}$$
$$\text{et } X \underset{\text{IG\_ET}(\mathcal{I})}{\overset{*}{\Longrightarrow}} (\omega_1, u_1)Y_1(\omega_2\bar{\alpha}, u_2 v)Z_1 \cdots Z_m(\alpha\omega_3, v'u_3) \cdots (\omega_n, u_n)Y_n(\omega_{n+1}, u_{n+1}).$$

De plus, $\rho(\delta\omega_1\omega_2\bar{\alpha})^R = x_0$ puisque $\rho(\delta\omega_1\omega_2)^R = x_2 = \alpha x_0$; et

$$\rho(\delta\omega_1\omega_2\bar{\alpha}\alpha\omega_3) = \rho(\delta\omega_1\omega_2\omega_3) = x_3.$$

Ceci implique que $\delta\omega_1\omega_2\bar{\alpha}\alpha\omega_3 \cdots \omega_{n+1}\bar{\delta}$ appartient à $\mathcal{D}_I$.

— Si la production p ne peut être appliquée, alors, *premier*$(x_2) \neq \alpha$. En appliquant la production $\text{IG\_ET}(p)$ à la dérivation de $\text{IG\_ET}(\mathcal{I})$, on obtient la dérivation

$$X \underset{\text{IG\_ET}(\mathcal{I})}{\overset{*}{\Longrightarrow}} (\omega_1, u_1)Y_1(\omega_2\bar{\alpha}, u_2 v)Z_1 \cdots Z_m(\alpha\omega_3, v'u_3)Y_3 \cdots Y_n(\omega_{n+1}, u_{n+1}).$$

Mais cette fois, le mot $\delta\omega_1\omega_2\bar{\alpha}\alpha\omega_3 \cdots \omega_{n+1}\bar{\delta}$ n'appartient pas à $\mathcal{D}_I$ puisque

$$\textit{dernier}(x_2) = \textit{dernier}(\rho(\delta\omega_1\omega_2)) \neq \alpha$$

et par conséquent *dernier*$(\rho(\delta\omega_1\omega_2\bar{\alpha})) = \bar{\alpha}$. C'est-à-dire que le mot $\delta\omega_1\omega_2\bar{\alpha}$ n'est pas un préfixe de $\mathcal{D}_I$ (il contient plus de symboles fermants que de symboles ouvrants).

□

Théorème 4.1. *Soit $\mathcal{I} = (N, T, I, P, S)$ une grammaire indexée en forme adéquate et $\mathcal{T} = \text{IG\_ET}(\mathcal{I}) = (N, \widehat{I}, T, \text{IG\_ET}(P), S)$ le transducteur algébrique ε-sûr associé. Pour tout $X^{\delta_0} \in N^{I^*}$:*

$$\mathscr{L}(X^{\delta_0}) = \mathcal{R}(X)(\delta^{-1} \cdot \mathcal{D}_I \cdot \bar{\delta}^{-1}) \quad \textit{avec } \delta = \delta_0^R.$$

Démonstration. D'après le lemme 4.5, il existe une dérivation terminale $X^{\delta_0} \underset{\mathcal{I}}{\overset{*}{\Longrightarrow}} u$ si et seulement si il existe une dérivation $X \underset{\mathcal{T}}{\overset{*}{\Longrightarrow}} (\omega, u)$ avec $\delta\omega\bar{\omega}\bar{\delta} \in \mathcal{D}_I$ avec $\delta = {\delta_0}^R$. C'est à dire

$$\begin{aligned} \mathscr{L}(X^{\delta_0}) &= \{u \mid \exists\omega : (\omega, u) \in \mathcal{R}(X) \text{ et } \delta\omega\bar{\delta} \in \mathcal{D}_I\} && \text{avec } \delta = {\delta_0}^R \\ &= \mathcal{R}(X)(\delta^{-1} \cdot \mathcal{D}_I \cdot \bar{\delta}^{-1}) && \text{avec } \delta = {\delta_0}^R. \end{aligned}$$

□

Théorème 4.2. *Un langage L est indexé si, et seulement si, il existe une transduction algébrique ε-sûre τ et $k \geq 1$ tels que $L = \tau(\mathcal{D}_k)$.*

Démonstration. ($\Rightarrow$) Soit $\mathcal{I} = (N, T, I, P, S)$ une grammaire indexée en forme adéquate et τ la transduction algébrique ε-sûre générée par le transducteur associé. Nous avons $\mathscr{L}(\mathcal{I}) = \mathscr{L}(S^{\varepsilon}) = \tau(\mathcal{D}_I)$. Le sens ($\Leftarrow$) découle de la bijectivité de la transformation IG_ET. Toute transduction algébrique ε-sûre peut être générée par une transducteur algébrique ε-sûr $\mathcal{T}$ en forme adéquate. Soit alors $\mathcal{T}_k$ le transducteur obtenu en restreignant $\mathcal{T}$ sur un alphabet d'entrée de taille k. A ce dernier on peut alors associer la grammaire indexée $\mathcal{I} = \text{IG\_ET}^{-1}(\mathcal{T}_k)$ telle que $\mathscr{L}(\mathcal{I}) = \mathcal{R}(\mathcal{T}_k)(\mathcal{D}_k)$. □

Exemple 4.6. *Considérons la grammaire indexée de l'exemple 4.4 et son transducteur associé. Le langage généré par la grammaire indexée est* $L = \{a^n b^n c^n d^n \mid n \geq 0\}$ *et la transduction décrite par le transducteur associé est*

$$\tau = \{(\$\alpha^n \bar{\alpha}^m \bar{\$} \$ \alpha^m \bar{\alpha}^n \bar{\$}, a^n b^m c^m d^n) \mid n, m \geq 0\}.$$

L'ensemble de couples (ω, u) *de la forme* $(\$\alpha^n \bar{\alpha}^m \bar{\$} \$ \alpha^m \bar{\alpha}^n \bar{\$}, a^n b^m c^m d^n)$ *tels que* $\omega \in \mathcal{D}_{\{\alpha,\$\}}$ *sont alors de la forme*

$$(\$\alpha^n \bar{\alpha}^n \bar{\$} \$ \alpha^n \bar{\alpha}^n \bar{\$}, a^n b^n c^n d^n).$$

Nous avons alors $\tau(\mathcal{D}_{\{\alpha,\$\}}) = L$.

Corollaire 4.7. *Un langage est indexé si et seulement si il existe un homomorphisme* f*, un homomorphisme* ε*-sûr (symétrique, symétrique et alphabétique)* g*, un langage régulier* R *et* $k, q \geq 1$ *tels que*

$$L = f\big(R \cap \mathcal{D}_k \cap g^{-1}(\mathcal{D}_q)\big).$$

Démonstration. D'après le théorème de représentation des transductions algébriques ε-sûres (Théorème 3.5), tout langage $\tau(\mathcal{D}_q)$ où τ est une transduction algébrique ε-sûre s'exprime de façon équivalente comme

$$f(R \cap \mathcal{D}_k \cap g^{-1}(\mathcal{D}_q))$$

où R est un langage régulier, f est un homomorphisme, k un entier positif et g est un homomorphisme ε-sûr (symétrique, symétrique et alphabétique). □

Corollaire 4.8. *Soit* L *un langage. Les affirmations suivantes sont équivalentes.*

1. L *est indexé.*
2. *Il existe un entier positif* k*, un langage régulier* R*, un homomorphisme* f *et un homomorphisme* ε*-sûr (symétrique, symétrique et alphabétique)* g *tels que*
$$L = f(R \cap \mathcal{D}_k \cap g^{-1}(\mathcal{D}_2)).$$
3. *Il existe une transduction algébrique* ε*-sûre* τ *telle que* $L = \tau(\mathcal{D}_2)$.

Démonstration. ($1 \Rightarrow 2$) D'après le corollaire 4.7, le langage L peut s'exprimer comme

$$L = f(R \cap \mathcal{D}_k \cap g^{-1}(\mathcal{D}_q))$$

où f est un homomorphisme, R est un langage régulier, g est un homomorphisme ε-sûr (symétrique, symétrique et alphabétique), k et q sont deux entiers positifs. Puisque pour tout $q \geq 1$, le langage $\mathcal{D}_q$ peut s'exprimer comme $\mu^{-1}(\mathcal{D}_2)$ où μ est l'homomorphisme de $\widehat{\{a_1, \ldots, a_q\}}^*$ dans $\widehat{\{0,1\}}^*$ tel que pour tout $i \in \{1, \ldots, q\}$:

$$\mu(a_i) = 01^i0 \qquad \text{et} \qquad \mu(\bar{a}_i) = \overline{01^i0};$$

cet homomorphisme étant clairement ε-sûr, le langage L peut alors s'exprimer comme

$$L = f(R \cap \mathcal{D}_k \cap (\mu \circ g)^{-1}(\mathcal{D}_2)).$$

$(2 \Rightarrow 3)$ D'après le théorème de représentation des transductions algébriques ε-sûrs (Théorème 3.5).

$(3 \Rightarrow 1)$ D'après le corollaire 4.7. □

Théorème 4.3. *Un langage L est indexé si, et seulement si, il existe un langage $K \in \varepsilon$-ALG, un homomorphisme f et $k \geq 1$ tels que*

$$L = f(K \cap \mathcal{D}_k).$$

Démonstration. $(\Leftarrow)$ La transformation $\tau = f \circ \cap K$ est algébrique ε-sûre puisqu'elle peut s'exprimer comme $\tau = \{(g(u), f(u)) \mid u \in K\}$ où g est l'identité. D'après le théorème 4.2, le langage L est donc indexé.

$(\Rightarrow)$ Soit $L \subseteq A^*$ un langage indexé. D'après les théorèmes 3.5 et 4.2, il existe un langage $K \subseteq B^* \in \varepsilon$-ALG, un homomorphisme $f : \widehat{B}^* \longrightarrow A^*$, un homomorphisme ε-sûr $g : \widehat{B}^* \longrightarrow \widehat{C}^*$ tels que $L = f(K \cap g^{-1}(\mathcal{D}_C))$. Considérons maintenant l'homomorphisme $\mu : \widehat{B}^* \longrightarrow \widehat{(C \cup A)}^*$ tel que pour tout $b \in \widehat{B} : \mu(b) = g(b)f(b)\overline{f(b)}$. Puisque g est ε-sûr, l'homomorphisme μ l'est aussi. De plus, par définition de μ, pour tout symbole $b \in \widehat{B} : g(b)$ et $\mu(b)$ sont équivalents modulo η-réduction ($f(b)\overline{f(b)} \to \varepsilon$). Par conséquent, pour tout mot $u \in \widehat{B}^* : g(u) \in \mathcal{D}_C$ si, et seulement si, $\mu(u) \in \mathcal{D}_{A \cup C}$, c'est à dire $g^{-1}(\mathcal{D}_C) = \mu^{-1}(\mathcal{D}_{A \cup C})$. De plus, $f = \pi_A \circ \mu$. Nous obtenons alors,

$$\begin{aligned} L = f(K \cap g^{-1}(\mathcal{D}_C)) &= f(K \cap \mu^{-1}(\mathcal{D}_{A \cup C})) \\ &= \pi_A\big(\mu(K \cap \mu^{-1}(\mathcal{D}_{A \cup C}))\big) \\ &= \pi_A(K' \cap \mathcal{D}_{A \cup C}) \quad \text{avec } K' = \mu(K). \end{aligned}$$

De plus, par propriété de clôture de ε-ALG, le langage K' est ε-sûr. Ceci conclut la preuve du théorème ci-dessus. □

4.3 Transductions $\mathcal{T}$-algébriques

Nous avons montré (Théorème 4.2) qu'un langage est indexé si et seulement si il est l'image du langage de Dyck par une transduction algébrique ε-sûre. Nous montrons ici que nous pouvons relaxer la contrainte « ε-sûre » : nous montrons qu'un langage est indexée si et seulement il est l'image du langage de Dyck par une transduction algébrique pour laquelle le domaine est un sous ensemble de $\mathcal{T}$.

Définition 4.9. *Une transduction $\tau \subseteq \widehat{A}^* \times B^*$ est dite $\mathcal{T}$-algébrique si elle est algébrique et si* $\mathrm{dom}(\tau) \subseteq \mathcal{T}_A$.

Puisque toute transduction algébrique ε-sûre est $\mathcal{T}$-algébrique, il nous suffira de montrer un seul sens de l'inclusion : tout langage indexé peut s'exprimer comme l'image du langage de Dyck par une transduction $\mathcal{T}$-algébrique. Nous montrons pour cela que pour toute transduction $\mathcal{T}$-algébrique $\tau \subseteq \widehat{A}^* \times B^*$, il existe une transduction algébrique ε-sûre $\tau' \subseteq \widehat{A}^* \times B^*$ telle que $\tau(\mathcal{D}_A) = \tau'(\mathcal{D}_A)$. La transduction τ' susmentionnée ne sera pas équivalente à τ. Celle-ci sera obtenue en ajoutant des éléments aux productions du transducteur générant τ de sorte que le transducteur obtenu soit ε-sûr ; le tout en préservant l'image du langage de Dyck par τ.

Lemme 4.10. *Soit $\mathcal{G} = (N, T = \widehat{A}, P, S)$ une grammaire algébrique réduite telle que $\mathscr{L}(\mathcal{G}) \subseteq \mathcal{T}_A$. Pour tout non-terminal $X \in N$: il existe un unique mot $\rho_X \in \widehat{A}^*$ tel que*

$$\rho(\mathscr{L}(X)) \;=\; \{\rho(u) \mid u \in \mathscr{L}(X)\} \;=\; \{\rho_X\}.$$

De plus, l'ensemble des mots ρ_X peut être effectivement calculé.

Démonstration. Puisque $\rho(u) = \varepsilon$ pour tout mot $u \in \mathscr{L}(\mathcal{G})$, alors pour tout non-terminal $X \in N$, il existe un unique mot ρ_X tel que $\rho(\mathscr{L}(X)) = \{\rho_X\}$. En effet supposons par l'absurde une dérivation $S \xRightarrow[\mathcal{G}]{*} u_1 X u_2$ et deux mots v_1, v_2 dérivables depuis X tels que $\rho(v_1) \neq \rho(v_2)$. On obtient alors

$$S \xRightarrow[\mathcal{G}]{*} u_1 v_1 u_2 \quad \text{et} \quad S \xRightarrow[\mathcal{G}]{*} u_1 v_2 u_2 \qquad \text{avec } \rho(u_1 v_1 u_2) \neq \rho(u_1 v_2 u_2);$$

ce qui contredit l'hypothèse que $\rho(u) = \varepsilon$ pour tout $u \in \mathscr{L}(\mathcal{G})$.

Effectivité du calcul des ρ_X. Les mots réduits ρ_X peuvent être effectivement calculés comme suit : Tant que $P \neq \emptyset$:

1. Choisir une production $X \longrightarrow u \in P$ avec $u \in T^*$.
2. Assigner $\rho_X = \rho(u)$.
3. Supprimer toute production $X \longrightarrow \Theta \in P$; faire $N = N - X$.
4. Remplacer toute production de la forme $Y \longrightarrow \Theta_1 X \Theta_2$ par $Y \longrightarrow \Theta_1 \rho_X \Theta_2$.

Terminaison. Toute grammaire algébrique réduite possède nécessairement au moins une production terminale : de la forme $X \longrightarrow u$ avec $u \in T^*$. Chaque itération dans la procédure supprime un non-terminal associé à une telle production et la grammaire qui en résulte reste évidemment réduite. Le nombre de non-terminaux et de productions décroit alors avec chaque itération. La procédure termine donc.

Validité. La validité de la procédure découle à la fois de l'unicité de chaque mot ρ_X : il suffit de considérer une production terminale pour calculer le mot réduit associé à celui-ci ; et de la confluence du système de réécriture $\{a\bar{a} \rightsquigarrow \varepsilon \mid a \in \widehat{A}\}$ (Lemme 2.16) : pour tout mot $u = x_1 x_2 x_3$, $\rho(u) = \rho(x_1 \rho(x_2) x_3)$. L'étape de 4 préserve alors la valeur des mots réduits associés à chaque non-terminal. Tous les mots ρ_X calculés correspondent donc à ceux de la grammaire initiale. □

Dans ce qui suit, nous montrons que pour toute transduction $\mathcal{T}$-algébrique $\tau \subseteq \widehat{A}^* \times B^*$, il existe une transduction algébrique ε-sûre $\tau' \subseteq \widehat{A}^* \times B^*$ telle que $\tau(\mathcal{D}_A) = \tau'(\mathcal{D}_A)$. Il en découlera que le langage $\tau(\mathcal{D}_A)$ est indexé.

Lemme 4.11. *Soit un mot $uvw \in \widehat{A}^*$ et $x = \rho(v)$. Alors, $uvw \in \mathcal{D}_A$ ssi $uv\bar{x}xw \in \mathcal{D}_A$.*

Démonstration. Rappelons qu'un mot u appartient à $\mathcal{D}$ ssi $\rho(u) = \varepsilon$ et pour tout préfixe $u_0 \preccurlyeq u$ et pour tout symbole $a \in A$: $|u_0|_a \geq |u_0|_{\bar{a}}$. Il est clair que $\rho(uv\bar{x}xw) = \rho(uvw)$. De plus, l'ajout du facteur $\bar{x}x$ dans le mot uvw (ou la suppression de celui-ci dans le mot $uv\bar{x}xw$) préserve la différence entre le nombre de symboles positifs et négatifs (puisque $x = \rho(v)$). □

Théorème 4.4. *Pour toute transduction $\mathcal{T}$-algébrique $\tau \subseteq \widehat{A}^* \times B^*$, il existe une transduction algébrique ε-sûre $\tau' \subseteq \widehat{A}^* \times B^*$ telle que $\tau(\mathcal{D}_A) = \tau'(\mathcal{D}_A)$.*

Démonstration. Soit $\mathcal{G} = (N, \widehat{A}, B, P, S)$ un transducteur algébrique générant τ. Nous pouvons supposer sans perte de généralité que toute production est sous l'une des formes

$$(1) X \longrightarrow YZ \quad \text{avec } Y, Z \in N; \quad (2) X \longrightarrow (\omega, u) \quad \text{avec } \omega \in \widehat{A}^*, u \in B^*;$$

et que de plus $\mathcal{G}$ est réduit : pour tout non-terminal X, il existe une dérivation $S \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} \Theta_1 X \Theta_2$ et une dérivation $X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} (\omega, u)$ avec $\omega \in \widehat{A}^*, u \in B^*$. D'après le lemme 4.10, pour tout $X \in N$, il existe un unique mot ρ_X tel que pour toute dérivation $X \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} (\omega, u)$, nous avons $\rho(\omega) = \rho_X$. Considérons pour chaque non-terminal X, cet unique mot réduit ρ_X.

Pour la clarté de l'analyse de la validité de notre construction, nous construisons le transducteur générant τ' en deux étapes. La première consiste à construire un transducteur $\mathcal{G}_1 = (N \cup N_\varepsilon, \widehat{A}, B, P_1, S)$ tel que $\mathcal{R}(\mathcal{G})(\mathcal{D}_A) = \mathcal{R}(\mathcal{G}_1)(\mathcal{D}_A)$ et tel que pour tout $X_\varepsilon \in N_\varepsilon$ et pour toute dérivation $X_\varepsilon \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}} (\omega, u)$, $\omega \in \mathcal{T}_A$. La deuxième étape consiste à construire à partir de $\mathcal{G}_1$ un transducteur $\mathcal{G}_2$ équivalent à ce dernier et n'ayant que $S \cup N_\varepsilon$ comme ensemble de non-terminaux. De ce fait, celui-ci sera ε-sûr.

Étape 1. Le transducteur $\mathcal{G}_1 = (N_1 = N \cup N_\varepsilon, \widehat{A}, B, P_1, S)$ est construit comme suit : $N_\varepsilon = \{X_\varepsilon \mid X \in N\}$ et l'ensemble de productions P_1 est défini comme suit

$$\begin{aligned} & \{X \longrightarrow (\omega, u) \in P\} \\ P_1 = \cup\ & \{X \longrightarrow Y_\varepsilon(\rho_Y, \varepsilon) Z_\varepsilon(\rho_Z, \varepsilon) \mid X \longrightarrow YZ \in P\} \\ \cup\ & \{X_\varepsilon \longrightarrow X(\overline{\rho_X}, \varepsilon) \mid X \in N\}. \end{aligned}$$

Toutes les productions de $\mathcal{G}_1$ sont alors de la forme

$$(1) X \longrightarrow (\omega, u); \quad (2) X \longrightarrow Y_\varepsilon(\rho_Y, \varepsilon) Z_\varepsilon(\rho_Z, \varepsilon); \quad (3) X_\varepsilon \longrightarrow X(\overline{\rho_X}, \varepsilon).$$

Remarque 4.12. *Par construction, pour toute production $X \longrightarrow (\omega, u)$ on a $\rho(\omega) = \rho_X$ et pour tout $X \longrightarrow Y_\varepsilon(\rho_Y, \varepsilon) Z_\varepsilon(\rho_Z, \varepsilon) \in P_1$ c'est-à-dire pour tout $X \longrightarrow YZ \in P$, on a $\rho_X = \rho(\rho_Y \rho_Z)$.*

Pour tout $X \in N_1$, soit $\mathrm{dom}_1(X) = \{\omega \in \widehat{A}^* \mid \exists u : X \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}} (\omega, u)\}$. Nous montrons dans un premier temps que

$$\forall X \in N : \rho(\mathrm{dom}_1(X)) = \{\rho_X\} \quad \text{et} \quad \rho(\mathrm{dom}_1(X_\varepsilon)) = \{\varepsilon\}. \tag{4.1}$$

Nous nous intéressons en particulier aux dérivations des non-terminaux $X \in N$. Les non-terminaux $X_\varepsilon \in N_\varepsilon$ ne sont dérivables que par des productions de la forme (3). Considérons alors la relation $\underset{\mathcal{G}_1}{\Longrightarrow}'$ qui décrit la dérivabilité d'un non-terminal $X \in N$ par soit une production de type (1), soit par un enchainement de production de type (2) suivie de deux productions de type (3) : Pour tout $\Theta, \Theta' \in (N \cup (\widehat{A}^* \times B^*))^*$, $\Theta \underset{\mathcal{G}_1}{\Longrightarrow}' \Theta'$ si il existe $\Theta_0, \Theta_1 \in (N \cup (\widehat{A}^* \times B))^*$ et $X \in N$ tels $\Theta = \Theta_0 X \Theta_1$ et

$$\Theta' = \Theta_0(\omega, u)\Theta_1 \quad \text{et il existe } X \longrightarrow (\omega, u) \in P_1;$$

ou alors

$$\Theta' = \Theta_0 Y(\overline{\rho_Y}\rho_Y, \varepsilon) Z(\overline{\rho_Z}\rho_Z, \varepsilon)\Theta_1 \quad \text{et il existe } X \longrightarrow Y_\varepsilon(\rho_Y, \varepsilon) Z_\varepsilon(\rho_Z, \varepsilon) \in P_1.$$

Soit $\underset{g,\mathcal{G}_1}{\Longrightarrow}'$ la relation de dérivabilité gauche vis-à-vis de $\underset{g,\mathcal{G}_1}{\Longrightarrow}'$. Soient $\overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}}'$ et $\overset{*}{\underset{g,\mathcal{G}_1}{\Longrightarrow}}'$ les clôtures réflexives et transitives de $\underset{\mathcal{G}_1}{\Longrightarrow}'$ et $\underset{g,\mathcal{G}_1}{\Longrightarrow}'$. Nous ne considérerons des dérivations gauches « $\overset{*}{\underset{g,\mathcal{G}_1}{\Longrightarrow}}$ » que pour simplifier nos énoncés. Nous pouvons émettre l'affirmation suivante

Affirmation 1.

$$\begin{aligned}\textit{Pour tout } X \in N : \{(\omega,u) \mid X \overset{*}{\underset{g,\mathcal{G}_1}{\Longrightarrow}}' (\omega,u)\} &= \{(\omega,u) \mid X \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}}' (\omega,u)\} \\ &= \{(\omega,u) \mid X \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}} (\omega,u)\} \\ &= \{(\omega,u) \mid X \overset{*}{\underset{g,\mathcal{G}_1}{\Longrightarrow}} (\omega,u)\}.\end{aligned}$$

Affirmation 2. Pour tout $X \in N : \rho(\mathrm{dom}_1(X)) = \{\rho_X\}$.

Preuve de l'affirmation . Par construction, pour toute dérivation $X \underset{\mathcal{G}_1}{\Longrightarrow} (\omega,u)$, on a $\rho(w) = \rho_X$. Inductivement, pour toute dérivation $X \underset{\mathcal{G}_1}{\Longrightarrow}' Y(\overline{\rho_Y}\rho_Y,\varepsilon)Z(\overline{\rho_Z}\rho_Z,\varepsilon)$, si $Y \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}}' (\omega_Y,u_Y)$ et $Z \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}}' (\omega_Z,u_Z)$, on obtient

$$\rho(\omega_Y\overline{\rho_Y}\rho_Y\omega_Z\overline{\rho_Z}\rho_Z) \;=\; \rho(\omega_Y\omega_Z) \;=\; \rho(\rho_Y\rho_Z) \;=\rho_X.$$

□

Affirmation 3. Pour tout $X_\varepsilon \in N_\varepsilon : \rho(\mathrm{dom}_1(X_\varepsilon)) = \{\varepsilon\}$.

Preuve de l'affirmation 3. D'après la définition de P_1, $\mathrm{dom}_1(X_\varepsilon) = \mathrm{dom}_1(X)\overline{\rho_X}$. D'après l'affirmation 1, on a alors $\rho(\mathrm{dom}_1(X_\varepsilon)) = \rho(\mathrm{dom}_1(X)\overline{\rho_X}) = \rho(\rho_X\overline{\rho_X}) = \varepsilon$. □

L'affirmation qui suit découle du fait que lors d'une dérivation $X \overset{*}{\underset{\mathcal{G}_1}{\Longrightarrow}}' Y_1(\overline{\rho_{Y_1}}\rho_{Y_1},\varepsilon)Y_2(\overline{\rho_{Y_2}}\rho_{Y_2},\varepsilon)$, chaque non-terminal Y_i précède un facteur $\overline{\rho_{Y_i}}\rho_{Y_i}$. Du fait que $\rho(\mathrm{dom}_1(Y_i)) = \{\rho_{Y_i}\}$, ce facteur $\overline{\rho_{Y_i}}\rho_{Y_i}$ sera toujours, tout au long des dérivations, soit précédé soit de Y_i, soit d'un facteur se réduisant à ρ_{Y_i}.

Affirmation 4. Toute dérivation gauche de taille ≥ 1 est de la forme

$$S \overset{*}{\underset{g,\mathcal{G}_1}{\Longrightarrow}}' (\omega,u)X_1(\omega_{1,1}\cdots\omega_{1,n_1},\varepsilon)X_2(\omega_{2,1}\cdots\omega_{2,n_2},\varepsilon)\cdots(X_m\omega_{m,1}\cdots\omega_{m,n_m},\varepsilon) \tag{4.2}$$

avec $m \geq 0$, chaque $n_i \geq 0$ et pour tout i,j : le mot $\omega\rho_{X_1}\omega_{1,1}\cdots\omega_{1,n_1}\cdots\rho_{X_i}\omega_{i,1}\cdots\omega_{i,j-1}$ admet une factorisation xy telle que $\omega_{i,j} = \overline{\rho(y)}\rho(y)$.

Preuve de l'affirmation 4. Par induction sur la taille des dérivations.

Cas initial. C'est trivialement vrai pour toute dérivation de la forme $X \underset{g,\mathcal{G}_1}{\Longrightarrow}' (\omega,u)$ ou de la forme $X \underset{g,\mathcal{G}_1}{\Longrightarrow}' Y(\overline{\rho_Y}\rho_Y,\varepsilon)Z(\overline{\rho_Z}\rho_Z,\varepsilon)$.

Pas d'induction. Supposons une dérivation de la forme 4.2, puisque toute étape de dérivation de X_1 est soit de la forme

$$X_1 \underset{g,\mathcal{G}_1}{\Longrightarrow}{}' (\omega_{X_1}, u_{x_1}) \quad \text{avec } \rho(\omega_{X_1}) = \rho_{X_1}$$

soit de la forme

$$X_1 \underset{g,\mathcal{G}_1}{\Longrightarrow}{}' Y(\overline{\rho_Y}\rho_Y, \varepsilon)Z(\overline{\rho_Z}\rho_Z, \varepsilon) \quad \text{avec } \rho(\rho_Y\overline{\rho_Y}\rho_Y\rho_Z\overline{\rho_Z}\rho_Z) = \rho_X,$$

quel que soit la dérivation de X_1, la dérivation obtenue est de la forme 4.2. □

Il est alors clair par construction et d'après l'affirmation 3 qu'il existe une dérivation

$$S \underset{g,\mathcal{G}}{\overset{*}{\Longrightarrow}} (v, u)$$

si et seulement si il existe une dérivation

$$S \underset{g,\mathcal{G}_1}{\overset{*}{\Longrightarrow}}{}' (v_1\omega_{1,1}\cdots\omega_{1,n_1}v_2\omega_{2,1}\cdots\omega_{2,n_2}\cdots v_m\omega_{m,1}\cdots\omega_{m,n_m}, u)$$

telle que $v = v_1v_2\cdots v_m$ et pour tout i, j : le mot $v_1\omega_{1,1}\cdots\omega_{1,n_1}v_2\omega_{2,1}\cdots\omega_{2,n_2}v_iv_{i,1}\cdots v_{i,j-1}$ admet une factorisation xy avec $\omega_{i,j} = \overline{\rho(y)}\rho(y)$. De plus, par itération du lemme 4.11,

$$\begin{aligned} v = v_1v_2\cdots v_m \in \mathcal{D}_A &\Leftrightarrow v_1\omega_{1,1}v_2\cdots v_m \in \mathcal{D}_A \\ &\Leftrightarrow v_1\omega_{1,1}\omega_{1,2}v_2\cdots v_m \in \mathcal{D}_A \\ &\vdots \\ &\Leftrightarrow v_1\omega_{1,1}\omega_{1,2}\cdots\omega_{1,n_1}v_2\cdots v_m \in \mathcal{D}_A \\ &\vdots \\ &\Leftrightarrow v_1\omega_{1,1}\cdots\omega_{1,n_1}v_2\omega_{2,1}\cdots\omega_{2,n_2}\cdots v_m\omega_{m,1}\cdots\omega_{m,n_m} \in \mathcal{D}_A. \end{aligned}$$

On a alors $\mathcal{R}(\mathcal{G}_1)(\mathcal{D}_A) = \tau(\mathcal{D}_A)$.

Etape 2. Rappelons que toute production de $\mathcal{G}_1$ est sous l'une des formes suivantes

$$(1)X \longrightarrow (\omega, u); \quad (2)X \longrightarrow Y_\varepsilon(\rho_Y, \varepsilon)Z_\varepsilon(\rho_Z, \varepsilon); \quad (3)X_\varepsilon \longrightarrow X(\overline{\rho_X}, \varepsilon).$$

Le transducteur algébrique ε-sûr $\mathcal{G}_2 = (N_\varepsilon \cup \{S\}, \widehat{A}, B, P_2, S)$ tel que

$$\mathcal{R}(\mathcal{G}_2)(\mathcal{D}_A) = \mathcal{R}(\mathcal{G}_1)(\mathcal{D}_A) = \tau(\mathcal{D}_A)$$

est construit à partir de $\mathcal{G}_1$ en remplaçant, pour tout $X \neq S$, toute production $X_\varepsilon \longrightarrow X(\overline{\rho_X}, \varepsilon)$ par $X_\varepsilon \longrightarrow \Theta(\overline{\rho_X}, \varepsilon)$ pour toute production $X \longrightarrow \Theta \in P_1$. Le transducteur obtenu est naturellement équivalent. Ce dernier est de plus ε-sûr puisque pour tout $X \in N_\varepsilon \cup \{S\}$: $\rho(\mathrm{dom}_1(X)) = \{\varepsilon\}$. Ceci conclut la preuve du théorème ci-dessus. □

Corollaire 4.13. *Soit $\tau \subseteq \widehat{A}^* \times B^*$ une transduction $\mathcal{T}$-algébrique. Le langage $\tau(\mathcal{D}_A)$ est indexé.*

Corollaire 4.14. *Un langage $L \subseteq B^*$ est indexé si et seulement si il existe une transduction $\mathcal{T}$-algébrique $\tau \subseteq \widehat{A}^* \times B^*$ telle que $L = \tau(\mathcal{D}_A)$.*

4.4 Principalité de la famille des indexés

Nous montrons dans cette dernière section que la famille des langages indexée est le cône principal généré par un langage de la forme $\mathcal{D}_6 \cap \sigma^{-1}(\mathcal{D}_2)$ où σ est un homomorphisme symétrique et alphabétique que nous définissons plus bas. Un corollaire du théorème 4.2, du théorème de représentations des transductions algébriques ε-sûres (Théorème 3.5) et du théorème de Nivat est le suivant :

Corollaire 4.15. *Un langage L est indexé si, et seulement si, il existe une transduction rationnelle τ, un homomorphisme ε-sûr (symétrique) g et $k, q \geq 1$ tels*

$$L = \tau(\mathcal{D}_k \cap g^{-1}(\mathcal{D}_q)).$$

Définition 4.16. *Soient A et B deux alphabets disjoints et $\sigma_{A,B} : \widehat{(A \cup B \cup B')}^* \longrightarrow \widehat{B}^*$ l'homomorphisme*

$$\sigma_{A,B}(a) = \varepsilon \ \ \forall a \in A, \quad \sigma_{A,B}(b) = b \ \ \forall b \in B, \quad \sigma_{A,B}(b') = \bar{b} \ \ \forall b' \in B'$$
$$\text{et } \sigma_{A,B}(\bar{x}) = \overline{\sigma_{A,B}(x)} \ \ \forall x \in \overline{A \cup B \cup B'}.$$

Nous écrivons σ au lieu de $\sigma_{A,B}$ lorsque les alphabets A et B sont implicitement définis.

Définition 4.17 (Langage de Dyck de niveau 2)**.** *Étant donné deux alphabets A et B, soit*

$\mathcal{D}^2_{A,B} = \mathcal{D}_{A \cup B \cup B'} \cap \sigma^{-1}(\mathcal{D}_B)$;
et $\mathcal{D}^2 = \mathcal{D}^2_{A,B}$ avec $A = \{a_1, a_2\}, B = \{b_1, b_2\}$.

Le langage $\mathcal{D}^2$ sera notre générateur pour la famille des indexés.

Exemple 4.18. *Soit $A = \{a_1, a_2\}$ et $B = \{b_1, b_2\}$.*
— *Le mot $u_1 = b_1 a_1 b'_1 \bar{b}'_1 \bar{a}_1 \bar{b}_1 \in \mathcal{D}^2$ puisque $u_1 \in \mathcal{D}_{A \cup B \cup B'}$ et $\sigma(u_1) = b_1 \bar{b}_1 b_1 \bar{b}_1$ appartient à $\mathcal{D}_B$.*

— *Le mot $v = b'_2 b_1 \bar{b}_1 \bar{b}'_2$ n'appartient pas à $\mathcal{D}^2$ puisque $\sigma(v) = \bar{b}_2 b_1 \bar{b}_1 \bar{b}_2 \notin \mathcal{D}_B$.*

Dans ce qui suit nous montrons que pour tout langage $L = \mathcal{D}_k \cap g^{-1}(\mathcal{D}_q)$, où g est un homomorphisme symétrique, il existe un homomorphisme f tel que $\mathcal{D}^2 = f^{-1}(L)$. Ceci en conjonction avec le corollaire 4.15, aura pour conséquence que $\text{IL} = \mathscr{C}(\mathcal{D}^2)$.

Considérons les classes d'homomorphismes suivantes.

Définition 4.19 (Homomorphismes ε-sûrs unilatéraux)**.** *Soit $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ un homomorphisme. Celui-ci est dit*
— positif *si* $\forall a \in A : \eta(f(a)) \in B^*$;
— ε-sûr unilatéral *si* $\forall u \in \mathcal{D}_A : f(u) \in \mathcal{D}_B$;
— fortement ε-sûr unilatéral *si* $\forall u : u \in \mathcal{D}_A \Leftrightarrow f(u) \in \mathcal{D}_B$.

Remarque 4.20. *La propriété* fortement ε-sûr unilatéral *peut de façon équivalente se formuler comme « $\mathcal{D}_A = f^{-1}(\mathcal{D}_B)$ ».*

Lemme 4.21 (Folklore)**.** *L'homomorphisme $f_k : \widehat{\{a_1, a_2, \ldots, a_k\}}^* \longrightarrow \widehat{\{0,1\}}^*$ tel que*

$$f_k(a_i) = 01^i0 \quad \text{et} \quad f_k(\bar{a}_i) = \overline{01^i0} \quad i \in \{1, \ldots, |A|\}$$

est fortement ε-sûr unilatéral (C'est celui-ci qui permet d'établir la version forte du théorème de Chomsky–Schützenberger : Pour tout $k \geq 1 : \mathcal{D}_k = f_k^{-1}(\mathcal{D}_2)$).

Lemme 4.22. *Soit $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ un homomorphisme symétrique. Si f est positif, alors il est également ε-sûr unilatéral.*

Démonstration. En effet, le langage $f(\mathcal{D}_A)$ est généré par la grammaire

$$S \longrightarrow \varepsilon + SS + \sum_{a \in A} f(a) S f(\bar{a}).$$

Si f est symétrique et positif, le langage $f(\mathcal{D}_A)$ est clairement inclus dans $\mathcal{D}_B$. □

Définition 4.23. *Soient $f_1 : \widehat{A}^* \longrightarrow \widehat{B}^*$ et $f_2 : \widehat{A}^* \longrightarrow \widehat{C}^*$ deux homomorphismes. Nous appelons* concaténation *de f_1 et f_2 et notons $f_1 \cdot f_2$ l'homomorphisme de $\widehat{A}^*$ vers $\widehat{(B \cup C)}^*$ tel que*

$$f_1 \cdot f_2(a) = f_1(a) f_2(a) \quad \text{et} \quad f_1 \cdot f_2(\bar{a}) = f_2(\bar{a}) f_1(\bar{a}) \qquad \forall a \in A.$$

Nous disons de plus que cette concaténation est disjointe *si les alphabets B et C sont disjoints.*

Exemple 4.24. *Soit $f : \widehat{\{a\}}^* \longrightarrow \widehat{\{a\}}^*$ et $g : \widehat{\{a\}}^* \longrightarrow \widehat{\{b\}}^*$ tels que*

$$f(a) = aa; \quad f(\bar{a}) = \bar{a}; \qquad g(a) = b\bar{b}; \quad g(\bar{a}) = b.$$

L'homomorphisme $f \cdot g : \widehat{\{a\}}^* \longrightarrow \widehat{\{b\}}^*$ est alors définit comme

$$f \cdot g(a) = aab\bar{b}; \quad f \cdot g(\bar{a}) = b\bar{a}.$$

Lemme 4.25. *Soient B et C deux alphabets disjoints. Soient $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ un homomorphisme fortement ε-sûr unilatéral et $g : \widehat{A}^* \longrightarrow \widehat{C}^*$ un homomorphisme ε-sûr unilatéral. Les homomorphismes $f \cdot g$ et $g \cdot f$ sont fortement ε-sûrs unilatéraux.*

Démonstration. La preuve pour les deux cas est la même. Soit $\mu = f \cdot g$. Nous montrons que pour tout $u \in \widehat{A}^* : u \in \mathcal{D}_A \Leftrightarrow \mu(u) \in \mathcal{D}_{B \cup C}$.

Affirmation 1. Pour tout $u \in \widehat{A}^* : u \in \mathcal{D}_A \Rightarrow \mu(u) \in \mathcal{D}_{B \cup C}$.

Preuve de l'affirmation 1. En effet, soit $a \in A$, on a

$$\mu(a\bar{a}) = f(a) g(a) g(\bar{a}) f(\bar{a}) = f(a) g(a\bar{a}) f(\bar{a}).$$

Puisque f et g sont ε-sûrs unilatéraux, $g(a\bar{a}) \in \mathcal{D}_C$, $f(a\bar{a}) \in \mathcal{D}_B$ par conséquent $\mu(a\bar{a}) \in \mathcal{D}_{B \cup C}$. L'homomorphisme μ est donc ε-sûr unilatéral. □

Affirmation 2. Pour tout $u \in \widehat{A}^* : \mu(u) \in \mathcal{D}_{B \cup C} \Rightarrow u \in \mathcal{D}_A$.

Preuve de l'affirmation 2. Soit $u \in \widehat{A}^*$. D'après le lemme 2.21, si $\mu(u) \in \mathcal{D}_{B \cup C}$, la projection $\pi_{\widehat{B}}$ de $\widehat{B \cup C}$ dans $\widehat{B}$ étant ε-sûr unilatéral (pour tout $a \in (B \cup C) : \pi_{\widehat{B}}(a\bar{a}) \in \mathcal{D}_B$), nous avons aussi $\pi_{\widehat{B}}(\mu(u)) \in \mathcal{D}_B$. Puisque les alphabets B et C sont disjoints, $\pi_{\widehat{B}}(\mu(u)) = f(u)$. Nous obtenons ainsi $f(u) \in \mathcal{D}_B$ et puisque f est fortement ε-sûr unilatéral, il s'en suit que $u \in \mathcal{D}_A$. □

Ceci conclut la preuve du lemme ci-dessus. □

Lemme 4.26. *Soient A et B deux alphabets, et $g : \widehat{A}^* \longrightarrow \widehat{B}^*$ un homomorphisme symétrique. Il existe un homomorphisme $f : \widehat{A}^* \longrightarrow \widehat{(A \cup B \cup B')}^*$ tel que*

$$\mathcal{D}_A \cap g^{-1}(\mathcal{D}_B) = f^{-1}(\mathcal{D}^2_{A,B}).$$

Démonstration. Nous devons montrer qu'il existe un homomorphisme f tel que pour tout mot $u \in \widehat{A}^* : u \in \mathcal{D}_A \Leftrightarrow f(u) \in \mathcal{D}_{A \cup B \cup B'}$ et $g(u) \in \mathcal{D}_B \Leftrightarrow \sigma \circ f(u) \in \mathcal{D}_B$.

Soit l'homomorphisme $\mu : \widehat{B}^* \longrightarrow \widehat{(B \cup B')}^*$ tel que

$$\forall b \in B : \mu(b) = b, \quad \mu(\bar{b}) = b'.$$

Soit l'homomorphisme $f : \widehat{A}^* \longrightarrow \widehat{(A \cup B \cup B')}^*$ tel que

$$\forall a \in A : f(a) = a\mu \circ g(a) \quad \text{et} \quad f(\bar{a}) = \overline{f(a)} = \overline{\mu \circ g(a)}\bar{a}.$$

Affirmation 1. Pour tout mot $u \in \widehat{A}^* : u \in \mathcal{D}_A \Leftrightarrow f(u) \in \mathcal{D}_B$.

Preuve de l'affirmation 1. En effet, l'homomorphisme f peut s'exprimer comme la concaténation disjointe $f = f_1 \cdot f_2$ où f_1 est l'identité et est par conséquent fortement ε-sûr unilatéral, et f_2 est l'homomorphisme de $\widehat{A}^*$ dans $\widehat{B}^*$ tel que

$$\forall a \in A : f_2(a) = \mu \circ g(a) \quad \text{et} \quad f_2(\bar{a}) = \overline{\mu \circ g(a)}.$$

Puisque g est symétrique, et par définition de μ, l'homomorphisme f_2 est symétrique et positif. D'après le lemme 4.22, l'homomorphisme f_2 est ε-sûr unilatéral et d'après le lemme 4.25, l'homomorphisme f est alors fortement ε-sûr unilatéral. □

Rappelons que l'homomorphisme σ remplace les symboles $b' \in B'$ par $\bar{b} \in \overline{B}$ et les symboles $\bar{b'} \in \overline{B'}$ par $b \in B$. Par conséquent, pour tout $b \in \widehat{B} : \sigma \circ \mu(b) = b$. Nous pouvons alors émettre l'affirmation suivante.

Affirmation 2. Pour tout $u \in \widehat{B}^* : \sigma \circ \mu(u) = u$.

Affirmation 3. Pour tout $u \in \widehat{A}^* : g(u) \in \mathcal{D}_B \Leftrightarrow \sigma \circ f(u) \in \mathcal{D}_B$.

Preuve de l'affirmation 3. En effet, par définitions de f et σ, et d'après l'affirmation 2, pour tout $a \in A$:

$$\begin{aligned} \sigma \circ f(a) &= \sigma(a\mu \circ g(a)) = \sigma(\mu \circ g(a)) = g(a) \\ \text{et } \sigma \circ f(\bar{a}) &= \sigma(\overline{\mu \circ g(a)}) = \overline{g(a)} = g(\bar{a}). \end{aligned}$$

Par conséquent $\sigma \circ f = g$. De ce fait, pour tout $u \in \widehat{A}^* : g(u) \in \mathcal{D}_B \Leftrightarrow \sigma \circ f(u) \in \mathcal{D}_B$. □

Il en découle que pour tout mot $u \in \widehat{A}^*$:

$$u \in \mathcal{D}_A \cap g^{-1}(\mathcal{D}_B) \Leftrightarrow f(u) \in \mathcal{D}_A \cap \sigma^{-1}(\mathcal{D}_B).$$

Ceci conclut la preuve de la proposition ci-dessus. □

Lemme 4.27. *Soient A et B deux alphabets. Il existe un homomorphisme symétrique f tel que $\mathcal{D}^2_{A,B} = f^{-1}(\mathcal{D}^2)$.*

Démonstration. Supposons $A = \{a_1, a_2, \cdots, a_{|A|}\}$ et $B = \{b_1, b_2, \cdots, b_{|B|}\}$. Soit alors

$$A_2 = \{0_A, 1_A\} \text{ et } B_2 = \{0_B, 1_B\}.$$

L'homomorphisme $f : \widehat{(A \cup B \cup B')}^* \longrightarrow \widehat{(A_2 \cup B_2 \cup B_2')}^*$ est le codage binaire des éléments des alphabets A, B et C définit comme suit :

$$a_i \in A \mapsto 0_A 1_A^i 0_A, \qquad b_i \in B \mapsto 0_B 1_B^i 0_B, \qquad b'_i \in B' \mapsto 0_{B'} 1_{B'}^i 0'_B ;$$
$$\text{et } f(\bar{x}) = \overline{f(x)} \ \ \forall x \in \overline{A \cup B \cup B'}.$$

Celui-ci est alors clairement fortement ε-sûr unilatéral. Nous pouvons émettre l'affirmation suivantes :

Affirmation 1. *Pour tout* $u \in (A \widehat{\cup B \cup} B')^* : u \in \mathcal{D}_{A \cup B \cup B'} \Leftrightarrow f(u) \in \mathcal{D}_{A_2 \cup B_2 \cup B'_2}$.

De plus, par définition les homomorphismes $\sigma_{A,B}$ et $\sigma_{A_2,B_2} \circ f$ sont comme suit :

$$\begin{aligned}
\forall a \in A : \sigma_{A,B}(a) = \varepsilon \quad & \text{et} \quad \sigma_{A_2,B_2} \circ f(a) = \varepsilon; \\
\forall b_i \in B : \sigma_{A,B}(b_i) = b_i \quad & \text{et} \quad \sigma_{A_2,B_2} \circ f(b_i) = \sigma_{A_2,B_2}(01_B^i 0) = 01_B^i 0; \\
\forall b'_i \in B' : \sigma_{A,B}(b'_i) = \bar{b}_i \quad & \text{et} \quad \sigma_{A_2,B_2} \circ f(b'_i) = \sigma_{A_2,B_2}(01_B'^i 0) = \overline{01_B^i 0}.
\end{aligned}$$

L'homomorphisme $\sigma_{A_2,B_2} \circ f$ peut alors s'exprimer comme $\mu \circ \sigma_{A,B}$ où μ est le codage binaire des symboles de B. Il s'en suit d'après le lemme 4.21 que pour tout $u \in (A \widehat{\cup B \cup} B')^*$:

$$\sigma_{A,B}(u) \in \mathcal{D}_B \Leftrightarrow \sigma_{A_2,B_2} \circ f(u) \in \mathcal{D}_{B_2}.$$

Ceci conclut la preuve du lemme ci-dessus. □

Théorème 4.5. *Un langage L est indexé si, et seulement si, il existe une transduction rationnelle τ telle que $L = \tau(\mathcal{D}^2)$.*

Démonstration. En combinant le corollaire 4.15 et les lemmes 4.27 et 4.26.

D'après le corollaire 4.15 tout langage s'exprimant comme $\tau(\mathcal{D}^2)$ où τ est une transduction rationnelle est indexé. Réciproquement, tout langage indexé L peut s'exprimer comme $\tau(\mathcal{D}_k \cap g^{-1}(\mathcal{D}_q))$ où l'homomorphisme g est ε-sûr et la transduction τ rationnelle. D'après les lemmes 4.26 et 4.27 le langage peut s'exprimer comme $\tau(\mathcal{D}^2)$. □

Corollaire 4.28. *La famille des langages indexés est le cône principal généré par $\mathcal{D}^2$.*

Chapitre 5

Caractérisation logique des langages indexés doubles

5.1 Introduction

La théorie des langages et la complexité descriptive partagent la longue tradition de caractérisations logiques de familles de langages. Le résultat fondateur en est le théorème de Büchi–Elgot–Trakhtenbrot [Bö60, Bö90, Elg61, Tra61] qui stipule qu'un langage est régulier si et seulement si il est définissable par une formule du second ordre monadique (MSO).

Dans [LST94], les auteurs prennent une autre approche et s'intéressent à des fragments syntaxiques/sémantiques du second ordre existentiel (∃SO). Pour toute classe de relation $\mathcal{B}$, ceux-ci définissent la logique $\exists\mathcal{B}$ FO (resp. $\exists\mathcal{B}$ MSO) comme l'ensemble de formules de la forme $\phi \equiv \exists X\, \phi_0$; où X est une variable représentant une relation binaire et ϕ_0 une formule du premier ordre (resp. du second ordre monadique). Un mot u satisfait ϕ si et seulement si il existe une relation $X \in \mathcal{B}$ sur l'ensemble de positions de u telle que le couple (u, X) satisfait ϕ_0. Ils montrent ainsi qu'un langage est algébrique si et seulement si il est définissable dans la logique ∃MatchFO et si et seulement si il est définissable dans la logique ∃MSO ; où Match la classe de relations de parenthésage appelés *matchings* (voir illustration figure 5.1). Ce résultat est visiblement analogue au théorème de Chomsky–Schützenberger. Des approches similaires

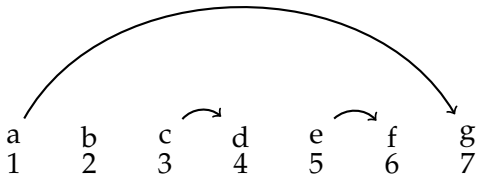


FIGURE 5.1 – Un matching $M = \{(1,7),(3,4),(5,6)\}$ sur le mot $abcdefg$.

ont été employées à diverses caractérisations [LMSV99, LMSV99, Lan06] de familles de langages/classes de complexité et dans [VDH16], les auteurs considèrent l'usage de *prédicats de comportement* sur des logique *à poids* pour caractériser des familles principales.

Nous pensons que la définissabilité des langages algébriques dans un tel fragment sémantique est possible par le fait que les grammaires algébriques soient lexicalisables et en particulier, tout langage algébrique peut être généré par une grammaire algébrique en forme normale

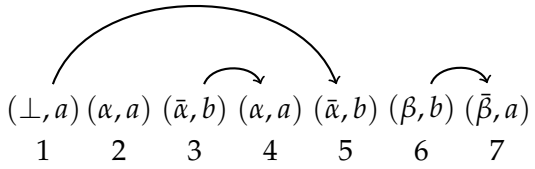


FIGURE 5.2 – Un élément de DyckMatch (ω, M) avec $\omega = \bot\alpha\bar{\alpha}\alpha\bar{\alpha}\beta\bar{\beta}$ et $M = \{(1,5),(3,4),(6,7)\}$ sur le mot *aababba*.

double de Greibach. Ceci permet de représenter une étape de dérivation par un couple dans une relation binaire. Les éléments de ce couple sont les positions du symbole terminal le plus gauche et du symbole terminal le plus à droite généré par une production en forme double.

La langages (grammaires) indexées ne bénéficient pas d'une telle propriété. Nous considérons alors dans un premier temps les langages indexés doubles. Ceux-ci sont générés par des grammaires indexés pour lesquelles chaque production d'empilement ou de dépilement produit deux symboles terminaux. Ceci permet à la fois de borner le nombre d'empilements et de dépilements effectués lors d'une dérivation par la taille du mot généré et de représenter chaque nœud d'empilement et de dépilement d'une dérivation par un couple d'éléments.

Nous montrons qu'un langage est indexé double si et seulement si il est définissable dans la logique ∃DyckMatchFO et si et seulement si il est définissable dans la logique ∃DyckMatchMSO. La classe DyckMatch ici est constituée de relations de parenthésage accompagné d'une relation d'étiquetage définissant un mot de Dyck (voir illustration figure 5.2). Cet étiquetage et ce parenthésage sont de plus liés par une condition supplémentaire.

Nous considérons deuxièmement un type précis de *projections* sur des structures relationnelles. Nous montrons qu'un langage est indexé si et seulement si il est la *projection* d'un langage définissable dans ∃DyckMatchFO (∃DyckMatchMSO). Nous montrons aussi que plus généralement, si on se restreint aux quantifications en considérant une logique $\exists L$ MSO où L est un langage ; alors la famille de langages définissables dans cette logique est précisément le cône croissant principal généré par le langage L. De plus, l'ensemble de projections des langages qui y sont définissables est le cône principal généré par L.

Plan Ce chapitre est sectionné comme suit :

— Dans la section 5.2, nous introduisons les définitions de base liées à la logique et aux caractérisations logiques.
— Dans la section 5.3, nous présentons les fragments sémantiques du second ordre ainsi que les arguments de la caractérisation de [LST94] et d'autres notions qui nous seront nécessaires dans notre généralisation.
— Dans la section 5.4, nous présentons les langages indexés doubles ainsi que la logique ∃DyckMatchFO.
— La section 5.5 est dédiée à la preuve de la caractérisation des indexés doubles dans ∃DyckMatchFO (∃DyckMatchMSO).
— Dans la section 5.6, nous discutons de l'usage de projections logiques pour caractériser des cônes principaux y compris la famille des indexés.

5.2 Logique du Second Ordre (Existentiel)

Dans cette section, nous présentons les définitions de base : structures, signatures, logiques du premier et second ordre.

5.2.1 Signatures et structures logiques

Définition 5.1 (Signatures)**.** *Une* signature *ou* vocabulaire *est un ensemble fini* $\mathcal{S}$ *de symboles* $R_1, \ldots, R_n$ *associé à des relations. L'arité de chaque symbole* R_i *est notée* $|R_i|$.

Définition 5.2 (Structures relationnelles)**.** *Soit* $\mathcal{S}$ *une signature. Une* $\mathcal{S}$*-structure est un couple* $(\mathcal{U}, \eta)$ *où* $\mathcal{U}$ *est appelé* univers *et* ι *est une* interprétation *des symboles de* $\mathcal{S}$ *: celle-ci associe à chaque symbole* R_i *une relation* $\iota(R_i) \subseteq \mathcal{U}^{|R_i|}$. *Pour la cohérence de la définition, l'univers* $\mathcal{U}$ *est considéré* non-vide.

Notations. Quand nécessaire, une structure est notée comme un uplet composé de l'univers et des interprétations des symboles relationnels : Pour une signature $\mathcal{S} = \{X_1, \ldots, X_n\}$, une $\mathcal{S}$-structure $(\mathcal{U}, \iota)$ est notée $(\mathcal{U}, R_1, \ldots, R_n)$ avec $R_i = \iota(X_i)$ pour tout X_i. Laissant de cette façon l'interprétation ι implicite.

5.2.2 Logique du premier ordre

Définition 5.3 (Logique du premier ordre (syntaxe))**.** *Soit* $\mathcal{S} = \{R_1, \ldots, R_m\}$ *une signature et* $\mathcal{V} = \{x_1, \ldots, x_n\}$ *un ensemble de* variables *dites du* premier ordre. *La logique du premier ordre sur* $\mathcal{S}$*, dénoté* $\mathrm{FO}(\mathcal{S})$*, est l'ensemble de formules* φ *de la forme*

$$\begin{aligned} R(x_1, \ldots, x_k) &\quad \text{avec } R \in \mathcal{S} \text{ d'arité } k \text{ et chaque } x_i \in \mathcal{V}, \\ \varphi \vee \varphi' &\quad \text{avec } \varphi, \varphi' \in \mathrm{FO}(\mathcal{S}), \\ \exists x : \varphi &\quad \text{avec } x \in \mathcal{V}, \varphi \in \mathrm{FO}(\mathcal{S}), \\ \neg\varphi &\quad \text{avec } \varphi \in \mathrm{FO}(\mathcal{S}). \end{aligned}$$

Formules closes. Une variable est x est dite *liée* dans une formule si cette formule de la forme $\exists x \phi$. Autrement, elle est dite *libre* (dans cette formule). Une formule sans variable libre est dite *close*.

Quand nécessaire, nous écrivons $\varphi(x_1, \ldots, x_n)$ pour préciser qu'une formule φ a précisément $x_1, \ldots, x_n$ comme variables libres.

Définition 5.4 (Logique du premier ordre (sémantique))**.** *Soit* $\mathcal{S}$ *une signature et* $\mathcal{V}$ *un ensemble de variables du premier ordre (des variables d'arité* 0*). Une valuation est une application de* $\mathcal{V}$ *dans* $\mathcal{U}$*. Pour toute formule* φ *ayant* $\mathcal{V}$ *comme ensemble de variables, on dit qu'une* $\mathcal{S}$*-structure* S *munie d'une valuation* v *satisfait* φ*, et on note* $(S, v) \models \varphi$*, si*

$$\begin{aligned} (S, v) &\models R(x_1, \ldots, x_k) && \text{si } \iota(R)(v(x_i), \ldots, v(x_k)) \\ (S, v) &\models \varphi \vee \varphi' && \text{si } (S, v) \models \varphi \text{ ou } (S, v) \models \varphi', \\ (S, v) &\models \exists x : \varphi && \text{si il existe } n \in \mathcal{U} \text{ et une valuation } v' \text{ telle que } (S, v') \models \varphi \\ & && \text{avec } v'(x) = n \text{ et } v'(y) = v(y) \ \ \forall y \in \mathcal{V} - \{x\}, \\ (S, v) &\models \neg\varphi && \text{si } (S, v) \text{ ne satisfait pas } \varphi. \end{aligned}$$

Notations et raccourcis.

— Une chaine de quantifications existentielles $\exists x_1 \cdots \exists x_n$ de longueur variable est notée en police gras : $\exists \mathbf{x}$.

— Etant donné une formule close ϕ et une structure S, on écrit $S \models \phi$ si quel que soit la valuation v, on a $(S, v) \models \phi$. En particulier la satisfiabilité d'une formule close $\exists \boldsymbol{x} \phi$ présume déjà l'existence d'une valuation satisfaisant ϕ : Quel que soit une valuation v, (S, v) satisfait $\exists \boldsymbol{x} \phi$ si il existe une valuation v' telle que $(S, v') \models \phi(\boldsymbol{x})$. Il n'est donc pas nécessaire de considérer v.

— Étant donné une signature $\mathcal{S}$ et une formule close ϕ, on note $\mathscr{L}(\phi)$ l'ensemble de structures S sur $\mathcal{S}$ telles que $S \models \phi$.

— Nous usons des raccourcis habituels :

$$\begin{aligned} \forall x \phi &\equiv \neg \exists x (\neg \phi), \\ \phi_1 \wedge \phi_2 &\equiv \neg(\neg \phi_1 \vee \neg \phi_2), \\ \phi_1 \Rightarrow \phi_2 &\equiv (\neg \phi_1) \vee \phi_2, \\ \phi_1 \Leftrightarrow \phi_2 &\equiv (\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_1). \end{aligned}$$

Mots et langages définissables. Soit A un alphabet. Nous notons $(A, <)$ la signature $\{<, (P_a)_{a \in A}\}$. Un mot $u = b_1 b_2 \cdots b_{|u|} \in A^+$ peut être codé comme une $(A, <)$-structure $(\mathcal{U}, \iota)$ notée $\underline{u}$ telle que

— $\mathcal{U} = \{1, \ldots, |u|\}$ (l'univers correspond à l'ensemble de positions dans le mot u ;

— $\iota(<)$ est l'ordre linéaire sur $\mathcal{U}$;

— pour tout $i \in \mathcal{U}$ et $a \in A : i \in \iota(P_a)$ si et seulement si $b_i = a$.

Un langage $L \subseteq A^+$ est dit *définissable* dans une logique si il existe une formule close ϕ dans cette logique telle que $L = \{u \in A^+ \mid \underline{u} \in \mathscr{L}(\phi)\}$.

Remarque 5.5. *En présence de la relation d'ordre $<$, nous pouvons user des raccourcis « $x > y, x \leq y, x = y, x \neq y$ » et des macros « $x + 1, \min, \max$ ».*

Exemple 5.6. *Le langage a^+b^+ est défini par la formule du premier ordre sur les mots suivante :*

$$\phi \equiv \exists x : \forall y : (y \leq x \Rightarrow P_a(y)) \wedge (y > x \Rightarrow P_b(y)).$$

5.2.3 Logique du second ordre et fragments

La logique du second ordre se définit de façon similaire à celle du premier ordre–à la différence qu'ici, les variables peuvent être d'arité quelconque.

Définition 5.7 (Logique du second ordre (SO))**.** *Soit $\mathcal{S}$ une signature, $\mathcal{V}_1$ un ensemble de variables du premier ordre et $\mathcal{V}_2$ un ensemble de variables du second ordre (des variables d'arité ≥ 1). La syntaxe des formules du second ordre sur $\mathcal{S}$ ayant $\mathcal{V}_1$ et et $\mathcal{V}_2$ ensemble de variables étend celle des formules du premier ordre sur $\mathcal{S}$ ayant $\mathcal{V}_1$ comme variables du premier ordre par les deux formes supplémentaires suivantes :*

$$\begin{aligned} \exists X \varphi \quad & \text{avec } X \in \mathcal{V}_2, \varphi \in \mathrm{SO}(\mathcal{S}), \\ X(t_1, \ldots, t_k) \quad & \text{avec } X \in \mathcal{V}_2 \text{ et chaque } t_i \in \mathcal{V}_1. \end{aligned}$$

La sémantique associée des formules du second ordre est une généralisation de celles du premier ordre comme suit : Soit $S = (\mathcal{U}, \iota)$ une $\mathcal{S}$-structure et v une valuation

— $(S, v) \models \exists X\, \varphi$ *avec* $X \in \mathcal{V}_2$ *si il existe une valuation* v' *telle que* $(S, v') \models \varphi$ *avec* $v'(X) \subseteq \mathcal{U}^{|X|}$ *et* $v'(Y) = v(Y)$ *pour tout* $Y \in \mathcal{V}_2 - \{X\}$;

— $(S, v) \models X(x_1, \ldots, x_n)$ *avec* $X \in \mathcal{V}_2$ *si* $v(X)(v(x_1), \ldots, v(x_n))$.

Définition 5.8 (Logique du Second Ordre Existentiel (∃SO))**.** *Soit* $\exists\mathrm{SO}(\mathcal{S})$ *l'ensemble de formule de* $\mathrm{SO}(\mathcal{S})$ *de la forme* $\exists\mathbf{X}\ \phi$ *où* $\mathbf{X}$ *est un vecteur de variable* X_i *du second ordre et* ϕ *une formule du premier ordre.*

Exemple 5.9. *Le langage de Dyck sur un alphabet* A *est défini par la formule du second ordre existentiel sur les mots suivante :*

$$\begin{aligned}\phi \equiv \exists R : \mathrm{Matching}(R) \wedge \forall x : (\exists y : R(x,y) \vee R(y,x)) \\ \wedge (\forall y : R(x,y) \Rightarrow \bigvee_{a \in A} (P_a(x) \wedge P_{\bar{a}}(y))).\end{aligned}$$

où $\mathrm{Matching}(R)$ *est la formule du premier ordre suivante :*

$$\begin{aligned}\forall x \forall y :& (R(x,y) \Rightarrow x < y) \\ & \wedge (\forall x' : (R(x',y) \Rightarrow x = x') \wedge (R(x,x') \Rightarrow y = x')) \\ & \wedge (\forall x' \forall y' : (R(x,y) \wedge R(x',y') \wedge x < x' < y) \Rightarrow x < y' < y).\end{aligned}$$

Définition 5.10 (Second Ordre Monadique)**.** *La logique du second ordre monadique–sur une signature* $\mathcal{S}$ *(*$\mathrm{MSO}(\mathcal{S})$*) est l'ensemble de formules de* $\mathrm{SO}(\mathcal{S})$ *dans lesquelles toute variable du second ordre est d'arité* 1.

Définition 5.11. *Soit* $\exists\mathrm{MSO}(\mathcal{S})$ *l'ensemble de formules de* $\mathrm{MSO}(\mathcal{S})$ *de la forme* $\exists\mathbf{X}\,\phi$ *où chaque* X_i *de* $\mathbf{X}$ *est d'arité* 1 *et* ϕ *est une formule du premier ordre.*

Théorème 5.1 (Büchi–Elgot–Trakhtenbrot [Bö0, Elg61, Tra61])**.** *Soit* $L \subseteq A^+$ *un langage. Les affirmations suivantes sont équivalentes.*

1. *Le langage* L *est régulier ;*
2. *Il existe une formule* $\phi \in \mathrm{MSO}(A, <)$ *telle que* $\mathscr{L}(\phi) = L$.
3. *Il existe une formule* $\phi \in \exists\mathrm{MSO}(A, <)$ *telle que* $\mathscr{L}(\phi) = L$.

Exemple 5.12. *Le langage* $\{a^{2n} \mid n \geq 1\}$ *est définissable par la formule* $\phi \in \exists\mathrm{MSO}(\{a\}, <)$ *suivante :*

$$\begin{aligned}\phi \equiv \exists X_0 \exists X_1 : \forall x : \big[& x < \max \Rightarrow ((X_0(x) \Rightarrow (\neg X_0(x+1) \wedge X_1(x+1))) \\ & \wedge (X_1(x) \Rightarrow (\neg X_1(x+1) \wedge X_0(x+1)))) \\ & \wedge (X_0(\min) \wedge X_1(\max))\big].\end{aligned}$$

Ici, la variable X_0 *définit l'ensemble des positions impaires du domaine, et la variable* X_1 *définit l'ensemble de positions paires du domaine. Il s'agit alors de dire que la dernière position du domaine est paire.*

5.3 Restrictions sémantiques du second ordre

Dans [LST94], les auteurs proposent l'usage de fragments sémantiques du second ordre existentiel pour la caractérisation de familles de langages. Ils définissent ces fragments de façon suivante :

Définition 5.13. *Soit A un alphabet et $\mathcal{B}$ une classe de relations. La logique $\exists\mathcal{B}$ FO$(A,<)$ est l'ensemble de formules (closes) de la forme $\phi \equiv \exists X\, \phi_0$ où X est une variable du second ordre, ϕ_0 une formule du premier ordre.*

La valuation de X est restreinte à $\mathcal{B}$: Un mot $u \in A^+$ satisfait la formule ϕ ci-dessus si et seulement si il existe une relation $R \in \mathcal{B}$ sur le domaine de S telle que le couple (S,R) satisfait ϕ_0.

Dans [LST94], les auteurs définissent de façon similaire la logique $\exists\mathcal{B}$ MSO$(A,<)$ composée cette fois de formules de la forme $\phi \equiv \exists X\, \phi_0$ où ϕ_0 est une formule MSO[5].

Définition 5.14. *Soit A un alphabet et $\mathcal{B}$ une classe de relations. Soit $\exists\mathcal{B}$ $\exists$MSO$(A,<)$ l'ensemble de formules (closes) de la forme $\phi \equiv \exists X\, \phi_0$ où X est une variable du second dont la valuation est restreinte à $\mathcal{B}$ et ϕ_0 est une formule* $\exists$MSO.

Plan de la section. Dans cette section, nous présentons dans un premier temps la caractérisation logique des langages algébriques prouvée dans [LST94] ainsi que ses principaux arguments que nous réemploierons plus tard pour caractériser la famille des langages indexés doubles. Nous présentons deuxièmement une approche alternative de la caractérisation de ALG dans $\exists$Match MSO.

5.3.1 Logiques pour les langages algébriques

Dans [LST94], les auteurs prouvent qu'un langage $L \subseteq A^+$ est algébrique si et seulement si il est définissable dans $\exists$Match$FO(A,<)$ si et seulement si il est définissable $\exists$Match MSO$(A,<)$ où la classe Match est constituée de relations binaires appelés *matching* (définition plus bas).

Définition 5.15 (matchings et mots imbriqués)**.** *Soit u un mot. Un* matching *sur u est une relation $M \subseteq \{1,\dots,|u|\}^2$ telle que pour tout $i,j,k,l \in \{1,\dots,|u|\}$:*

- *si $M(i,j)$ alors $i<j$ (M est compatible avec l'ordre) ;*
- *si $M(i,j)$, alors $M(k,j)$ implique $i=j$ et $M(i,k)$ implique $j=k$ (une position ne peut appartenir à deux couples distincts de M) ;*
- *si $M(i,j)$, alors $M(k,l)$ et $i<k<j$ implique $i<l<j$.*

Chaque couple (i,j) de M est appelé arc.

Un mot imbriqué est un couple (u,M) où u est un mot et M un matching sur u.

Exemple 5.16. De façon similaire à l'exemple 5.9, le langage de Dyck sur un alphabet A est défini par la formule dsuivante :

$$\phi \equiv \exists M : \forall x : (\exists y : M(x,y) \vee M(y,x)) \\ \wedge (\forall y : M(x,y) \Rightarrow \bigvee_{a\in A} (P_a(x) \wedge P_{\bar{a}}(y))).$$

5. Le fragment $\exists\mathcal{B}$ MSO$(A,<)$ n'est pas *stricto sensu* un fragment du second ordre existentiel puisqu'une formule MSO peut commencer par une quantification universelle sur une variable monadique. Cependant, il est souvent équivalent au fragment $\exists\mathcal{B}$ $\exists$MSO$(A,<)$ de la même façon que les logiques MSO et $\exists$MSO sont équivalentes sur le mots

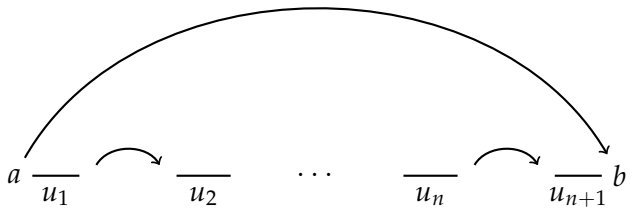


FIGURE 5.3 – Représentation d'une production dont le motif est $au_1|u_2|\cdots|u_{n+1}b$.

Il n'est plus nécessaire, comme c'était le cas dans l'exemple 5.9, d'explicitement restreindre la valuation de la variable du second ordre M à l'ensemble de matchings par une formule. Cette restriction est maintenant implicite.

Exemple 5.17. *Le langage* $\{a^n b^n \mid n \geq 0\}$ *peut être défini dans* $\exists\text{Match FO}(A, <)$ *par la formule* ϕ *suivante :*

$$\begin{aligned}\phi \equiv \exists M :&(\forall x \forall y : M(x,y) \Rightarrow P_a(x) \wedge P_b(y)) \\ &\wedge \exists x_0 : \forall x : (x \leq x_0 \Rightarrow \exists y : y > x_0 \wedge M(x,y)) \\ &\quad \wedge (x > x_0 \Rightarrow \exists y : y \leq x_0 \wedge M(y,x)).\end{aligned}$$

Preuve de la caractérisation

La preuve du fait que tout langage définissable dans $\exists\text{Match MSO}(A, <)$ est algébrique s'appuie sur le fait que les mots de feuilles de langages d'arbres MSO-définissables sont des langages algébriques [Don70, MW67, TW68]. Dans [LST94], les auteurs montrent alors que pour toute formule MSO sur $(A, <, M)$, il existe une formule MSO ϕ' sur une signature d'arbres telle que pour tout mot imbriqué (u, M), $(u, M) \models \phi$ si et seulement u est le mot de feuilles d'un arbre qui satisfait ϕ'. La preuve du fait que tout langage algébrique $L \subseteq A^+$ soit définissable dans $\exists\text{Match FO}(A, <)$ s'articule autours des arguments suivants :

Définition 5.18 (Motifs). *Soit* $p = X \longrightarrow u_1 Y_1 u_2 Y_2 \cdots Y_n u_{n+1}$ *une production d'une grammaire algébrique (chaque* u_i *est une suite de terminaux et* Y_i *un non-terminal). Le motif de la productions* p *est le mot* $\text{Patt}(p) = u_1|u_2|\cdots|u_{n+1}$.

Lemme 5.19. *Tout langage algébrique* $L \subseteq A^+$ *peut être généré par une grammaire algébrique* (N, A, P, S) *dans laquelle chaque production est sous l'une des formes suivantes*

$$(1) S \longrightarrow u \text{ avec } u \in A^+ \qquad (2) X \longrightarrow a\Theta b \text{ avec } a, b \in A, \Theta \in (N \cup A)^*$$

et dans laquelle si deux productions ont le même motif, alors elles ont le même membre gauche.

Les membres droits des productions de type (2) commencent et se terminent par un symbole terminal. Les positions de ces symboles sont associées à des arcs. Le motif $au_1|u_2|\cdots|u_{n+1}b$ d'une production p est représenté par un arc entre la position du symbole a et celle du symbole b et par des arcs séparant chaque u_i de u_{i+1} comme illustré sur la figure 5.3.

Il est alors possible de décrire le motif d'une production par une formule du premier ordre. Le fait que le membre gauche d'une production soit identifiable par son motif permet alors de décrire le fait qu'un sous-mot ait été généré depuis ce non-terminal. Ceci se fait par les formules du premier ordre qui suivent.

Motif. Soit une production p ayant comme motif $au_1|u_2|\cdots|u_{n+1}b$. La formule $\phi_p(x,y)$ ci-dessous exprime le fait que pour un couple (u,M), le sous-mot de u entre deux positions x et y et le matching M correspondent au motif de p :

$$\begin{aligned}\phi_p(x,y) \equiv\ & a(x) \wedge b(y) \wedge \exists x_1 \exists y_1 \cdots \exists x_n \exists y_n : (x < x_1 < y_2 < \cdots x_n < y_n < y) \\ & \wedge \big(\varphi_{au_1}(x,x_1) \wedge \varphi_{u_2}(y_1,x_2) \wedge \cdots \varphi_{u_{n+1}b}(y_n,y)\big) \wedge \\ & \bigwedge_{i=1}^{n} M(x_i+1, y_i-1)\end{aligned}$$

où $\varphi_u(x,y)$ est une formule qui est vraie si le sous-mot entre les positions x et y est u.

Non-terminal. La formule $\phi_X(x,y)$ ci-dessous exprime le fait que pour un couple (u,M), le sous-mot de u entre deux positions x et y et le matching M correspondent au motif d'une production d'un non-terminal X.

$$\phi_X(x,y) \equiv \bigvee_{p:X \longrightarrow \Theta} \phi_p(x,y).$$

Production. Soit une production $p = X \longrightarrow au_1Y_1u_2\cdots Y_nu_{n+1}b$. La formule $\widehat{\phi}_p(x,y)$ ci-dessous exprime le fait que pour un couple (u,M), le sous-mot de u entre les positions x et y et le matching M correspondent à une dérivation par p.

$$\begin{aligned}\widehat{\phi}_p(x,y) \equiv\ & a(x) \wedge b(y) \wedge \exists x_1 \exists y_1 \cdots \exists x_n \exists y_n : (x < x_1 < y_2 < \cdots x_n < y_n < y) \\ & \wedge \big(\varphi_{au_1}(x,x_1) \wedge \varphi_{u_2}(y_1,x_2) \wedge \cdots \varphi_{u_{n+1}b}(y_n,y)\big) \wedge \\ & \bigwedge_{i=1}^{n} \big(M(x_i+1, y_i-1) \wedge \phi_{Y_i}(x_i+1, y_i-1)\big)\end{aligned}$$

Langage généré. Le langage généré par la grammaire est alors défini par la formule $\phi \equiv \exists M\ \phi_0$ avec

$$\begin{aligned}\phi_0 \equiv \bigvee_{S \longrightarrow u \in P, u \in T^*} \varphi_u \vee \big[& M(\min,\max) \wedge \phi_X(\min,\max) \\ & \wedge (\forall (x,y) \in M : \bigvee_{p \in P} \widehat{\phi}_p(x,y))\big].\end{aligned}$$

5.3.2 Approche alternative

La caractérisation de ALG dans $\exists$Match FO$(A,<)$ emploie des arguments propres aux grammaires algébriques. Sa caractérisation dans $\exists$Match MSO$(A,<)$ quant à elle peut se déduire de travaux plus récents d'Alur et Madhusudan [AM09, AM04]. Ceux-ci étudient des langages de mots imbriqués reconnaissables par deux formalismes (équivalents) : les « nested word automata » et « visibly pushdown automata ». Ils proposent de plus une caractérisation logique de ces langages. La définissabilité de ALG dans $\exists$Match MSO se déduit de sa relation avec les langages reconnaissables par automates à pile visible ; qui coïncide avec la relation entre la logique que les auteurs proposent et $\exists$Match MSO.

Notations. Pour tout alphabet A, soit $\langle A = \{\langle a \mid a \in A\}$ et $A\rangle = \{a\rangle \mid a \in A\}$.

Définition 5.20. *Soient A_{int}, A_c et A_r trois alphabets disjoints. Un mot $u \in (A_{\text{int}} \cup A_c \cup A_r)^+$ est dit bien structuré vis-à-vis de $(A_{\text{int}}, A_c, A_r)$ si $f(u) \in \mathcal{D}_{\{a\}}$ où f est l'homomorphisme tel que $f(A_{\text{int}}) = \{\varepsilon\}$, $f(A_c) = \{a\}$ et $f(A_r) = \{\bar{a}\}$.*

Représentations des mots imbriqués. Soit A un alphabet. Un mot imbriqué (u, M) avec $u \in A^+$ peut être représenté comme un mot bien structuré $u' \in (A \cup \langle A \cup A\rangle)^+$ vis-à-vis de $(A, \langle A, A\rangle)$; et réciproquement. Le codage se fait comme suit : supposons $u = a_1 a_2 \cdots a_{|u|}$. Le mot u' est $a'_1 a'_2 \cdots a'_{|u|}$; et pour tout $i \in \{1, \ldots, |u|\}$, $a'_i = \langle b \in \langle A$ si $a_i = b$ et $(i, j) \in M$ pour un certain j ; $a'_i = b\rangle \in \langle A$ si $a_i = b$ et $(j, i) \in M$ pour un certain j ; $a'_i = b \in A$ si $a_i = b$ et $(i, j), (j, i) \notin M$ quelque soit j. De ce fait, il est alors possible de représenter des mots bien structurés vis-à-vis d'un triplet d'alphabets par les mêmes structures logiques que les mots imbriqués. Dans ce qui suit, nous confondons alors ces deux notions.

Les automates à pile visibles sont des automates à pile pour lesquels les instructions de pile effectuées sont conditionnées par le mot d'entrée. Ils se définissent formellement comme suit :

Définition 5.21 (Automates à pile visible)**.** *Un automate à pile visible est un automate à pile $\mathcal{A} = (Q, T, q_0, Z, \Delta, F)$.*

- *où T est un alphabet structuré $T = A_c \cup A_{\text{int}} \cup A_r$. L'alphabet A_c est appelé alphabet de* call*, A alphabet « interne » et A_r alphabet de* return*.*
- *Q, q_0, Z, Δ, F sont respectivement l'ensemble d'états, l'état initial, l'alphabet de pile, la relation de transition et l'ensemble d'états acceptants.*
- *La relation de transition Δ est partitionnée en $\Delta_c, \Delta_i, \Delta_r$*

$$\Delta_c \subseteq Q \times A_c \times Z \times Q; \quad \Delta_i \subseteq Q \times A_{\text{int}} \times \{\varepsilon\} \times Q; \quad \Delta_r \subseteq Q \times A_r \times \overline{Z} \times Q.$$

Lemme 5.22 (Relation avec ALG [AM09])**.** *Un langage $L \subseteq A^+$ est algébrique si et seulement si il existe un langage à pile visible $L' \subseteq (A \cup \langle A \cup A\rangle)^+$ tel que*

$$L = \{u \mid \exists M : (u, M) \in L'\}.$$

Proposition 5.23 ([AM09])**.** *Soit $L \subseteq (A \cup \langle A \cup A\rangle)^+$ un langage. Le langage L est reconnaissable par un automate à pile visible si et seulement si il est définissable (modulo représentation sous forme de mots imbriqués) par une formule $\mathrm{MSO}(A, <, M)$ sur la signature des mots imbriqués.*

Corollaire 5.24. *Un langage L est algébrique si et seulement si il est définissable dans $\exists\mathrm{Match}\,\mathrm{MSO}(A, <)$.*

En effet, d'après le lemme 5.22 et la proposition 5.23, un langage $L \subseteq A^+$ est algébrique si et seulement si il existe une formule MSO sur les mots imbriqués telle que

$$L = \{u \mid \exists M : (u, M) \models \phi\};$$

c'est-à-dire, si et seulement si il est définissable dans $\exists\mathrm{Match}\,\mathrm{MSO}(A, <)$.

5.4 Langages indexés doubles

Un fait des logiques de la forme $\exists\mathcal{B}$ FO et $\exists\mathcal{B}$ MSO est qu'une unique (un nombre fini de) quantification(s) de variables sémantiquement restreintes dans $\mathcal{B}$ est autorisée. Soulignons

de plus l'évidence que le domaine de cette relation est l'ensemble des positions du mot. Ceci implique intuitivement qu'un langage L ne peut être définissable dans une logique du type $\exists\mathcal{B}$ FO ou $\exists\mathcal{B}$ MSO que si pour tout mot u, la quantité d'information qui doit être devinée pour vérifier que ce mot appartient à ce langage ne doit être trop grande et doit pouvoir s'exprimer comme une relation $R \in \mathcal{B}$ sur le domaine $\{1, \ldots, |u|\}$.

Tout langage algébrique $L \subseteq A^+$ peut être généré par une grammaire en forme double : toute dérivation est sous l'une des formes

$$X \longrightarrow a\Theta b; \quad X \longrightarrow a.$$

Le fait que les grammaires algébriques soient *lexicalisables* (toute production produit un symbole terminal) a pour effet que la taille d'une dérivation générant un mot peut être bornée par la taille de ce mot. Nous avons vu par la preuve de [LST94] qu'en particulier, le fait que les grammaires algébriques peuvent être mises en forme double permet d'associer chaque élément de dérivation à un couple d'une relation binaire. Un résultat que nous considérons encore plus indicatif de la définissabilité des langages algébriques est que d'après [Okh12], tout langage algébrique peut s'exprimer comme $\tau(\mathcal{D}_k)$ où la transduction rationnelle τ est *croissante*. Nous abordons plus en détail la définissabilité des cônes croissants dans la section de discussion.

Les langages indexés quant à eux ne bénéficient pas de ces privilèges. La forme normale connue pour les grammaires indexées est

$$(1)X \longrightarrow YZ; \quad (2)X \longrightarrow Y^{\alpha}; \quad (3)X^{\alpha} \longrightarrow Y; \quad (4)X \longrightarrow a.$$

Sous cette forme, la génération d'un mot peut nécessiter un nombre non-borné de productions de type (1), (2) ou (3). Ceci nous amène à considérer une restriction des langages indexés que nous appelons *langages indexés doubles*.

5.4.1 Langages indexés doubles

Définition 5.25. *Une grammaire indexée (N, T, I, P, S) est dite en forme faiblement double si chacune de ses productions est sous l'une des formes suivantes*

$$(1)X \longrightarrow YZ; \quad (2)X \longrightarrow aY^{\alpha}b \text{ avec } a, b \in T, \alpha \in I \cup \{\varepsilon\};$$
$$(3)X^{\alpha} \longrightarrow aYb \text{ avec } a, b \in T, \alpha \in I \cup \{\varepsilon\}; \quad (4)X \longrightarrow u, u \in T^*.$$

Cette grammaire est dite de rang k si $|I| = k$. Un langage indexé est dit double et de rang k si il peut être généré par une grammaire en forme faiblement double de rang k.

Nous choisissons le terme faiblement double du fait que toutes les productions ne sont pas en forme double : les productions de type (1) et (4) ne le sont pas. En particulier, nous considérons cette « double lexicalisation » des productions d'empilement et de dépilement afin de pouvoir décrire ces nœuds de dérivations par un couple d'éléments. Nous pensons que cette restriction est plus forte que la restriction de lexicalisation. Par exemple, nous n'avons pu trouver de grammaire indexée double pour le langage de mots de la forme

$$a^n\$a\$a^2\$ \cdots \$a^{n-1}\$a^n \quad \text{avec } n \geq 1.$$

Celui-ci est pourtant généré par la grammaire

$$S \longrightarrow aS_0{}^{\$};$$
$$S_0{}^{\$} \longrightarrow \$; \qquad S_0 \longrightarrow aS_0{}^{\alpha} + aX^{\alpha};$$
$$X^{\$} \longrightarrow \$a; \qquad X^{\alpha} \longrightarrow XYa;$$
$$Y^{\$} \longrightarrow \$a; \qquad Y^{\alpha} \longrightarrow Ya.$$

Exemple 5.26. *Le langage de mots de la forme w^n avec $|w|, n > 1$ (afin de simplifier l'exemple) est un indexé double. La grammaire ci-dessous exploite le fait suivant : un mot, disons, $(abcde)^5$ peut se factoriser comme $ab(cdeab)^4cde$. Pour simplifier notre exemple, nous utilisons des productions de la forme $X \longrightarrow aY^{\alpha}bc$. Un telle production peut être substituée par $X \longrightarrow Y_0Z$ avec $Y \longrightarrow aYb$ et $Z \longrightarrow c$. La grammaire générant ce langage est alors la suivante :*

$$\begin{aligned}
S &\longrightarrow \sum_{a,b\in T} \left(aS_0{}^{(\$,a,b)}b + aY^{(\$,a,b)}b\right) + \sum_{a,b,c\in T} \left(aS_0^{(\$,a,b,c)}bc + aY^{(\$,a,b,c)}bc\right);\\
S_0 &\longrightarrow \sum_{a,b\in T} \left(aS_0{}^{(a,b)}b + aX^{(a,b)}b\right);\\
X &\longrightarrow YY + YX;\\
Y^{(a,b)} &\longrightarrow bYa; \qquad Y^{(\$,a,b)} \longrightarrow ba; \qquad Y^{(\$,a,b,c)} \longrightarrow bca.
\end{aligned}$$

Exemple 5.27. *Pour tout homomorphisme symétrique et alphabétique $g : \widehat{A}^* \longrightarrow \widehat{B}^*$, le langage $\mathcal{D}_A \cap g^{-1}(\mathcal{D}_B)$ est un indexé double de rang $|g(A) - \{\varepsilon\}|$ car il peut être généré par la grammaire indexée*

$$\begin{aligned}
S &\longrightarrow \varepsilon + SS,\\
S &\longrightarrow aS^{g(a)}\bar{a} \quad \text{pour tout } a \in A \text{ tel que } g(a) \in B \cup \{\varepsilon\},\\
S^{\overline{g(a)}} &\longrightarrow aS\bar{a} \quad \text{pour tout } a \in A \text{ tel que } g(a) \in \overline{B}.
\end{aligned}$$

5.4.2 Logique pour les indexés doubles

Nous présentons ici la logique $\exists\text{Dyck}_k\text{MatchFO}(A, <)$. Nous définissons celle-ci comme une extension de $\exists\text{MatchFO}(A, <)$ par une quantification existentielle supplémentaire sur une relation d'étiquetage (un ensemble de prédicats unaires). Celle-ci sera restreinte à définir un mot de Dyck et sera de plus sémantiquement liée à la relation de matching.

Définition 5.28 (Symbole neutre)**.** — *Soit $\mathcal{D}_{A,\perp}$ le langage de Dyck sur l'alphabet A étendu d'un symbole neutre « $\perp$ ».*
Nous notons de plus $\mathcal{D}_{k,\perp}$ le langage $\mathcal{D}_{A,\perp}$ pour un certain A tel que $|A| = k$. De même, $\mathcal{T}_{A,\perp}$ désigne l'extension de $\mathcal{T}_A$ par le symbole neutre $\perp$.
— *Pour tout alphabet A, nous notons $A_\perp$ l'alphabet $A \cup \{\perp\}$.*

Exemple 5.29. *Si $A = \{a, b\}$, alors $a\perp\bar{a}b\bar{b}\perp\perp \in \mathcal{D}_{A,\perp}$.*

Définition 5.30 (Produit lettre-à-lettre)**.** *Soit le produit $\odot$ qui à tout $u = a_1a_2\cdots a_n \in A^*$ et $v = b_1b_2\cdots b_n \in B^*$ associe le mot $u \odot v = (a_1, b_1)(a_2, b_2)\cdots(a_n, b_n) \in (A \times B)^*$.*
Pour tout langage L et M, $L \odot M$ désigne le langage

$$L \odot M = \{u \odot v \mid u \in L, v \in M\}.$$

Définition 5.31 (Mots Dyck-imbriqués)**.**
— *Soit $k \geq 1$. Un* matching de Dyck *d'ordre k est un couple (ω, M) tel que*
1. *$\omega \in \mathcal{D}_{k,\perp}$;*
2. *M est un matching sur ω ;*

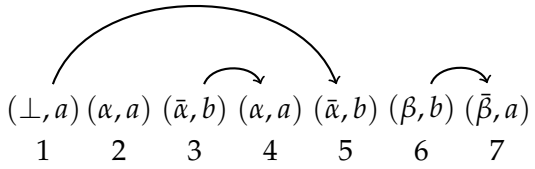


FIGURE 5.4 – Un mot Dyck-imbriqué d'ordre 2.

3. *pour tout arc* $(i,j) \in M$: *le sous-mot* $\omega[i;j] \in \mathcal{T}_{k,\perp}$.

— *Soit* DM_k *l'ensemble de matching de Dycks d'ordre k.*
— *Un mot Dyck-imbriqué d'ordre k est un couple* $(\omega \odot u, M)$ *tel que* $(\omega, M) \in DM_k$.

Exemple 5.32. Le couple (ω, M) avec $\omega = \perp\alpha\bar{\alpha}\alpha\bar{\alpha}\beta\bar{\beta}$ et $M = \{(1,5),(3,4),(6,7)\}$ est un matching de Dyck et le couple $(\omega \odot u, M)$ avec $u = aababba$ est un mot Dyck-imbriqué comme illustré sur la figure 5.4.

Définition 5.33 (Logique pour les indexés doubles)**.** *Soit A un alphabet et k un entier. Nous désignons par* $\exists\text{Dyck}_k\text{MatchFO}(A,<)$ *l'ensemble de formules de la forme* $\phi \equiv \exists(\mathbf{D}, X)\ \phi_0$ *où*

1. $\mathbf{D}$ *est une suite de* $2k+1$ *variables unaires sémantiquement restreintes à définir un mot : chaque position du domaine est étiquetée par un unique* D_i ;
2. *la variable X est binaire ;*
3. ϕ_0 *est une formule du premier ordre sur des mots Dyck-imbriqués.*

Le langage défini par la formule ϕ *est*

$$\mathscr{L}(\phi) = \{u \in A^+ : \exists(\omega, M) \in DM_k : (\underline{\omega \odot u}, M) \models \phi_0\}.$$

La quantification sur les variables (D_i) est inspirée des quantifications de Lindström [Lin66, BV98] sur les mots. Celles-ci consistent à restreindre les valuations possibles d'un ensemble de variables par un langage.

Exemple 5.34. Soit $g : \widehat{A}^* \longrightarrow \widehat{B}^*$ un homomorphisme symétrique et alphabétique. Le langage $(\mathcal{D}_A \cap g^{-1}(\mathcal{D}_B)) - \{\varepsilon\}$ peut être défini par la formule $\phi \equiv \exists(\mathbf{D}, M)\ \phi_0$ où ϕ_0 est la formule du premier ordre sur des mots Dyck-imbriqués suivante :

$$\begin{aligned}\phi_0(\mathbf{D}, M) \equiv \forall x :&(\exists y : M(x,y) \vee M(y,x)) \\ &\wedge \forall y : M(x,y) \Rightarrow (\bigvee_{a \in A} (P_a(x) \wedge D_{g(a)}(x) \wedge P_{\bar{a}}(y) \wedge D_{\overline{g(a)}}(y))).\end{aligned}$$

Note : Définissabilité des matching de Dycks. Du fait que les langages $\mathcal{D}_{k,\perp}$ et $\mathcal{T}_{k,\perp}$ soient $\exists$SO-définissables, et que la relation de Matching est FO-définissable (voir exemple 5.9), l'ensemble DM_k est $\exists$SO-définissable : Il existe une formule ϕ de $\exists$SO sur la signature $\{<, (P_a)_{a\in A}, M\}$ telle qu'une structure $(\underline{\omega}, \iota(M))$ satisfait ϕ si et seulement si $(\omega, \iota(M)) \in \text{DM}_{|A|}$. Celle-ci est

$$\begin{aligned}\phi \equiv \exists M_1 \exists M_2 :\ &\mathit{matching}(M_1) \wedge \mathit{matching}(M_2) \\ &\wedge \mathit{Dyck}(M_1) \wedge \mathit{Matching} \wedge \mathit{Bilateral}(M_2)\end{aligned}$$

— La formule *Matching* est une formule du premier ordre qui dit que l'interprétation du symbole relationnel M est un matching.

— Les formules $matching(X)$ expriment que leur variable libre X est un matching sur la totalité des positions étiquetées par un symbole différent de $\perp$.
— La formule $Dyck(M_1)$ est une formule du premier ordre qui dit l'interprétation des $(P_a)_{a\in A}$ définit un mot de $\mathcal{D}_{A,\perp}$ en supposant que la valuation de M_1 est un matching (si $M_1(x,y)$ alors $P_a(x)$ et $P_{\bar{a}}(y)$ pour un certain $a \in A$).
— La formule *Bilateral* dit que l'interprétation des $(P_a)_{a\in A}$ est un mot de $\mathcal{T}_{A,\perp}$ en supposant que l'interprétation de M_2 est un matching (si $M_2(x,y)$ alors $P_a(x)$ et $P_{\bar{a}}(y)$ pour un certain $a \in \widehat{A}$). Elle dit de plus que les arcs de M_2 et de M ne se croisent pas ($M(x,y) \wedge M_2(x',y') \Rightarrow \neg(x < x' < y < y' \vee x' < x < y' < y)$). La conjonction des deux exprime ainsi que le sous-mot entre chaque arc de M appartient à $\mathcal{T}$.

Nous énonçons notre théorème principal.

Théorème 5.2. *Un langage $L \subseteq A^+$ est un indexé double de rang k si et seulement si il est définissable dans $\exists\text{Dyck}_k\text{MatchFO}(A,<)$.*

La section 5.5 est dédiée à la preuve de ce théorème.

5.5 Preuve de la caractérisation logique

5.5.1 Transductions ε-sûres non-effaçantes

Afin de prouver que les langages indexés doubles sont les langages définissables dans la logique $\exists$DyckMatchFO, nous utilisons le même schémas de preuve que celui de [LST94] ; et pour cela, nous avons besoin d'une représentation des langages indexés doubles par des « grammaires algébriques ». Nous considérons alors la restriction suivante de transducteurs algébriques ε-sûrs générant les langages indexés doubles comme images du langage de Dyck.

Définition 5.35. *Un transducteur algébrique ε-sûr $(N, \widehat{A}, T, P, S)$ est dit non-effaçant si chacune de ses productions est sous l'une des formes suivantes :*

$$(1)X \longrightarrow YZ; \quad (2)X \longrightarrow (\alpha,a)Y(\bar{\alpha},b) \quad \textit{avec } \alpha \in \widehat{A} \cup \{\varepsilon\}, a,b \in T;$$
$$(3)X \longrightarrow (\varepsilon,u) \quad \textit{avec } u \in T^*.$$

Une transduction algébrique ε-sûre est dite non-effaçante si elle peut être générée par un transducteur algébrique ε-sûr non-effaçant.

Rappels. Rappelons (Définition 4.1) que IG_ET est la transformation qui à toute grammaire indexée en forme adéquate $\mathcal{I}$ associe un transducteur algébrique ε-sûr $\text{IG\_ET}(\mathcal{I})$ tel que

$$\mathscr{L}(\mathcal{I}) = \mathcal{R}(\text{IG\_ET}(\mathcal{I}))(\mathcal{D}_{|I|}).$$

Celle-ci transforme toute production de la forme

$$X \longrightarrow u\Theta^\alpha v \text{ avec } u,v \in T^*, \alpha \in I \cup \{\varepsilon\}, \Theta \in N^*$$
$$\text{en } X \longrightarrow (\alpha,u)\Theta(\bar{\alpha},v).$$

Similairement toute production $X^\alpha \longrightarrow u\Theta v$ est transformée en $X \longrightarrow (\bar{\alpha},u)\Theta(\alpha,v)$. Cette construction est de plus une bijection entre l'ensemble des grammaires indexées en forme adéquate et l'ensemble des transducteurs algébriques ε-sûrs en forme adéquate. Pour toute grammaire indexée $\mathcal{I}$ en forme adéquate, nous faisons référence à $\text{IG\_ET}(\mathcal{I})$ comme étant le *transducteur associé* à $\mathcal{I}$.

Lemme 5.36. *La transformation* IG_ET *est bijective entre l'ensemble de grammaires indexées en forme faiblement double et l'ensemble des transducteurs algébriques ε-sûrs non-effaçants.*

Démonstration. Naturellement, les production d'une grammaire indexée en forme faiblement double sont de la forme

$$(1)X \longrightarrow YZ; \quad (2)X \longrightarrow aY^{\alpha}b \text{ avec } a,b \in T, \alpha \in I \cup \{\varepsilon\};$$
$$(3)X^{\alpha} \longrightarrow aYb \text{ avec } a,b \in T, \alpha \in I \cup \{\varepsilon\}; \quad (4)X \longrightarrow u, u \in T^*.$$

Les productions du transducteur associé à une telle grammaire sont donc de la forme

$$(1)X \longrightarrow YZ; \quad (2)X \longrightarrow (\alpha,a)Y^{\alpha}(\bar{\alpha},b) \text{ avec } a,b \in T, \alpha \in \widehat{I} \cup \{\varepsilon\};$$
$$(3)X \longrightarrow (\varepsilon,u), u \in T^*.$$

□

Ceci a pour conséquence le lemme suivant

Lemme 5.37. *Soit $L \subseteq A^*$ un langage et k un entier. Le langage L est un indexé double de rang k si et seulement si il existe une transduction algébrique ε-sûre non-effaçante τ telle que $L = \tau(\mathcal{D}_k)$.*

Puisque nous ne parlerons que de langages inclus dans A^+, considérons les lemmes suivants.

Lemme 5.38. *Tout langage indexé double $L \subseteq A^+$ peut être généré par une grammaire indexée double dans laquelle pour toute production de la forme $X \longrightarrow u$, on a $u \in A^+$.*

Démonstration. Puisque $\varepsilon \notin L$, le langage L est généré par une grammaire indexée $\mathcal{I}$ en forme faiblement double dans laquelle pour toute production de la forme $X \longrightarrow \varepsilon$, on a $X \neq S$; S étant l'axiome de $\mathcal{I}$. Ces productions peuvent être supprimées par substitutions : Pour toute production $X \longrightarrow \varepsilon$ et pour toute production $Y \longrightarrow \Theta X \Theta'$, ajouter $Y \longrightarrow \Theta\Theta'$; puis supprimer la production $X \longrightarrow \varepsilon$. Les productions de la grammaire obtenue par ce procédé sont maintenant de la forme

$$(1)X \longrightarrow YZ; \quad (1')X \longrightarrow Y; \quad (2)X \longrightarrow aY^{\alpha}b \text{ avec } a,b \in T, \alpha \in I \cup \{\varepsilon\};$$
$$(3)X^{\alpha} \longrightarrow aYb \text{ avec } a,b \in T, \alpha \in I \cup \{\varepsilon\}; \quad (4)X \longrightarrow u, u \in T^+.$$

Il suffit maintenant de supprimer les productions de la forme $X \longrightarrow Y$ par substitutions. □

Lemme 5.39. *Soit $L \subseteq A^+$ un langage indexé double de rang k, pour un certain k. Il existe un transducteur algébrique non-effaçant $\mathcal{G}$ tel que $\mathcal{R}(\mathcal{G})(\mathcal{D}_k) = L$ et dans lequel pour toute production de la forme $X \longrightarrow (\varepsilon, u)$, on a $u \in A^+$.*

Démonstration. D'après le lemme 5.38, L peut être généré par une grammaire indexée en forme faiblement double et dans laquelle pour toute production de la forme $X \longrightarrow u$, on a $u \in A^+$. Pour toute production de la forme $X \longrightarrow (\varepsilon, u)$ dans le transducteur associé, on a alors $u \in A^+$. □

5.5.2 Définissabilité des indexés doubles

Approche. Nous montrons dans ce qui suit que tout langage indexé double $L \subseteq A^+$ est définissable dans $\exists \mathrm{Dyck}_k\mathrm{MatchFO}(A, <)$. Comme attendu, l'idée ici est que la relation de matching de Dyck permet de décrire les dérivations d'un transducteur ε-sûr non-effaçant pour lequel le domaine est restreint au langage de Dyck. De même que pour la caractérisation de Lautemann *et al* , la possibilité de décrire ces dérivations par des formules du premier ordre nécessitera de pouvoir identifier le membre gauche d'une production par la forme de son membre droit. Pour ce faire, nous montrons par les trois lemmes suivants que pour tout langage indexé double $L \subseteq A^+$ de rang k, il existe une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ avec $T = I_\perp \times A$ et $|I| = k$, et telle que

— $L = \pi_2(\mathscr{L}(\mathcal{G}) \cap \pi_1^{-1}(\mathcal{D}_{k,\perp}))$;
— pour tout non-terminal $X \in N$, $\mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$;
— toute production est sous l'une des formes suivantes :

$$S \longrightarrow u \text{ avec } u \in T^+ \; ; \quad X \longrightarrow t_1 \Theta t_2 \text{ avec } t_1, t_2 \in T;$$

— et si deux productions ont le même motif, alors elles ont le même membre gauche.

Une fois obtenue une telle grammaire, il nous suffira d'appliquer la construction de Lautemann *et al* pour obtenir la formule du premier ordre ϕ_0, sur des mots imbriqués décrivant les dérivations de la grammaire $\mathcal{G}$. En considérant notre restriction sémantique sur les mots Dyck-imbriqués, la formule ϕ de $\exists \mathrm{Dyck}_{k,A}\mathrm{MatchFO}$ décrivant le langage L sera tout simplement $\phi \equiv \exists(\mathbf{D}, M)\ \phi_0$. Commençons par prouver les deux premiers points.

Lemme 5.40. *Un langage $L \subseteq A^+$ un langage indexé double de rang k si et seulement si il existe une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ avec $T = \widehat{I}_\perp \times A$ et $|I| = k$ et telle que*

1. $L = \pi_2(\mathscr{L}(\mathcal{G}) \cap \pi_1^{-1}(\mathcal{D}_{I,\perp}))$;
2. *pour tout $X \in N$, $\mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$;*
3. *et chaque production est sous l'une des formes suivantes*

$$X \longrightarrow YZ; \quad X \longrightarrow t_1 Y t_2 \text{ avec } t_1, t_2 \in T; \quad X \longrightarrow u, u \in T^+.$$

Démonstration. ($\Rightarrow$) D'après le lemme 5.39, il existe un transducteur algébrique ε-sûr non-effaçant $\mathcal{G}_0 = (N, \widehat{I}, A, P, S)$ avec $|I| = k$, tel que $\mathcal{R}(\mathcal{G})(\mathcal{D}_I) = L$ et dans lequel chaque production est sous l'une des formes

$$(1) X \longrightarrow YZ; \quad (2) X \longrightarrow (\alpha, a) Y (\bar{\alpha}, b) \text{ avec } \alpha \in \widehat{I} \cup \{\varepsilon\}, a, b \in A;$$
$$(3) X \longrightarrow (\varepsilon, u) \text{ avec } u \in A^+.$$

La grammaire $\mathcal{G} = (N, \widehat{I_\perp} \times A, P', S)$ désirée décrit les couples de $\mathcal{G}_0$ comme des produits synchronisés sur l'alphabet $\widehat{I_\perp} \times A$. Elle est construite comme suit :

1. Ajouter toute production $X \longrightarrow YZ \in P$ à P' ;
2. pour toute production $X \longrightarrow (\varepsilon, u) \in P$, ajouter $X \longrightarrow (\perp, u_1)(\perp, u_2) \cdots (\perp, u_{|u|})$ à P' (chaque u_i désigne ici le i-ème symbole de u) ;
3. pour tout $X \longrightarrow (\alpha, a) Y (\bar{\alpha}, b) \in P$, ajouter $X \longrightarrow (\alpha, a) Y (\bar{\alpha}, b)$ à P si $\alpha \neq \varepsilon$; sinon, ajouter $X \longrightarrow (\perp, a) Y (\perp, b)$.

La grammaire obtenue satisfait les conditions du lemme.

Le sens ($\Leftarrow$) est trivialement vrai d'après le lemme 5.39. □

Nous souhaitons maintenant transformer notre grammaire de sorte que toute production non-terminale soit en forme double. La mise sous forme normale double quadratique de Greibach d'une grammaire algébrique exploite la régularité des ensemble de phrases de dérivations gauches et droites (voir [ABB97]). Nous souhaitons juste nous assurer que celle-ci préserve la propriété

$$\text{« pour tout non-terminal } X \in N, \mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+ \text{ ».}$$

Lemme 5.41. *Soit $L \subseteq A^+$ un indexé double de rang k. Il existe une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ avec $T = \widehat{I}_\perp \times A$ et $|I| = k$ telle que*

1. $L = \pi_2(\mathscr{L}(\mathcal{G}) \cap \pi_1^{-1}(\mathcal{D}_{I,\perp}))$;
2. *pour tout non-terminal X, $\mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$;*
3. *et toute production est sous l'une des formes suivantes*

$$X \longrightarrow t_1 \Theta t_2 \text{ avec } t_1, t_2 \in T; \quad X \longrightarrow u \text{ avec } u \in T^+.$$

Démonstration. Prenons comme point de départ le lemme 5.40. Nous avons une grammaire algébrique $\mathcal{G}_0 = (N, T, P, S)$ satisfaisant les points 1 et 2 du lemme et dans laquelle chaque production est sous l'une des formes suivantes

$$(1) X \longrightarrow YZ; \ \ (2) X \longrightarrow t_1 Y t_2 \text{ avec } t_1, t_2 \in T; \ \ (3) X \longrightarrow u \text{ avec } u \in T^+$$

Nous construisons dans un premier temps une grammaire $\mathcal{G}_1$ dans laquelle toutes les productions de la forme (1)– qui sont les seules à ne pas générer de terminal–seront transformées en productions de la forme $X \longrightarrow t\Theta$. Soit l'ensemble

$$\mathit{Term} = \{u \in T^+ \mid \exists X \longrightarrow u \in P\} \cup \{t_1 Y t_2 \in TNT \mid \exists X \longrightarrow t_1 Y t_2 \in P\};$$

et pour tout $X \in N, t \in \mathit{Term}$:

$$\text{soit } L_{X,t} = \{\Theta \in N^* \mid \exists \Theta_0 \in N^* : X \underset{g,}{\overset{*}{\Longrightarrow}} \Theta_0 \underset{g,}{\Longrightarrow} t\Theta \text{ et } t \text{ n'est pas préfixe de } \Theta_0\}.$$

$$\text{et } R_{X,t} = \{\Theta \in N^* \mid \exists \Theta_0 \in N^* : X \underset{d,}{\overset{*}{\Longrightarrow}} \Theta_0 \underset{d,}{\Longrightarrow} t\Theta \text{ et } t \text{ n'est pas suffixe de } \Theta_0\}.$$

Remarque 5.42.
— *Par définition de $\mathcal{G}_0$, pour tout $t \in \mathit{Term}$: $\pi_T(t) \in \mathcal{T}_{I,\perp} \odot A^+$.*
— *Puisque chaque $L_{X,t} \subseteq N^*$, alors pour tout $\Theta \in L_{X,t}$: $\mathscr{L}_{\mathcal{G}_0}(\Theta) \subseteq \mathcal{T}_{I,\perp} \odot A^+$.*
— *pour tout non-terminal X, nous avons $X = \bigcup_{t \in \mathit{Term}} \mathscr{L}(t)\mathscr{L}(L_{X,t})$.*

D'après le théorème 2.2, pour tout $X \in N$ et $t \in \mathit{Term}$, le langage $L_{X,t} \subseteq N^*$; et par conséquent a un nombre fini de quotients gauches et droits par N^*. Soit alors $\mathcal{F}_0$ la fermeture de $\{L_{X,t} \mid X \in N, t \in \mathit{Term}\}$ par quotient gauche et droit avec N, et $\mathcal{F} = \mathcal{F}_0 - \emptyset$. Chaque langage $L \in \mathcal{F}$ sera représenté par un non-terminal du même nom. La grammaire $\mathcal{G}_1 = (N', T, P', S)$ désirée est construite comme suit : $N' = N \cup \{L \in \mathcal{F}\}$;
— $X \longrightarrow tL_{X,t} \in P'$ pour tout $t \in \mathit{Term}$ tel que $L_{X,t} \neq \emptyset$;
— $L \longrightarrow XL' \in P'$ pour tout $L, L' \in \mathcal{F}, X \in N$ tels que $L' = X \cdot^{-1} L$;
— $L \longrightarrow \varepsilon \in P'$ pour tout $L \in \mathcal{F}$ tel que $\varepsilon \in L$.

À ce stade, chaque production de la grammaire $\mathcal{G}_1$ est sous l'une des formes

$$(1)X \longrightarrow tL_{X,t} \text{ avec } t \in \mathit{Term};\ \ (2)L \longrightarrow XL';\ \ (3)L \longrightarrow \varepsilon.$$

Les productions de type (2) et (3) assurent que chacune des nouvelles variables L génère bien la suite de non-terminaux qui lui est associée. Celles-ci correspondent aux productions des grammaires linéaires droites pour chacun de ces langages régulier $L \in N^*$. Les productions de type (1) quant à elles assurent que le langage généré par chaque non-terminal X reste inchangé. Nous pouvons donc émettre l'affirmation suivante.

Affirmation . *Pour tout* $X \in N \cup \mathcal{F} : \mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^*$.

Il nous suffit maintenant de substituer les non-terminaux X par $tL_{X,t}$ dans les membres droits des productions de la forme (2) ; puis d'éliminer les ε-productions de la forme (3). On obtient ainsi une grammaire dans laquelle chaque production est sous l'une des formes

$$\begin{aligned} X &\longrightarrow t\Theta \text{ avec } t \in \mathit{Term}, \Theta = L_{X,t} \text{ ou } \Theta = \varepsilon; \\ L &\longrightarrow tL_{X,t}L' \text{ avec } L \in \mathcal{F}, t \in \mathit{Term}, X \in N, L' \in \mathcal{F} \cup \{\varepsilon\}. \end{aligned}$$

De plus, pour tout $X \in \mathcal{F} \cup N, \mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$.

La grammaire $\mathcal{G}$ en forme double que nous souhaitons est obtenue en ajoutant un terminal droit aux productions. Cela se fait de façon similaire, à la difference que cette fois, il s'agit de considérer les langages $R_{X,t}$. Pour toute production de la forme $L \longrightarrow tL_{X,t}L'$, ou de la forme $X \longrightarrow tL'$, le non-terminal $L' \in \mathcal{F}$ est substitué par $L''R_{X,t}t$ tel que $L'' = L' \cdot X^{-1}$ et $R_{X,t} \neq \emptyset$. Des productions de la forme $L \longrightarrow \varepsilon$ seront aussi ajoutées, puis supprimées. □

Finalement, nous montrons qu'il est possible de transformer nos grammaires de sorte que deux productions ayant deux membres gauches distincts ne puissent avoir le même motif. La construction est tirée de [LST94]. Et celle-ci ne compromettra pas les autres critères que doivent remplir nos grammaires. Elle se fait comme suit.

Lemme 5.43. *Soit* $L \subseteq A^+$ *un langage indexé double de rang* k. *Il existe une grammaire algébrique* $\mathcal{G} = (N, T, P, S)$ *avec* $T = I_\perp \odot A$ *et* $|I| = k$ *telle que*

1. $L = \pi_2(\mathscr{L}(\mathcal{G}) \cap \pi_1^{-1}(\mathcal{D}_{I,\perp}))$;
2. *pour tout non-terminal* X, $\mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$;
3. *et toute production est sous l'une des formes suivantes*

$$(1)X \longrightarrow t_1\Theta t_2 \text{ avec } t_1, t_2 \in T;\ \ (2)X \longrightarrow u \text{ avec } u \in T^+$$

4. *si deux productions ont le même motif, alors elles ont le même membre gauche.*

Démonstration. Prenons comme point de départ une grammaire telle que celle du lemme 5.41. Celle-ci respecte les trois premiers critères. Le dernier s'obtient par la construction suivante.

1. Supprimer par substitutions toute production terminale $X \longrightarrow t$ avec $t \in T$ et $X \neq S$;
2. Énumérer les non-terminaux de 1 à $|N|$.
3. Pour tout $i \in \{2, \ldots, |N|\}$: tant qu'il existe une production $p : X_i \longrightarrow \Theta$ du même motif qu'une production $X_j \longrightarrow \Theta'$ avec $j < i$: remplacer p par toute production que l'on puisse obtenir par substitutions des non-terminaux dans Θ.

La procédure se termine puisque chaque substitution soit rend une production terminale, soit augmente sa taille et la rend de motif différent que ceux des autres productions. De plus, le langage généré par la grammaire reste inchangé. □

Proposition 5.44. *Tout langage indexé double $L \subseteq A^+$ de rang k est définissable par une formule de* $\exists \mathrm{Dyck}_k\mathrm{MatchFO}(A, <)$.

Démonstration. Soit $L \subseteq A^+$ un langage indexé double de rang k. Considérons pour L une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ avec $T = \widehat{I}_\perp \times A$ et $|I| = k$, selon les critères du lemme 5.43. D'après la construction de Lautemann *et al* , il existe une formule du premier ordre ϕ sur les mots imbriqués telle que pour tout mot $u \in T^+$

$$u \in \mathscr{L}(\mathcal{G}) \Leftrightarrow \exists M : (u, M) \models \phi.$$

De plus, pour tout mot imbriqué (u, M) tel que $(u, M) \models \phi$ et pour tout $(i, j) \in M$, il existe $X \in N$ tel que $u[i; j] \in \mathscr{L}(X)$. Du fait que $\mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$ pour tout $X \in N$, si $\pi_1(u) \in \mathcal{D}_{I,\perp}$ alors, $(\pi_1(u), M) \in \mathrm{DM}_k$. Par conséquent, pour tout mot imbriqué (u, M),

$$(u, M) \models \phi \wedge (\pi_1(u), M) \in \mathrm{DM}_k \Leftrightarrow (u, M) \models \phi \wedge \pi_1(u) \in \mathcal{D}_{I,\perp}.$$

Nous obtenons de ce fait

$$\begin{aligned} L &= \{\pi_2(u) \mid u \in \mathscr{L}(\mathcal{G}) \wedge \pi_1(u) \in \mathcal{D}_{I,\perp}\} \\ &= \{\pi_2(u) \mid \exists M : (u, M) \models \phi \wedge (\pi_1(u), M) \in \mathrm{DM}_k\} \\ &= \{v \in A^+ \mid \exists M : \exists \omega \in \widehat{I}_\perp^+ : (\omega, M) \in \mathrm{DM}_k \wedge (\omega \odot v, M) \models \phi\} \end{aligned}$$

Le langage L est alors défini par la formule $\exists(\mathbf{D}, M)\ \phi$. Ceci conclut la preuve de la proposition. □

5.5.3 Logique vers grammaires

Approche. Nous souhaitons maintenant montrer que tout langage définissable dans la logique $\exists \mathrm{Dyck}_k\mathrm{MatchFO}(A, <)$ est un indexé double de rang k. L'approche ici est la suivante. Étant donné une formule $\phi \equiv \exists(\mathbf{D}, M)\ \phi_0$ de $\exists \mathrm{Dyck}_k\mathrm{MatchFO}(A, <)$, le langage défini par ϕ est

$$\mathscr{L}(\phi) = \{u \in A^+ \mid \exists(\omega, M) \in \mathrm{DM}_k : \underline{(\omega \odot u, M)} \models \phi_0\}.$$

Considérons le langage

$$\begin{aligned} L_0 = \{(\omega \odot u) \mid & \exists M : (\omega \odot u, M) \models \phi \\ & \wedge \forall (i, j) \in M : \omega[i; j] \in \mathcal{T}_{k,\perp}\}. \end{aligned}$$

Le langage $\mathscr{L}(\phi)$ peut alors s'exprimer comme

$$\mathscr{L}(\phi) = \pi_2(L_0 \cap \pi_1^{-1}(\mathcal{D}_{k,\perp})).$$

Pour prouver que $\mathscr{L}(\phi)$ est un indexé double de rang k, il nous suffira donc de prouver que le langage L_0 peut être généré par une grammaire algébrique telle que pour tout non-terminal X, $\mathscr{L}(X) \subseteq \mathcal{T}_{k,\perp} \odot A^+$. Nous employons comme point de départ le fait que tout langage de mots imbriqués définissable par une formule MSO peut être reconnu par un automate à pile visible : le langage $\{(\omega \odot u, M) \mid (\omega \odot u, M) \models \phi\}$ est reconnaissable par un automate à pile visible.

Proposition 5.45. *Tout langage définissable dans* $\exists\mathrm{Dyck}_k\mathrm{MatchFO}(A,<)$ *est un indexé double de rang k.*

Démonstration. Soit une formule $\phi \equiv \exists(\mathbf{D}, M)\ \phi_0$. D'après la proposition 5.23, Le langage de mots imbriqués

$$\{(\omega \odot u, M) \mid \underline{(\omega \odot u, M)} \models \phi_0\}$$

est reconnu par un automate à pile visible $\mathcal{A} = (Q, T \cup \langle T \cup T\rangle, q_0, Z, \Delta, F)$. avec $T = I_\perp \times A$ et I de taille k. Considérons alors le langage

$$L_0 = \{(\omega \odot u) \mid \omega \in \mathcal{T}_{k,\perp}, \exists M : (\omega \odot u, M) \in \mathscr{L}(\phi_0) \wedge \forall (i,j) \in M : u[i;j] \in \mathcal{T}_{k,\perp}\}.$$

Nous construisons à partir $\mathcal{A}$ un automate à pile $\mathcal{A}_2 = (Q', T, q_0, Z \cup I \cup I', \Delta', F)$ reconnaissant le langage L_0. Avec, $I' = \{x' \mid x \in I\}$, $Q' \subseteq Q \cup Q \times Q$ et la relation de transitions Δ' est construite comme suit.

- Pour toute transition de Δ de la forme $(q, (\perp, a), z, p)$ ou $(q, \langle(\perp, a), z, p)$ ou $(q, (\perp, a)\rangle, z, p)$; ajouter $(q, (\perp, a), z, p)$ à Δ'.

- *Pour toute transition call* $(q, \langle(\alpha, a), z, p) \in \Delta$ avec $\alpha \in I$, ajouter les transitions $(q, \varepsilon, z, (q,p)_c)$ et $((q,p)_c, (\alpha, a), \alpha, p)$ à Δ'.
  De même, pour toute transition call $(q, \langle(\bar\alpha, a), z, p) \in \Delta$ avec $\alpha \in I$, ajouter les transitions $(q, \varepsilon, z, (q,p)_c)$ et $((q,p)_c, (\bar\alpha, a), \alpha', p) \in \Delta$.

- *Pour toute transition return* $(q, (\bar\alpha, a)\rangle, \bar z, p) \in \Delta$ avec $\alpha \in \Gamma$, ajouter les deux transitions $(q, (\bar\alpha, a), \bar\alpha, (q,p)_r)$ et $((q,p)_r, \varepsilon, \bar z, p)$ à Δ'.
  De même, pour toute transition return $(q, (\alpha, a)\rangle, \bar z, p) \in \Delta$ avec $\alpha \in I$, ajouter les transitions $(q, (\alpha, a), \bar\alpha', (q,p)_r)$ et $((q,p)_r, \varepsilon, \bar z, p)$ à Δ'.

- *Pour toute transition interne* $(q, (\alpha, a), \varepsilon, p) \in \Delta$ avec $\alpha \in I$, ajouter les transitions $(q, (\alpha, a), \alpha, p)$ et $(q, (\alpha, a), \bar\alpha', p)$ à Δ'.
  De même pour toute transition interne $(q, (\bar\alpha, a), \varepsilon, p) \in \Delta$ avec $\alpha \in I$, ajouter les transitions $(q, (\bar\alpha, a), \bar\alpha, p)$ et $(q, (\bar\alpha, a), \alpha', p)$ à Δ'.

Du fait que les actions de pile de $\mathcal{A}_2$ correspondent aux symboles lus, nous pouvons émettre l'affirmation suivante.

Affirmation 1. Pour tout calcul $(q, \omega \odot u, \perp) \underset{\mathcal{A}_2}{\vdash}^+ (p, \varepsilon, \perp)$, *on a* $\omega \in \mathcal{T}_{I,\perp}$.

Ceci implique en particulier que pour tout calcul

$$(q, \omega \odot u, \perp) \underset{\mathcal{A}_2}{\vdash} ((q,p)_c, \omega \odot u, z\perp) \underset{\mathcal{A}_2}{\vdash}^+ ((q',p')_r, \varepsilon, z\perp) \underset{\mathcal{A}_2}{\vdash} (p', \varepsilon, \perp)$$

avec $z \in Z$, on a $\omega \odot u \in \mathcal{T}_{I,\perp} \odot A^+$. Par définition de $\mathcal{A}_2$ (vis-à-vis de $\mathcal{A}$), nous pouvons alors émettre l'affirmation suivante

Affirmation 2. Soit $\omega \odot u \in \widehat{I}^+_\perp \odot A^+$ *et* $q, p \in Q$. *Il existe un calcul* $(q, \omega \odot u, \perp) \underset{\mathcal{A}_2}{\vdash}^+ (p, \varepsilon, \perp)$ *si et seulement si il existe un matching* M *tel que* $\omega \odot u[i;j] \in \mathcal{T}_{I,\perp}$ *pour tout* $(i,j) \in M$ *et un calcul* $(q, (\omega \odot u, M), \perp) \underset{\mathcal{A}}{\vdash}^+ (p, \varepsilon, \perp)$.

L'affirmation ci-dessus implique en particulier qu'un mot $(\omega \odot u)$ appartient à $\mathscr{L}(\mathcal{A}_2)$ si et seulement il existe un matching M tel que $(\omega \odot u, M)$ appartient à $\mathscr{L}(\mathcal{A})$ et $\omega \odot u[i;j] \in \mathcal{T}_{I,\perp}$

pour tout $(i,j) \in M$. Nous obtenons alors

Affirmation 3. $\mathscr{L}(\mathcal{A}_2) = L_0$.

Nous pouvons maintenant construire une grammaire algébrique $\mathcal{G} = (N, T, P, S)$ telle que $\mathscr{L}(\mathcal{G}) = \mathscr{L}(\mathcal{A}_2)$ et telle que pour tout $X \in N$: $\mathscr{L}(X) \subseteq \mathcal{T}_{k,\perp} \odot A^+$. Il s'agit de la conversion classique d'automates à pile en grammaires algébriques. Pour toute paire d'états $q, p \in Q'$ considérons l'ensemble

$$L_{q,p} = \{u \in T^+ \mid (q, u, \perp) \underset{\mathcal{A}_2}{\vdash}^+ (q, \varepsilon, \perp)\}.$$

Il est clair que $\mathscr{L}(\mathcal{A}_2) = \bigcup_{f \in F} L_{q_0,f}$. Chaque non-terminal de $\mathcal{G}$ est associé à un de ces langages $L_{q,p}$. L'ensemble de productions de productions P est construit comme suit. Pour toute paire d'états $q, p \in Q'$:

— pour toute transition $(q, a, \varepsilon, p) \in \Delta'$: ajouter $L_{q,p} \longrightarrow a$ à P ;
— pour toute état $r \in Q'$: ajouter $L_{q,p} \longrightarrow L_{q,r} L_{r,p}$ à P ;
— pour toutes transitions (q, a, x, r) et $(s, b, \bar{x}, p) \in \Delta'$: ajouter $L_{q,p} \longrightarrow a L_{r,s} b$ à P.

Finalement, ajouter $S \longrightarrow \sum_{f \in F} L_{q_0,f}$ à P.

Par construction de $\mathcal{G}$ et d'après l'affirmation 1, nous pouvons émettre l'affirmation suivante

Affirmation 4. $\mathscr{L}(\mathcal{G}) = \mathscr{L}(\mathcal{A}_2)$ *et pour tout* $X \in N : \mathscr{L}(X) \subseteq \mathcal{T}_{I,\perp} \odot A^+$.

Remarquons que par définition de $\mathcal{A}_2$, pour toute transition de la forme (q, a, ε, p), on a $a \in \perp \times A$; et pour toute transition de la forme (q, a, z, p) avec $z \in \widehat{Z}$, on a $a \in (\perp \times A) \cup \{\varepsilon\}$. Chaque production de $\mathcal{G}$ est alors sous l'une des formes suivantes :

$(1) X \longrightarrow YZ$ avec $Y, Z \in N$;
$(2) X \longrightarrow aYb$ avec $Y \in N, a, b \in \widehat{I} \times A$;
$(3) X \longrightarrow aYb$ avec $Y \in N, a, b \in (\perp \times A) \cup \{\varepsilon\}$;
$(4) X \longrightarrow a$ avec $a \in \perp \times A$.

Les productions de type (3) peuvent ensuite être substituée en productions de type (1) et (4). D'après le lemme 5.40, le langage

$$\mathscr{L}(\phi) = \pi_2(L_0 \cap \pi_1^{-1}(\mathcal{D}_{k,\perp})) = \pi_2(\mathscr{L}(\mathcal{G}) \cap \pi_1^{-1}(\mathcal{D}_{k,\perp}))$$

est un langage indexé double de rang k. Ceci conclut la preuve de la proposition ci-dessus. □

Corollaire 5.46. *Pour tout k, la famille de langage indexés doubles de rang k coïncide avec la famille de langages définissables dans* $\exists \mathrm{Dyck}_k \mathrm{MatchFO}(A, <)$.

Remarquons que dans la preuve de la proposition ci-dessus, nous avons considéré la MSO-définissabilité des langages de mots imbriqués reconnus par des automates à pile visible. Il s'en suit le corollaire suivant.

Corollaire 5.47. *Pour tout k, la famille de langage indexés doubles de rang k coïncide avec la famille de langages définissables dans* $\exists \mathrm{Dyck}_k \mathrm{MatchMSO}(A, <)$.

5.6 Discussions

Dans cette section, nous discutons de l'usage d'une transformation pour caractériser les langages indexés et de façon générale, des cônes principaux.

Définition 5.48 ($\pi_{-\perp}$-Projection)**.** *Soit $\pi_{-\perp}$ la projection d'un alphabet $A_\perp$ dans A. Pour toute formule du second ordre $\phi \in \mathrm{SO}(A_\perp, <)$. Soit*

$$\mathscr{L}_{-\perp}(\phi) = \{\pi_{-\perp}(u) \mid u \in \mathscr{L}(\phi)\}$$

En terme de structures logiques, cette projection consiste à supprimer les éléments du domaine étiquetés par $\perp$.

Proposition 5.49. *Un langage $L \subseteq A^+$ est indexé si et seulement si il existe un entier k et une formule $\phi \in \exists\mathrm{Dyck}_k\mathrm{MatchFO}(A_\perp, <)$ ($\phi \in \exists\mathrm{Dyck}_k\mathrm{MatchMSO}(A, <)$) telle que $L = \mathscr{L}_{-\perp}(\phi)$.*

Démonstration. La preuve en est simple. Le sens ($\Leftarrow$) découle de la proposition 5.45 et des propriétés de clôture des langages indexés. Le sens ($\Rightarrow$) découle du fait que tout langage indexé $L \subseteq A^+$ peut s'exprimer comme $\pi_{-\perp}(L_0)$ où $L_0 \subseteq A^+$ est un indexé double. En effet pour toute grammaire indexée $\mathcal{I} = (N, A, I, P, S)$ il existe une grammaire indexée en forme faiblement double $\mathcal{I}' = (N, A_\perp, I, P, S)$ telle que $\mathscr{L}(\mathcal{I}) = \pi_{-\perp}(\mathscr{L}(\mathcal{I}'))$. La grammaire $\mathcal{I}'$ est construite en transformant toute production $X \longrightarrow Y^\alpha$ en $X \longrightarrow \perp Y^\alpha \perp$, et toute production $X^\alpha \longrightarrow Y$ en $X^\alpha \longrightarrow \perp Y \perp$. □

En particulier, la définissabilité des langages indexés comme projections de langages définissables dans $\exists$DyckMatchMSO peut s'expliquer de façon complètement indépendante d'un quelconque formalisme de reconnaissance des langages indexés. Elle peut s'expliquer par le fait que la famille des langages indexés est un cône principal (Théorème 4.5). Dans ce qui suit, nous montrons que si un langage L est définissable dans une logique du type $\exists\mathcal{B}$ MSO, alors son cône principal appartient à l'ensemble de projections de langages définissables dans $\exists\mathcal{B}$ MSO. Il nous suffira pour cela de montrer que les logiques du type $\exists\mathcal{B}$ MSO sont stables par transductions rationnelles croissantes (ce qui transpire assez visiblement de leurs définitions) et d'exploiter le lien entre cônes et cônes croissants. Nous terminerons néanmoins en soulignant que la logique $\exists$SO n'est pas stable par projection.

5.6.1 Définissabilité des cônes (croissants)

Considérons ici une classe de relations $\mathcal{B}$. Nous désignons par $\mathcal{L}_\mathcal{B}$ la famille de langages définissables dans une logique $\exists\mathcal{B}\,\mathrm{MSO}(A, <)$ pour un alphabet A.

Rappelons (Théorème 1.6) qu'une transduction rationnelle τ est croissante si et seulement si il existe un homomorphisme strictement alphabétique f, un homomorphisme alphabétique g et un langage régulier R tels que $\tau = f \circ \cap R \circ g^{-1}$.

Lemme 5.50. *La famille $\mathcal{L}_\mathcal{B}$ est fermée par intersection avec des langages réguliers.*

Démonstration. De façon évidente, considérons une formule $\phi \equiv \exists X\, \phi$ de $\exists\mathcal{B}\,\mathrm{MSO}(A, <)$ et un langage régulier $R \subseteq A^+$. D'après le théorème de Büchi, il existe une formule ϕ_R de $\mathrm{MSO}(A, <)$ telle que $R = \mathscr{L}(\phi_R)$. Le langage $\mathscr{L}(\phi) \cap R$ est alors définissable par la formule $\phi' \equiv \exists X\, \phi \wedge \phi_R$. □

Lemme 5.51. *La famille $\mathcal{L}_\mathcal{B}$ est fermée par homomorphismes strictement alphabétiques.*

Démonstration. La construction pour la fermeture par homomorphismes strictement alphabétiques est comme suit : Soit $\phi \equiv \exists X\, \phi$ une formule de $\exists\mathcal{B}\,\mathrm{MSO}(A,<)$ et $h : A^* \longrightarrow B^*$ un homomorphisme strictement alphabétique. La formule ϕ' telle que $\mathscr{L}(\phi') = h(\mathscr{L}(\phi))$ est

$$\phi' \equiv \exists X(\exists Q_a)_{a\in A}\, \psi \wedge \phi'_0$$

où ψ est une formule du premier ordre imposant que les $(Q_a)_{a\in A}$ définissent une partition du domaine : chaque position du domaine appartient à un unique ensemble Q_a ; et ϕ'_0 est construite en remplaçant chaque occurrence de $P_a(x)$ par $(Q_a(x) \wedge (P_{f(a)}(x)))$. □

Lemme 5.52. *La famille $\mathcal{L}_\mathcal{B}$ est fermée par homomorphismes alphabétiques inverses.*

Démonstration. Tout homomorphisme alphabétique $g : B^* \longrightarrow A^*$ peut s'exprimer comme $g = \pi_A \circ h$ où π_A est la projection de $A_\perp$ dans A et $h : B^* \longrightarrow {A_\perp}^*$ est l'homomorphisme strictement alphabétique tel que pour tout $b \in B$:

$$h(b) = \begin{cases} g(b) & \text{si } g(b) \neq \varepsilon \\ \perp & \text{si } g(b) = \varepsilon. \end{cases}$$

L'image inverse de g s'exprime alors comme $g^{-1} = h^{-1} \circ \pi_A^{-1}$. Il nous suffit donc de prouver la fermeture par images inverses d'homomorphismes strictement alphabétiques et par projections inverses.

Homomorphismes strictement alphabétiques inverses. La construction pour la fermeture par images inverses d'homomorphismes strictement alphabétiques est similaire à celle pour la fermeture par images directes. Soit $\phi \equiv \exists X\, \phi$ une formule de $\exists\mathcal{B}\,\mathrm{MSO}(A,<)$ et $h : A^* \longrightarrow B^*$ un homomorphisme strictement alphabétique. La formule ϕ' telle que $\mathscr{L}(\phi') = h^{-1}(\mathscr{L}(\phi))$ est

$$\phi' \equiv \exists X(\exists Q_a)_{a\in A}\, \psi \wedge \phi'_0$$

où ψ est une formule du premier ordre imposant que les $(Q_a)_{a\in A}$ définissent une partition du domaine ; et ϕ'_0 est cette fois ci construite en remplaçant chaque occurrence de $P_a(x)$ par $(Q_a(x) \wedge (\bigvee_{b\in h^{-1}(a)} P_b(x)))$.

Projection inverses. Soit $\phi \equiv \exists X\, \phi_0$ une formule de $\exists\mathcal{B}\,\mathrm{MSO}(A,<)$. La formule ϕ' telle que pour tout mot $u \in {A_\perp}^+$, $\underline{u} \models \phi' \Leftrightarrow \underline{\pi_{-\perp}(u)} \models \phi$ est

$$\phi' \equiv \exists X\, \phi'_0 \wedge \varphi$$

avec $\varphi \equiv \exists x : \neg(P_\perp(x))$ et ϕ'_0 est obtenue en transformant toute sous-formule $\exists x : \psi$ de ϕ_0 où x est une variable du premier ordre en $\exists x : (\neg P_\perp(x) \wedge \psi')$. La formule ψ' étant obtenue par le même procédé. □

Les trois lemmes précédents impliquent alors la proposition suivante.

Proposition 5.53. *La famille $\mathcal{L}_\mathcal{B}$ est fermée par transductions rationnelles croissantes.*

Démonstration. D'après les lemmes 5.50,5.51 et 5.52. □

Corollaire 5.54. *Soit $\mathcal{F}$ une famille de langages. Si $\mathcal{F} \subseteq \mathcal{L}_\mathcal{B}$, alors $\mathscr{C}^<(\mathcal{F}) \subseteq \mathcal{L}_\mathcal{B}$.*

Lemme 5.55. *Soit $\mathcal{F}$ une famille de langage. Alors,*

$$\mathscr{C}(\mathcal{F}) = \{\pi_{-\perp}(L) \mid L \in \mathscr{C}^{<}(\mathcal{F})\}.$$

Démonstration. L'inclusion ($\subseteq$) est vraie car la classe de transduction rationnelles est fermée par composition. L'inclusion ($\supseteq$) découle du fait que toute transduction rationnelle τ puisse s'exprimer comme $\pi_{-\perp} \circ \tau_0$, où τ_0 est une transduction rationnelle croissante : D'après [Niv68], toute transduction rationnelle τ peut s'exprimer comme $\tau = h \circ \cap R \circ g^{-1}$ où h et g sont deux homomorphismes alphabétiques et R est un langage régulier. L'homomorphisme h peut lui-même s'exprimer comme $\pi_{-\perp} \circ h_0$ où $h_0 : \text{dom}(h) \longrightarrow \text{co-dom}(h)$ est l'homomorphisme strictement alphabétique tel que pour tout $a \in \text{dom}(h) : h_0(a) = h(a)$ si $h(a) \neq \varepsilon$, $h(a) = \perp$ sinon. La transduction τ s'exprime alors comme $\tau = \pi_{-\perp} \circ \tau_0$ avec $\tau_0 = h_0 \circ \cap R \circ g^{-1}$ qui d'après le théorème 1.6 est croissante. □

Proposition 5.56. *Soit $\mathcal{F}$ une famille de langages. Si $\mathcal{F} \subseteq \mathcal{L}_{\mathcal{B}}$, alors*

$$\mathscr{C}(\mathcal{F}) \subseteq \{\pi_{-\perp}(L) \mid L \in \mathcal{L}_{\mathcal{B}}\}.$$

Démonstration. D'après la proposition 5.53, $\mathcal{L}_{\mathcal{B}}$ est fermé par transductions rationnelles croissantes. Si $\mathcal{F} \subseteq \mathcal{L}_{\mathcal{B}}$, alors $\mathscr{C}^{<}(\mathcal{F}) \subseteq \mathcal{L}_B$. En conjonction avec le lemme 5.55, nous obtenons

$$\mathscr{C}(\mathcal{F}) \subseteq \{\pi_{-\perp}(L) \mid L \in \mathcal{L}_{\mathcal{B}}\}.$$

□

La caractérisation des langages indexés comme projections des langages définissables dans ∃DyckMatchMSO se déduit alors du fait que d'après le théorème 4.5, $\text{IL} = \mathscr{C}(\mathcal{D}^2)$ où $\mathcal{D}^2$ est un langage de la forme $\mathcal{D}_6 \cap \sigma^{-1}(\mathcal{D}_2)$ et l'homomorphisme σ est symétrique et alphabétique. Nous avons aussi vu que tout langage de la forme $\mathcal{D}_A \cap g^{-1}(\mathcal{D}_B)$ avec $g : \widehat{A} \longrightarrow \widehat{B}$ est définissable dans ∃DyckMatchMSO (voir exemple 5.34). Il est difficile d'argumenter sur l'inclusion inverse, et de façon générale sur l'inclusion « $\mathscr{C}(\mathcal{F}) \supseteq \{\pi_{-\perp}(L) \mid L \in \mathcal{L}_{\mathcal{B}}\}$ pour un certain $\mathcal{F}$ »–à moins d'avoir plus d'information sur un éventuel isomorphisme qui envoie chaque structure de la forme (u, M) avec $M \in \mathcal{B}$ sur le domaine de u, vers un mot. Ceci est pourtant trivialement vrai si on remplace la quantification sur $\mathcal{B}$ par une quantification « à la Lindström » sur les mots comme suit :

Pour tout langage $X \subseteq A^*$, soit

$$X_\perp \subseteq A_\perp^* = \pi_{-\perp}^{-1}(L) = \{\alpha_1 a_1 \alpha_2 a_2 \cdots a_n \alpha_{n+1} \mid a_1 a_2 \cdots a_n \in X \text{ et chaque } \alpha_i \in \perp^*\}.$$

Soit alors $\exists X\, \text{MSO}(B, <)$ l'ensemble de formules de la forme $\phi \equiv (\exists X_a)_{a \in A_\perp}\, \phi_0$ telles que chaque variable X_a est unaire l'ensemble des $X_a, a \in A_\perp$ est restreint à définir un mot de $L_\perp$ sur le domaine de la structure ; ϕ_0 est une formule MSO sur la signature $(A_\perp \times B, <)$. Un mot $v \in B^+$ satisfait ϕ si et seulement si il existe $u \in X_\perp$ tel que $(u \odot v)$ satisfait ϕ_0. En d'autre termes :

$$\mathscr{L}(\phi) = \pi_2(\mathscr{L}(\phi_0) \cap \pi_1^{-1}(X_\perp)).$$

Désignons par $\mathcal{L}_X$ l'ensemble de langages définissables dans cette logique. Il n'est pas difficile de voir que $\mathcal{L}_X = \mathscr{C}^{<}(X)$. Il s'en suit que

$$\mathscr{C}(X) = \{\pi_{-\perp}(L) \mid L \in \mathcal{L}_X\}.$$

Terminons cette discussion en soulignant que bien qu'il soit assez pratique de considérer la projection « $\pi_{\_\perp}$ » pour caractériser des familles de langages–pour peu que celles-ci soient des cônes principaux, la logique $\exists$SO (et même SO) n'est pas stable par cette transformation. Nous entendons, la projection d'un langage définissable dans $\exists$SO n'est pas nécessairement définissable dans $\exists$SO.

Proposition 5.57. *Il existe un langage L définissable dans $\exists$SO tel que $\pi_{\_\perp}(L)$ n'est pas définissable dans $\exists$SO.*

Démonstration. Supposons un langage récursivement énumérable L qui n'est pas définissable dans $\exists$SO. D'après [HOY85, HN81], le langage L peut s'exprimer comme $L = h(K \cap \mathcal{D}_k)$ où K est un langage algébrique et h un homomorphisme. Le langage L peut donc s'écrire comme $L = \pi_{\_\perp}(L_0)$ avec $L_0 = f(K \cap \mathcal{D}_k)$ où f est la version non-effaçante de h (si $h(a) = \varepsilon$, alors $f(a) = \perp$). Le langage L_0 est donc définissable dans $\exists$SO puisque que les langages K et $\mathcal{D}_k$ sont algébriques et donc définissables dans $\exists$SO et que f est une transduction rationnelle croissante. □

Deuxième partie

Caractérisations des langages d'ordres supérieurs

Contenu de la partie

Cette partie est dédiée à l'étude des langages d'ordres supérieurs.

— Dans le chapitre 6, nous présentons brièvement les langages d'ordres supérieurs.

— Dans le chapitre 7, nous étendons les notions de langages de Dyck et de transductions ε-sûres et prouvons des caractérisations des langages d'ordres supérieurs par ces objets.
— Dans le chapitre 8, nous nous intéressons aux cônes fidèles principaux. Nous montrons sous quelles conditions un cône fidèle principal admet un théorème de représentation par homomorphisme inverse et nous appliquons ces résultats à des sous-classes des langages d'ordres supérieurs.

Les chapitres 7 et 8 sont indépendants.

Chapitre 6

Langages d'ordres supérieurs

Dans [Mas74], Maslov introduit une généralisation des grammaires indexées d'Aho ainsi qu'une classe d'automates équivalente, les « multilevel stack automata ». Les langages générés par ces grammaires et reconnus par ces automates de différents niveaux forment une hiérarchie *stricte* et *infinie* de familles de langages $(\text{LANG}^k)_{k\geq 0}$ incluses dans la classe des langages contextuels et dans laquelle en particulier, le niveau 0 est celui des langages réguliers, le niveau 1 est celui des langages algébriques et le niveau 2 est celui des langages indexés. Celle-ci est connue sous le nom de « hiérarchie des langages d'ordres supérieurs » ou « hiérarchie OI ». Le nom « hiérarchie OI » vient de son équivalence avec la généralisation des grammaires à macro OI de Fischer introduite par Damm [Dam82].

Tout au long de leur histoire, différents types de formalismes reconnaissant ces langages ont été introduits, en particulier, les automates à piles itérées de Damm, Goerdt et Engelfriet [DG86, Eng83]. Ici, les éléments d'une pile de niveau k (k-pile) sont des $(k-1)$-piles. Nous présentons ici la hiérarchie par ce dernier formalisme. Un élément important de cette présentation est la notion de *générateurs de Maslov*. Celui-ci désigne l'ensemble de suites d'instructions permettant d'obtenir une pile vide depuis une pile vide.

6.1 Piles itérées

Nous présentons ici la notion de *piles itérées* de Damm, Goerdt et Engelfriet.

Définition 6.1. *Une séquence d'alphabets de pile de taille k est une suite* $\boldsymbol{A} = A_1, A_2, \ldots, A_k$ *d'alphabets* deux à deux disjoints *et telle qu'aucun alphabet* A_i *ne contient le symbole* $\perp$.

Notations. Etant donné une séquence d'alphabets de pile $\boldsymbol{A}$, $A_{l,k}$ désigne *l'alphabet* $\bigcup_{i=l}^{k} A_i$, et $\widehat{A}_{l,k}$ désigne l'alphabet $\bigcup_{i=l}^{k} \widehat{A_i}$.

Définition 6.2 (Piles itérées)**.** *Soit* $\boldsymbol{A}$ *une séquence d'alphabets de pile. Les ensembles* k*-Piles*$(\boldsymbol{A})$ *se définissent comme suit :*

$$0\text{-}Piles(\boldsymbol{A}) = \{\varepsilon\}$$
$$(k+1)\text{-}Piles(\boldsymbol{A}) = (A_{k+1}[k\text{-}Piles(\boldsymbol{A})])^* \perp [k\text{-}Piles(\boldsymbol{A})] \text{ pour } k \geq 0.$$

Une k-pile sur $\boldsymbol{A}$ *est un élément de* k*-Piles*$(\boldsymbol{A})$.

Remarque 6.3. *Toute k-pile admet une unique décomposition* $a_k[\omega]\Omega_0$ *avec* $\omega \in (k-1)$*-pile,* $a_k \in A_k \cup \{\perp\}$ *et*

$$\Omega_0 = \varepsilon \text{ si } a_k = \perp, \quad \Omega_0 \in k\text{-}Piles(\boldsymbol{A}) \text{ sinon .}$$

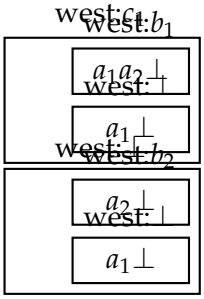


FIGURE 6.1 – Représentation de la 3-pile $c_1\big[b_1[a_1a_2\bot]\ \bot[a_1\bot]\big]\ \bot\big[b_2[a_2\bot]\ \bot[a_1\bot]\big]$.

Conventions
- La k-pile vide est notée $\bot_k$ avec $\bot_0 = \varepsilon$ et $\bot_k = \bot[\bot_{k-1}]$ pour $k > 0$.
- Une 1-pile, disons, $\alpha[\varepsilon]\beta[\varepsilon]\bot[\varepsilon]$ est simplement notée $\alpha\beta\bot$.

Exemple 6.4. *Soit $\boldsymbol{A} = A_1, A_2, A_3$ avec $A_1 = \{a_1, a_2\}$, $A_2 = \{b_1, b_2\}$ et $A_3 = \{c_1, c_2\}$. Alors le mot*

$$c_1\big[b_1[a_1a_2\bot]\ \bot[a_1\bot]\big]\ \bot\big[b_2[a_2\bot]\ \bot[a_1\bot]\big]$$

est une 3-pile sur $\boldsymbol{A}$ (voir illustration figure 6.1).

Définition 6.5. *Pour tout $k \geq 1$, soit* $\text{CONTENU}_k : (k+1)\text{-}Piles(\boldsymbol{A}) \longrightarrow k\text{-}Piles(\boldsymbol{A})$ *la fonction telle que pour toute $(k+1)$-pile $\Omega = a_k[\omega]\Omega_0$ on a* $\text{CONTENU}_k(\Omega) = \omega$.

Exemple 6.6.

$$\text{CONTENU}_2(c\big[b_2[\bot]\ \bot[a\bot]\big]\ \bot\big[b_1[\bot]\ \bot[aaa\bot]\big]) = b_2[\bot]\ \bot[a\bot].$$

6.1.1 Instructions sur les piles itérées

Comme jeu d'instructions sur les piles itérées, nous utilisons celui des opérations symétriques (voir par exemple [Car05]). Un empilement se fait en copiant le contenu de la pile la plus haute, et un dépilement consiste à « défaire » un empilement.

Définition 6.7 (COPIE). *Soit $\Omega = c[\omega]\Omega_0$ une k-pile sur $\boldsymbol{A} = A_1, A_2, \ldots, A_k$ et $a \in A_l$ avec $l \leq k$. Alors*

$$\text{COPIE}_a(\Omega) = \begin{cases} a[\omega]c[\omega]\Omega_0 & \text{si } l = k, \\ c[\text{COPIE}_a(\omega)]\Omega_0 & \text{sinon.} \end{cases}$$

Exemple 6.8. *Soit la 3-pile $\Omega = c_1\big[\bot[a_1a_2\bot]\big]\ \bot_3$ où $A_1 = \{a_1, a_2\}$, $A_2 = \{b_1, b_2\}$ et $A_3 = \{c_1, c_2\}$. Alors nous avons (voir figure 6.2) :*

$$\text{COPIE}_{a_2}(\Omega) = c_1\big[\bot[a_2a_1a_2\bot]\big]\ \bot_3,$$
$$\text{COPIE}_{b_2}(\Omega) = c_1\big[b_2[a_1a_2\bot]\ \bot[a_1a_2\bot]\big]\bot_3,$$
$$\text{COPIE}_{c_2}(\Omega) = c_2\big[\bot[a_1a_2\bot]\big]\ c_1\big[\bot[a_1a_2\bot]\big]\bot_3.$$

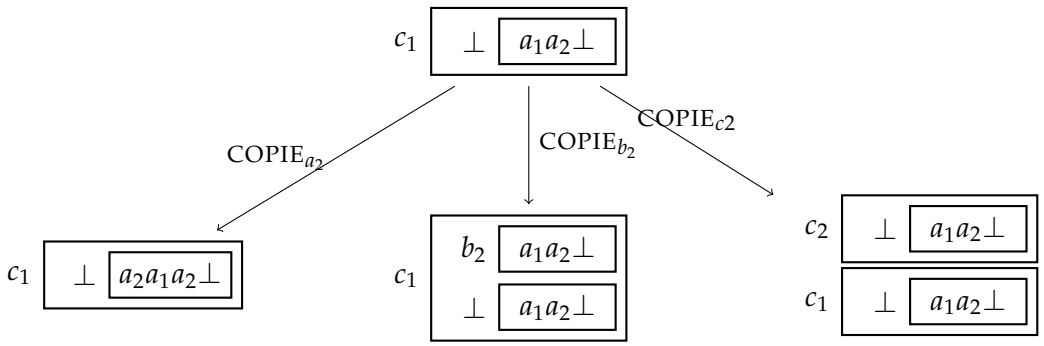


FIGURE 6.2 – Application des instructions $\text{COPIE}_{a_2}, \text{COPIE}_{b_2}$ et COPIE_{c_2} sur la 3-pile $\Omega = c_1\big[\bot[a_1a_2\bot]\big]\,\bot_3$. Ici, $A_1 = \{a_1, a_2\}, A_2 = \{b_1, b_2\}$ et $A_3 = \{c_1, c_2\}$.

Définition 6.9 ($\overline{\text{COPIE}}$)**.** *Pour tout $a \in A_k$, l'instruction $\overline{\text{COPIE}_a}$ se définit comme l'inverse de l'instruction COPIE_a. Celle-ci, à une k-pile Ω associe l'unique k-pile Ω' telle que $\text{COPIE}_a(\Omega') = \Omega$. S'il n'existe pas de tel Ω', alors le résultat est considéré comme* indéfini.

Exemple 6.10. *Soit la 3-pile $\Omega = c\big[b_1[a_1a_2\bot]\; b_2[a_1a_2\bot]\bot_2\big]\,\bot_3$ où $A_1 = \{a_1, a_2\}, A_2 = \{b_1, c_2\}$ et $A_3 = \{c_1, c_2\}$. Alors nous avons :*

$$\begin{aligned}
\overline{\text{COPIE}_c}(\Omega) &= \textit{ indéfini},\\
\overline{\text{COPIE}_{b_1}}(\Omega) &= c\big[b_2[a_1a_2\bot]\big],\\
\overline{\text{COPIE}_{b_2}}(\Omega) &= \textit{ indéfini},\\
\overline{\text{COPIE}_{a_1}}(\Omega) &= c\big[b_1[a_2]\; b_2[a_1a_2]\big],\\
\overline{\text{COPIE}_{a_2}}(\Omega) &= \textit{ indéfini}.
\end{aligned}$$

6.1.2 Représentation de suites d'instructions comme des mots

Soit un entier $k \geq 1$ et $\boldsymbol{A}$ une séquence d'alphabets de pile de taille k. Pour tout $l \in \{1, \ldots, k\}$ et $a \in A_l$, l'instruction COPIE_a est représentée par le symbole a. On note

$$[\![a]\!] = \text{COPIE}_a \quad \text{et} \quad [\![\bar{a}]\!] = \overline{[\![a]\!]} = \overline{\text{COPIE}_a}.$$

Une suite d'instructions est alors représentée par un mot $s = a_1a_2\cdots a_n \in \widehat{A_{1,k}}^*$ et

$$[\![s]\!] = [\![a_n]\!] \circ \cdots \circ [\![a_2]\!] \circ [\![a_1]\!].$$

Nous écrivons aussi $\Omega \xrightarrow{a} \Omega'$ pour signifier que $\Omega' = [\![a]\!](\Omega)$.

Exemple 6.11. *Soit $\boldsymbol{A} = A_1, A_2, A_3$ avec $A_1 = \{a\}, A_2 = \{b\}$ et $A_3 = \{c\}$. Alors*

$$[\![abc\bar{b}]\!](\bot_3) = c\big[\bot[a\bot]\big]\,\bot\big[b[a\bot]\,\bot[a\bot]\big]$$

comme illustré ci-dessus (voir aussi figure 6.3).

$$\begin{aligned}
\bot_3 \xrightarrow{a} \bot\big[\bot[a\bot]\big] \xrightarrow{b} \bot\big[b[a\bot]\,\bot[a\bot]\big] &\xrightarrow{c} c\big[b[a\bot]\,\bot[a\bot]\big]\,\bot\big[b[a\bot]\,\bot[a\bot]\big]\\
&\xrightarrow{\bar{b}} c\big[\bot[a\bot]\big]\,\bot\big[b[a\bot]\,\bot[a\bot]\big].
\end{aligned}$$

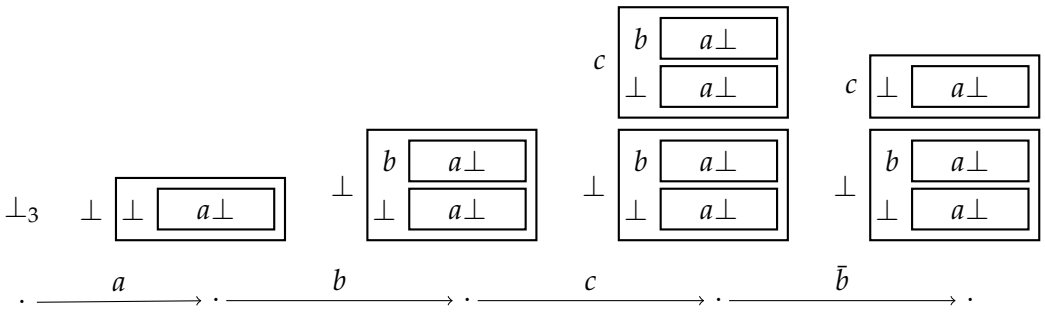


FIGURE 6.3 – Résultat de $[\![abc\bar{b}]\!]$ sur la pile $\perp_3$.

6.2 Hiérarchie des langages d'ordres supérieurs

Nous présentons brièvement la notion d'automates à piles itérées et les langages d'ordres supérieurs. Un automate à k-pile est un automate fini muni d'une k-pile comme structure de mémoire. Un langage de niveau k (un langage de LANG^k) est un langage reconnu par un automate à k-pile.

Définition 6.12 (Syntaxe des automates à k-pile)**.** *Un automate à k-pile (k-AP) est une structure* $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \Delta, F)$ *dans laquelle*

- Q *est l'ensemble fini d'états,*
- Σ *est l'alphabet de travail,*
- $q_0 \in Q$ *est l'état initial,*
- $\boldsymbol{A}$ *est une suite d'alphabets de pile de taille k,*
- $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\widehat{A}_{1,k} \cup \{\varepsilon\}) \times Q$ *est la relation de transitions,*
- *et* $F \subseteq Q$ *est l'ensemble d'états acceptants.*

Définition 6.13 (Sémantique des automates à k-pile)**.** *Une configuration est un triplet* $(q, \Omega, u) \in Q \times k\text{-}Piles(\boldsymbol{A}) \times A^*$*. Une étape de calcul de* $\mathcal{A}$ *est décrite par la relation binaire* $\underset{\mathcal{A}}{\vdash}$ *entre configurations. Étant données deux configurations* $c_1 = (q, \Omega, u)$ *et* $c_2 = (p, \Omega', v)$*, nous avons* $c_1 \underset{\mathcal{A}}{\vdash} c_2$ *si et seulement si il existe une transition* (q, a, α, p) *avec* $[\![\alpha]\!](\Omega) = \Omega'$ *et* $u = av$*. La relation* $\underset{\mathcal{A}}{\vdash}^*$ *désigne la clôture réflexive et transitive de* $\underset{\mathcal{A}}{\vdash}$*. Le langage reconnu par* $\mathcal{A}$ *est*

$$\mathscr{L}(\mathcal{A}) = \{u \in \Sigma^* \mid \exists q_f \in F : (q, \perp_k, u) \underset{\mathcal{A}}{\vdash}^* (q_f, \perp_k, \varepsilon)\}.$$

Nous désignons par LANG^k *l'ensemble de langages reconnaissables par automates à k-pile.*

Notations. Nous décrivons un calcul par une suite de transitions effectuées e.g.

$$q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_{n+1}$$

avec chaque $(q_i, a_{i+1}, \alpha_{i+1}, q_{i+1}) \in \Delta$.

Définition 6.14 (Hiérarchie d'ordres supérieurs)**.** *Pour tout* $k \geq 0$*, soit* LANG^k *la famille de langages reconnaissables par automates à k-pile.*

La hiérarchie des langages d'ordres supérieurs est la hiérarchie des familles de langages $(\text{LANG}^k)_{k\geq 0}$.

Cette hiérarchie est stricte et infinie. En particulier, LANG^0 est la famille des langages réguliers, LANG^1 est celle des langages algébriques, et LANG^2 est celle des langages indexés. De plus,

Théorème 6.1 ([Mas74]). *Pour tout $k \geq 0$, LANG^k est une famille agréable de langages.*

6.2.1 Générateurs de Maslov

Maslov a énoncé dans [Mas74] l'existence d'un langage décrivant des suites d'instructions acceptantes d'un automate à k-pile. Nous les présentons ici comme étant des suites d'instructions de pile permettant d'obtenir une pile vide depuis une pile vide. Bien que ce ne soit *stricto sensu* la définition de Maslov (nous utilisons ici des automates à k-pile), le langage que nous présentons ci-dessous joue le même rôle vis-à-vis de la hiérarchie des ordres supérieurs. Nous attribuons la définition ci-dessous à Maslov.

Définition 6.15 (Générateurs de Maslov). *Soit $k \geq 0$ et $\boldsymbol{A}$ une séquence d'alphabets de pile de taille k. Soit*

$$\mathcal{M}^k_{\boldsymbol{A}} = \{s \in \widehat{A}^*_{1,k} \mid [\![s]\!](\bot_k) = \bot_k\}.$$

Notation. Nous désignons par $\mathcal{M}^k$ le langage $\mathcal{M}^k_{\boldsymbol{A}}$ où chaque alphabet A_i de la séquence $\boldsymbol{A}$ est de taille 2.

Exemple 6.16. *Soit la séquence $\boldsymbol{A} = A_1, A_2$ avec $A_1 = \{a\}$ et $A_2 = \{b\}$. Le mot $u = ab\bar{a}a\bar{b}\bar{a}$ appartient à $\mathcal{M}^2_{\boldsymbol{A}}$. Nous pouvons en effet observer que $[\![u]\!](\bot_2) = \bot_2$ comme suit :*

$$\bot_2 \xrightarrow{a} \bot[a\bot] \xrightarrow{b} b[a\bot]\,\bot[a\bot] \xrightarrow{\bar{a}} b[\bot]\,\bot[a\bot] \xrightarrow{a} b[a\bot]\,\bot[a\bot] \xrightarrow{\bar{b}} \bot[a\bot] \xrightarrow{\bar{a}} \bot_2.$$

Le mot $v = ab\bar{a}\bar{b}ba\bar{b}\bar{a}$ n'appartient pas à $\mathcal{M}^2_{\boldsymbol{A}}$ car en effet nous avons

$$\bot_2 \xrightarrow{a} \bot[a\bot] \xrightarrow{b} b[a\bot]\bot[a\bot] \xrightarrow{\bar{a}} b[\bot]\bot[a\bot] \xrightarrow{\bar{b}} \textit{indéfini}.$$

Exemple 6.17. *Soit $A_1 = \{a\}, A_2 = \{b\}, A_3 = \{c\}$ et la séquence $\boldsymbol{A} = A_1, A_2, A_3$. Le mot $u = ab\bar{a}ca\bar{a}\bar{c}a\bar{b}\bar{a}$ appartient à $\mathcal{M}^3_{\boldsymbol{A}}$.*

Lemme 6.18. *Pour tout $k \geq 0$, et toute séquence d'alphabets de pile $\boldsymbol{A}$, le langage $\mathcal{M}^k_{\boldsymbol{A}}$ appartient à LANG^k.*

Démonstration. En effet chaque langage $\mathcal{M}^k_{\boldsymbol{A}}$ est reconnu par le k-AP ayant pour ensemble de transitions $\Delta = \{(q_0, \alpha, \alpha, q_0) \mid \alpha \in \widehat{A}_{1,k}\}$ et q_0 comme unique état, à la fois initial et acceptant. Puisque l'automate reconnait sur pile vide, le langage reconnu par celui-ci est alors

$$\{u \in \widehat{A}^*_{1,k} \mid [\![u]\!](\bot_k) = \bot_k\} = \mathcal{M}^k_{\boldsymbol{A}}.$$

□

Théorème 6.2 ([Mas74]). *Pour tout $k \geq 0$:*

$$\text{LANG}^k = \mathscr{C}(\mathcal{M}^k).$$

Le théorème ci-dessus peut s'expliquer comme suit. Puisque chaque famille $\textsc{Lang}^k$ est une famille agréable de langages, et que $\mathcal{M}^k \in \textsc{Lang}^k$, il en suit que

$$\mathscr{C}(\mathcal{M}^k) \subseteq \quad \mathscr{C}(\textsc{Lang}^k) = \textsc{Lang}^k \qquad \forall k \geq 0.$$

Réciproquement, pour tout $k \geq 0$ et étant donné un k-AP $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \Delta, F)$, un calcul $q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_n$ avec $q_n \in F$ est acceptant si et seulement si $[\![\alpha_1\alpha_2\cdots\alpha_n]\!](\bot_k) = \bot_k$, c'est-à-dire si et seulement si la suite d'instructions $\alpha_1\alpha_2\cdots\alpha_n$ appartient à $\mathcal{M}^k_{\boldsymbol{A}}$. Le langage reconnu peut alors se décrire comme $\mathscr{L}(\mathcal{A}) = \tau(\mathcal{M}^k_{\boldsymbol{A}})$, où τ est la transduction rationnelle décrite par le transducteur $\mathcal{T} = (Q, \widehat{A}_{1,k}, \Sigma, q_0, \Delta', F)$ obtenu en remplaçant toute transition (q, a, α, p) de Δ par $(q, \alpha|a, p)$. Le calcul susmentionné correspond ainsi au calcul

$$q_0 \xrightarrow{\alpha_1|a_1} q_1 \xrightarrow{\alpha_2|a_2} q_2 \cdots q_{n-1} \xrightarrow{\alpha_n|a_n} q_n$$

dans $\mathcal{T}$. Ceci combiné au fait qu'une instruction de pile sur un alphabet de taille quelconque peut être codée par une suite d'instructions sur un alphabet de taille 2 (une instruction a_k avec $a_k \in A_l$ peut être décrite par la suite d'instruction $0_l 1_l^k 0_l$), il en résulte que $\textsc{Lang}^k \subseteq \mathscr{C}(\mathcal{M}^k)$ et ainsi le théorème ci-dessus.

6.2.2 Séquences d'instructions valides

Nous aurons besoin d'argumenter sur la syntaxe des mots appartenant aux générateurs de Maslov. Les objets que nous présentons ci-dessous nous seront utiles à cette fin.

Définition 6.19 (Séquences d'instructions valides[Fra05])**.** *Soit $k \geq 0$,*
— *$\mathcal{V}^k_{\boldsymbol{A}}$ est l'ensemble des $u \in \widehat{A}^*_{1,k}$ tel que la pile $[\![u]\!](\bot_k)$ est définie.*
— *$\mathcal{P}^k_{\boldsymbol{A}}$ est l'ensemble des $\rho(u)$ tels que $u \in \mathcal{V}^k_{\boldsymbol{A}}$.*
En particulier, $\mathcal{V}^0_{\boldsymbol{A}} = \mathcal{P}^0_{\boldsymbol{A}} = \{\varepsilon\}$.

L'opération $\overline{\textsc{copie}}$ est définie comme l'inverse de $\textsc{copie}$: $\overline{\textsc{copie}_{k,a}} \circ \textsc{copie}_{k,a} = \mathrm{id}$; et pour toute k-pile Ω, si la pile $\overline{\textsc{copie}_a}(\Omega)$ est définie, alors $\textsc{copie}_a \circ \overline{\textsc{copie}_a}(\Omega) = \Omega$. Sous condition qu'un suite d'instruction appartienne à $\mathcal{V}^k_{\boldsymbol{A}}$, la composition $\textsc{copie}_a \circ \overline{\textsc{copie}_a} = \overline{\textsc{copie}_a} \circ \textsc{copie}_a = \mathrm{id}$ coïncide avec la réduction de Dyck bilatérale $\{a\bar{a} \rightsquigarrow \varepsilon \mid a \in \widehat{A}_{1,k}\}$. Il en découle ainsi le lemme suivant.

Lemme 6.20 ([Fra05])**.** *Pour tout $u \in \mathcal{V}^k_{\boldsymbol{A}}$: $[\![u]\!](\bot_k) = [\![\rho(u)]\!](\bot_k)$. En particulier, $\mathcal{P}^k_{\boldsymbol{A}} \subseteq \mathcal{V}^k_{\boldsymbol{A}}$.*

Définition 6.21 ([Fra05])**.** *Pour tout $k \geq 0$, soit $\textsc{pile}_k : \mathcal{P}^k_{\boldsymbol{A}} \longrightarrow k\text{-Piles}(\boldsymbol{A})$ la fonction qui à tout mot $u \in \mathcal{P}^k_{\boldsymbol{A}}$ associe la k-pile $[\![u]\!](\bot_k)$.*

Exemple 6.22. *Conformément à l'exemple 6.17, si $A_1 = \{a\}, A_2 = \{b\}, A_3 = \{c\}$, alors*

$$\textsc{pile}_k(ab\bar{a}c) = c\big[b[\bot]\,\bot[a\bot]\big]\,\bot\big[b[\bot]\,\bot[a\bot]\big].$$

Proposition 6.23 ([Fra05])**.** *Pour tout $k \geq 0$: la fonction $\textsc{pile}_k$ est bijective.*

Lemme 6.24. *Pour tout $k \geq 1$: $\mathcal{M}^k_{\boldsymbol{A}} = \mathcal{T}_{\boldsymbol{A}_{1,k}} \cap \mathcal{V}^k_{\boldsymbol{A}}$.*

Démonstration. Par définition, $\mathcal{M}^k_{\boldsymbol{A}}$ désigne l'ensemble des $u \in \mathcal{V}^k_{\boldsymbol{A}}$ tels que $[\![u]\!](\bot_k) = \bot_k$; et d'après le lemme 6.20, l'ensemble de mots $u \in \mathcal{V}^k_{\boldsymbol{A}}$ tels que $\textsc{pile}_k(\rho(u)) = \bot_k$. Par bijectivité de $\textsc{pile}_k$ (Proposition 6.23), $\textsc{pile}_k^{-1}(\bot_k) = \varepsilon$. Le langage $\mathcal{M}^k_{\boldsymbol{A}}$ est donc l'ensemble de mots $u \in \mathcal{V}^k_{\boldsymbol{A}}$ tels que $\rho(u) = \varepsilon$. □

Chapitre 7

Itérations du langage de Dyck

7.1 Introduction

Il existe dans la littérature plusieurs caractérisations de certaines familles des langages d'ordres supérieurs vis-à-vis des arbres et du langage de Dyck. Dans [MW67], il est montré qu'un langage est algébrique si et seulement si il est la frontière d'une forêt reconnaissable. De façon analogue, le théorème de Chomsky–Schützenberger stipule qu'un langage est algébrique si et seulement si il est l'image du langage de Dyck par une transduction rationnelle. En effet, le langage de Dyck permet de décrire des structures arborescentes et la transduction rationnelle appliquée permet de décrire la frontière d'une structure arborescente reconnaissable. De même, Guessarian montre dans [Gue83] qu'un langage est indexé si et seulement si il est la frontière d'une forêt algébrique et nous avons vu (Chapitre 4) qu'un langage est indexé si et seulement si il est l'image d'un langage de Dyck *indexé* par une transduction rationnelle. De façon générale, un langage appartient à un niveau $k+1$ de la hiérarchie si et seulement si il est la frontière d'une forêt de niveau k [Dam82]. Il n'existe toutefois pas de notion de langages de Dyck d'ordres supérieurs pour les niveaux > 2 même si des langages tels que les générateurs de Maslov s'y apparentent.

Dans ce chapitre, nous introduisons cette notion. Nous définissons un langage de Dyck de niveau > 1 comme un ensemble de mots pour lesquels l'image par un homomorphisme ε-sûr donné appartient à un langage de Dyck de niveau $k-1$. Nous montrons alors que pour tout $k \geq 1$, un langage appartient à LANG^k si et seulement si il est l'image par transduction rationnelle d'un langage de Dyck de niveau k. De plus, chaque niveau de la hiérarchie des ordres supérieurs est le cône rationnel principal généré par *un* langage de Dyck du même niveau. Nous étudions ensuite l'incidence d'une telle caractérisation sur une famille de langages. *Que peut-t-on dire sur une famille de langages si elle est le cône rationnel généré par une classe quelconque d'itération du langage de Dyck (par une classe quelconque d'homomorphismes) ?* Nous montrons que toute telle famille admet comme formalisme de reconnaissance une certaine restriction de machines à plusieurs piles pour lesquelles la fonction d'écriture sur la structure peut être spécifiée par une séquence d'homomorphismes. Nous les appelons « machines à séquences » de niveaux supérieurs. Une notion similaire est introduite par Sorokin dans [Sor14] pour la caractérisation de forêts algébriques *simples*. Nous introduisons aussi une notion de transductions et transducteurs à séquences et nous montrons qu'un langage appartient à une famille générée par une itération du langage de Dyck de niveau $k+l$ si et seulement si il est l'image d'un langage de Dyck de niveau k par une transduction à séquence de niveau l. Ce résultat explique en particulier pourquoi les langages indexés peuvent être décrits comme des images du langage de

Dyck par des transductions algébriques ε-sûres. Cette dernière forme de caractérisation n'est donc pas une particularité des langages indexés mais une conséquence du fait que la famille des langages indexés est générée par une itération du langage de Dyck. De façon analogue, Vogler montre dans [Vog88] que les langages générés par des grammaires d'ordres supérieurs de niveau k et contrôlés par des langages d'ordres supérieurs de niveau l appartiennent au niveau $k+l+1$ de la hiérarchie des langages d'ordres supérieurs.

Plan du chapitre La section 7.2 est préliminaire. Dans la section 7.3, nous introduisons la notion de langage de Dyck itéré. Les section 7.4, 7.5 et 7.6 sont dédiées à la preuve de la caractérisation des familles de langages d'ordres supérieurs comme cônes rationnels générés par des itérations du langage Dyck par des homomorphismes ε-sûrs. La section 7.7 résume plusieurs théorèmes de représentations des langages d'ordre supérieurs. Dans la section 7.8, nous étudions de façon générale les familles de langages générées par des itérations du langage de Dyck et nous introduisons les notions de machines et transductions à séquences. Dans la section 7.9, nous approfondissons certaines notions que nous avons volontairement omises dans les sections précédentes afin de simplifier la lecture du chapitre.

7.2 Préliminaires

Notations Tout au long de ce chapitre, nous utilisons les notations suivantes.

- Les suites (d'alphabets) sont notées en police gras (eg : $\boldsymbol{A}$).
- Soit $\boldsymbol{A} = A_1, A_2, \ldots, A_k, \ldots$ une suite finie d'alphabets,
  - $|\boldsymbol{A}|$ désigne la taille de la suite $\boldsymbol{A}$;
  - $\widehat{\boldsymbol{A}}$ désigne la séquence d'alphabets $\widehat{A_1}, \widehat{A_2}, \ldots, \widehat{A_k}, \ldots$;
  - $A_{l,k}$ désigne *l'alphabet* $\bigcup_{i=l}^{k} A_i$, et $\widehat{A}_{l,k}$ l'alphabet $\bigcup_{i=l}^{k} \widehat{A_i}$;
  - pour tout $l \leq |\boldsymbol{A}|$: $\pi_{\boldsymbol{A},l}$ (resp. $\pi_{\widehat{\boldsymbol{A}},l}$) désigne la projection dans $A_{1,l}$ (resp. $\widehat{A}_{1,l}$). Nous ne précisons pas l'alphabet de départ afin de ne pas surcharger nos notations. L'alphabet de départ d'une projection sera toujours celui défini dans le contexte e.g. dans une expression « $\pi_B(X)$ » ou encore « $X \cap \pi_B^{-1}(Y)$ », l'alphabet de départ de π_B est l'alphabet de X. S'il peut y avoir une ambigüité (e.g. plusieurs alphabets définis dans un même contexte), nous préciserons les alphabets de départ et d'arrivée de π.

Définition 7.1. *Soit $\boldsymbol{A}$ une séquence d'alphabets. Pour tout alphabet B, nous appelons* l'extension *de $\boldsymbol{A}$ par B la séquence $A_1, A_2, \ldots, A_{|\boldsymbol{A}|}, B$.*

Rappels.

- Le langage de Dyck sur un alphabet A se définit de façon équivalente comme l'ensemble de mots $u \in \widehat{A}^*$ tels que $\rho(u) = \varepsilon$ et pour tout préfixe $u_0 \preccurlyeq u : \rho(u_0) \in A^*$.
- Rappelons (Définition 3.1) qu'un homomorphisme $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ est dit ε-sûr si $f(\mathcal{T}_A) \subseteq \mathcal{T}_B$; et symétrique si pour tout $a \in \widehat{A}$: $f(\bar{a}) = \overline{f(a)}$. Lorsque nous précisons qu'un homomorphisme est symétrique, nous ne précisons que ses valeurs pour l'alphabet positif A.

7.3 Itérations du langage de Dyck

Nous introduisons ici la notion de langages de Dyck itérés. Un langage de Dyck itéré de niveau k est un ensemble de mots bien parenthésés pour lesquels l'image par un homomorphisme donné appartient à un langage de Dyck itéré de niveau $k-1$. Cette notion est présente dans la littérature « au niveau 2 ». En effet Weir [Wei88], Kanazawa [Kan14b] et Sorokin [Sor14] ont considéré différents langages de Dyck de niveau 2 afin de caractériser les forêts d'arbres algébriques simples. Nous avons nous même considéré de tels langages (Chapitre 4) afin de caractériser la famille des langages indexés.

Définition 7.2 (Séquences d'homomorphismes)**.** *Pour tout $k \geq 1$, soit $\boldsymbol{A}$ une séquence d'alphabets de taille k. Une séquence d'homomorphismes sur $\boldsymbol{A}$ est une séquence $\boldsymbol{f} = (f_i)_{1 \leq i < k}$ telle que chaque f_k est un homomorphisme de $\widehat{A_{k+1}}^*$ vers $\widehat{A_k}^*$.*

— *Pour tout $i \leq k$: $f_{i,i}$ désigne l'homomorphisme f_i ;*

— *pour tout $i \leq j$: $f_{i,j}$ désigne l'homomorphisme $f_i \circ \cdots \circ f_{j-1} \circ f_j$.*

Convention. Nous considérons une séquence d'homomorphismes vide, comme une séquence ne contenant que la fontion *identité*.

Notation. Nous userons tout au long de cette section plusieurs séquences d'homomorphismes. L'indice de chaque fonction sera toujours indicatif de son alphabet d'arrivée.

Définition 7.3 (Itérations du langage de Dyck)**.** *Pour tout $k \geq 1$, soit $\boldsymbol{A}$ une séquence d'alphabets disjoints de taille k et $\boldsymbol{f}$ une séquence d'homomorphismes sur $\boldsymbol{A}$. Le langage $D_{k,\boldsymbol{f}}$ se définit inductivement comme suit :*

$$D_{1,\boldsymbol{f}} = \mathcal{D}_{A_1}; \qquad D_{k+1,\boldsymbol{f}} = \mathcal{D}_{A_{k+1}} \cap f_k^{-1}(D_{k,\boldsymbol{f}}).$$

Remarque 7.4. *Un langage, disons,*

$$\mathcal{D}_{A_5} \cap f_4^{-1}\big[\mathcal{D}_{A_4} \cap f_3^{-1}\big(\mathcal{D}_{A_3} \cap f_2^{-1}(\mathcal{D}_{A_2} \cap f_1^{-1}(\mathcal{D}_{A_1}))\big)\big]$$

s'écrit de façon équivalente (par distributivité) comme

$$\mathcal{D}_{A_5} \cap f_4^{-1}(\mathcal{D}_{A_4}) \cap f_{3,4}^{-1}(\mathcal{D}_{A_3}) \cap f_{2,4}^{-1}(\mathcal{D}_{A_2}) \cap f_{1,4}^{-1}(\mathcal{D}_{A_1}).$$

Nous pouvons alors le décomposer en itérations de niveaux inférieurs comme suit :

$$\big(\mathcal{D}_{A_5} \cap f_4^{-1}(\mathcal{D}_{A_4} \cap f_3^{-1}(\mathcal{D}_{A_3}))\big) \cap f_{2,4}^{-1}(\mathcal{D}_{A_2} \cap f_1^{-1}(\mathcal{D}_{A_1})).$$

De façon générale, un langage $D_{k+l,\boldsymbol{f}}$ peut s'exprimer comme $D_{k,\boldsymbol{f}} \cap f_{1,k}^{-1}(D_{l,\boldsymbol{g}})$ où $\boldsymbol{g}$ est la séquence des $l-1$ derniers homomorphismes de $\boldsymbol{f}$.

7.3.1 Itérations par des homomorphismes ε-sûrs

Nous définissons alors les langages de Dyck *d'ordre supérieurs* comme des langages de Dyck itérés par des séquences d'homomorphismes ε-sûrs.

Définition 7.5 (Langages de Dyck d'ordres supérieurs)**.** *Pour tout $k \geq 1$, ε-DYCK^k désigne l'ensemble de langages de la forme $D_{k,\boldsymbol{f}}$ où $\boldsymbol{f}$ est une séquence d'homomorphismes ε-sûrs.*

Exemple 7.6. *Soit* $A_1 = \{a\}, A_2 = \{a, b\}, A_3 = \{a, b, c\}$ *et* $\boldsymbol{A} = A_1, A_2, A_3$ *;* $\boldsymbol{f}$ *la séquence d'homomorphismes* symétriques *sur* $\boldsymbol{A}$ *telle que*

$$f_2 : \widehat{A}_3^* \longrightarrow \widehat{A}_2^* : \begin{cases} a \mapsto a \\ b \mapsto b \\ c \mapsto \bar{b} \end{cases} \quad et \quad f_1 : \widehat{A}_2^* \longrightarrow \widehat{A}_1^* : \begin{cases} a \mapsto a \\ b \mapsto \bar{a} \end{cases}$$

Le mot $u = abc\bar{c}\bar{b}\bar{a}$ *appartient alors à* $D_{3,\boldsymbol{f}}$ *comme illustré ci-dessous.*

$$\begin{aligned} u &= abc\bar{c}\bar{b}\bar{a} \in \mathcal{D}_{A_3} \\ f_2(u) &= ab\bar{b}b\bar{b}\bar{a} \in \mathcal{D}_{A_2} \\ f_1 \circ f_2(u) &= a\bar{a}a\bar{a}a\bar{a} \in \mathcal{D}_{A_1}. \end{aligned}$$

Le mot $v = acb\bar{b}c\bar{a}$ *n'appartient pas à* $D_{3,\boldsymbol{f}}$ *puisque* $f_2(v) = a\bar{b}b\bar{b}ba \notin \mathcal{D}_{A_2}$*. De même, le mot* $w = babc\bar{c}\bar{b}\bar{a}\bar{b}$ *n'appartient pas à* $D_{3,\boldsymbol{f}}$ *puisque* $f_1 \circ f_2(w) = \bar{a}a\bar{a}a\bar{a}aa \notin \mathcal{D}_{A_1}$.

La raison pour laquelle nous appelons ces langages, « langages de Dyck d'ordres supérieurs » est que comme nous le montrerons, le cône généré par chaque famille ε-DYCK^k est précisément la famille LANG^k.

7.4 Propriétés des générateurs de Maslov

Afin de montrer que chaque famille ε-DYCK^k génère précisément la famille LANG^k, nous avons besoin de mettre en relation les langages de ε-DYCK^k et les langages $\mathcal{M}^k_{\boldsymbol{A}}$. L'inconvénient de la définition des générateurs de Maslov est que celle-ci est dépendante de la notion d'automates à k-pile. Dans cette section, nous prouvons que les générateurs de Maslov peuvent se décrire comme des itérations d'une version à la fois unilatérale et bilatérale du langage de Dyck (en considérant une partition de l'alphabet) que nous appelons langage de Dyck *mixte*. Des travaux similaires sur des suites d'instructions ont déjà été menés. Nous pouvons citer [Car05] où Carayol étudie la notion de *régularité* des suites d'instructions.

Cette section est découpée en trois sous-sections. Dans la première nous introduisons le langage de Dyck *mixte* ; dans la deuxième nous prouvons quelques propriétés des suites d'instructions ; et dans la troisième, nous prouvons notre caractérisation des générateurs de Maslov.

7.4.1 Langage de Dyck mixte

Nous introduisons ici le langage de Dyck *mixte*. Celui-ci décrit des mots qui sont simultanément bilateralément bien parenthésés sur une partie de l'alphabet et unilatéralement bien parenthésé sur l'autre. Celui-ci nous permettra de décrire les générateurs de Maslov.

Prenons en guise d'exemple simple la séquence d'alphabets de pile $\boldsymbol{A} = A_1, A_2$ avec $A_1 = \{a\}$ et $A_2 = \{b\}$. Le mot u ci-dessous appartient à $\mathcal{M}^2_{\boldsymbol{A}}$:

$$u = ab\bar{a}a\bar{b}\bar{a}.$$

La succession des symboles $b \in \widehat{A_2}$ décrit un mot de Dyck– ce qui est normal puisque les instructions COPIE doivent être effectuées avant les $\overline{\text{COPIE}}$. Nous pouvons aussi remarquer que le facteur $\bar{a}a$ (entre le b et le $\bar{b}$) n'appartient pas à $\mathcal{D}_{A_1}$ mais plutôt à $\mathcal{T}_{A_1}$. D'une part ceci est possible du fait que le symbole « a » a été empilé avant d'être copié par « b », l'instruction $\bar{a}$

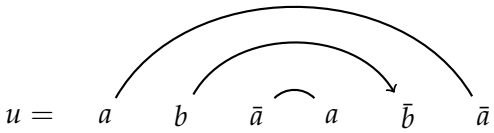


FIGURE 7.1 – Exemple de mot $u \in \widetilde{\mathcal{D}}_{A_2,A_1}$ avec $A_1 = \{a\}$ et $A_2 = \{b\}$.

du facteur $\bar{a}a$ est donc valide. D'autre part, l'appartenance de ce facteur à $\mathcal{T}_{A_1}$ est imposée par la sémantique de l'opération $\overline{\text{COPIE}_b}$. En effet, l'instruction $\overline{\text{COPIE}_b}$ dans cette suite ne peut s'effectuer que parce que la pile $[\![ab]\!](\bot_2)$ est équivalente à $[\![ab\bar{a}a]\!](\bot_2)$. Le langage de Dyck *mixte* que nous définissons ci-dessous nous permet d'exprimer ces deux types de dépendance.

Définition 7.7. *Soient A_1 et A_2 deux alphabets disjoints et S la réduction $\{a\bar{a} \rightsquigarrow \varepsilon \mid a \in \widehat{A_1 \cup A_2}\}$.*

Pour tout mot $u \in (\widehat{A_1 \cup A_2})^*$: $\widetilde{\rho}_{A_2,A_1}(u)$ désigne l'unique mot S-réduit équivalent à u et $\widetilde{\mathcal{D}}_{A_2,A_1}$ désigne le langage

$$\widetilde{\mathcal{D}}_{A_2,A_1} = \{u \in (\widehat{A_2 \cup A_1})^* \mid \widetilde{\rho}_{A_2,A_1}(u) = \varepsilon\}.$$

Tel qu'il est défini, il est évident que le langage $\widetilde{\mathcal{D}}_{A_2,A_1}$ est algébrique et généré par la grammaire

$$S \longrightarrow \varepsilon + SS + \sum_{a \in A_2 \cup \widehat{A_1}} aS\bar{a}.$$

Remarque 7.8. *Par définition, nous avons $\widetilde{\mathcal{D}}_{A_2,\emptyset} = \mathcal{D}_{A_2}$ et $\widetilde{\mathcal{D}}_{\emptyset,A_1} = \mathcal{T}_{A_1}$.*

Exemple 7.9. *Soit $A_1 = \{a\}$ et $A_2 = \{b\}$, alors*

— *le mot $u = ab\bar{a}a\bar{b}\bar{a}$ appartient à $\widetilde{\mathcal{D}}_{A_2,A_1}$ comme illustré sur la figure 7.1 ;*

— *le mot $v = ab\bar{a}\bar{b}ba\bar{b}\bar{a}$ n'appartient pas à $\widetilde{\mathcal{D}}_{A_2,A_1}$ (bien qu'il appartient à $\mathcal{T}_{A_2 \cup A_1}$ et que sa projection sur $\widehat{A_1}$ appartient à $\mathcal{D}_{A_1}$). En effet, nous pouvons observer qu'aucun de ses facteurs de taille 2 n'est réductible par $\{a\bar{a} \rightsquigarrow \varepsilon \mid a \in A_2 \cup \widehat{A_1}\}$. Nous pouvons aussi remarquer qu'il n'appartient pas à $\mathcal{M}^2_{\boldsymbol{A}}$.*

Lemme 7.10. *Soient A_1 et A_2 deux alphabets disjoints. Nous avons*

$$\widetilde{\mathcal{D}}_{A_2,A_1} \subsetneq \mathcal{T}_{A_2 \cup A_1} \cap \pi^{-1}_{\widehat{A_1}}(\mathcal{D}_{A_1}).$$

Démonstration. Ceci découle de la définition même de $\widetilde{\mathcal{D}}_{A_2,A_1}$. Tout mot appartenant à $\widetilde{\mathcal{D}}_{A_2,A_1}$ est bilatéralement bien parenthésé sur l'alphabet $\widehat{A_2 \cup A_1}$ et unilatéralement bien parenthésé sur l'alphabet $\widehat{A_1}$. Nous avons vu par l'exemple 7.9 que cette inclusion est stricte. □

7.4.2 Propriétés des suites d'instructions

Nous prouvons ici quelques propriétés utiles des suites d'instructions.

Lemme 7.11. *Pour tout $k \geq 1$ et pour tout $u \in \mathcal{M}^k_{\boldsymbol{A}}$: $\pi_{\widehat{A_k}}(u) \in \mathcal{D}_{A_k}$.*

Démonstration. D'après le lemme 6.24, pour tout mot $u \in \mathcal{M}_{\boldsymbol{A}}^k$, nous avons $u \in \mathcal{T}_{A_{1,k}}$ et par conséquent $\pi_{\widehat{A_k}}(u) \in \mathcal{T}_{A_k}$ (la projection est un homomorphisme ε-sûr). De plus, chaque instruction $\text{COPIE}_{k,a}$ précède l'instruction $\overline{\text{COPIE}_{k,a}}$ associée. Tout préfixe $u_0 \preccurlyeq \pi_{\widehat{A_k}}(u)$ contient donc plus symboles a que de symbole $\bar{a}$ quelque soit $a \in A_k$. Le mot $\pi_{\widehat{A_k}}$ est donc par définition un mot de Dyck. □

Lemme 7.12. *Soit $k \geq 1$ et $u \in \mathcal{V}_{\boldsymbol{A}}^k$. Alors*

$$\pi_{\widehat{\boldsymbol{A}},k-1}(u) \in \mathcal{V}_{\boldsymbol{A}}^{k-1} \quad \textit{et} \quad [\![\pi_{\widehat{\boldsymbol{A}},k-1}(u)]\!](\perp_{k-1}) = \text{CONTENU}_{k-1}([\![u]\!](\perp_k)).$$

Le lemme ci-dessus découle du fait qu'une instruction de niveau l s'applique sur la pile de niveau l la plus haute.

Démonstration. Nous prouvons le lemme ci-dessus par induction sur la taille de u.

Cas initial. Pour $u = \varepsilon$, l'affirmation est trivialement vraie.

Pas d'induction. Supposons un mot $u \in \mathcal{V}_{\boldsymbol{A}}^k$ tel que

$$[\![\pi_{\widehat{A},k-1}(u)]\!](\perp_k) = \text{CONTENU}_{k-1}([\![u]\!](\perp_k)).$$

Soit un symbole $a \in \widehat{A}_{1,k}$ tel que $ua \in \mathcal{V}_{\boldsymbol{A}}^k$. Nous montrons que

$$[\![\pi_{\widehat{A},k-1}(ua)]\!](\perp_{k-1}) = \text{CONTENU}_{k-1}([\![ua]\!](\perp_k)).$$

Soit la k-pile $\Omega = [\![u]\!](\perp_k)$. Considérons sa factorisation $\Omega = c[\omega]\Omega_0$. Nous avons ici $\text{CONTENU}_{k-1}(\Omega) = \omega$. Nous considérons trois cas différents selon l'alphabet auquel appartient a.

— Cas 1 : $a \in \widehat{A_l}$ avec $l \leq k-1$.
Par définition de l'instruction COPIE, nous avons

$$[\![ua]\!](\perp_k) = [\![a]\!](\Omega) = c[[\![a]\!](\omega)]\Omega_0.$$

Par conséquent, $\text{CONTENU}_{k-1}([\![ua]\!](\perp_k)) = [\![a]\!](\omega)$ et puisque $[\![a]\!](\Omega)$ est définie, la $(k-1)$-pile $[\![a]\!](\omega)$ est donc aussi définie. De plus, puisque $l \leq k-1$, nous avons $\pi_{\widehat{A}_{1,k-1}}(ua) = \pi_{\widehat{A}_{1,k-1}}(u)a$ et par conséquent,

$$\begin{aligned}[\![\pi_{\widehat{A}_{1,k-1}}(ua)]\!](\perp_{k-1}) &= [\![\pi_{\widehat{A}_{1,k-1}}(u)a]\!](\perp_{k-1}) = [\![a]\!]([\![\pi_{\widehat{A}_{1,k-1}}(u)]\!](\perp_{k-1})) = [\![a]\!](\omega) \\ &= \text{CONTENU}_{k-1}([\![ua]\!](\perp_k)).\end{aligned}$$

— Cas 2 : $a \in A_k$.
Dans ce cas, nous avons par définition de l'instruction COPIE

$$[\![ua]\!](\perp_k) = [\![a]\!]([\![u]\!](\perp_k)) = a[\omega]c[\omega]\Omega_0 \quad \text{et} \quad \text{CONTENU}_{k-1}([\![ua]\!](\perp_k)) = \omega.$$

Puisque $a \in A_k$, nous avons aussi $\pi_{\widehat{A},k-1}(ua) = \pi_{\widehat{A},k-1}(u)$ et par conséquent

$$[\![\pi_{\widehat{A},k-1}(ua)]\!](\perp_{k-1}) = [\![\pi_{\widehat{A},k-1}(u)]\!](\perp_{k-1}) = \omega = \text{CONTENU}_{k-1}([\![ua]\!](\perp_k)).$$

— Cas 3 : $a = \bar{b} \in \overline{A_k}$.
Puisque par hypothèse le $u\bar{b}$ appartient à $\mathcal{V}_{\boldsymbol{A}}^k$, la pile $\overline{\text{COPIE}_b}(\Omega)$ est définie et la pile $\Omega = [\![u]\!](\perp_k)$ est donc de la forme $b[\omega]c[\omega]\Omega_0$. Nous avons donc $\text{CONTENU}_{k-1}(\Omega) = \omega$ et de façon similaire au cas précédent, nous avons $\pi_{\widehat{A}_{1,k-1}}(u\bar{b}) = \pi_{\widehat{A}_{1,k-1}}(u)$ et

$$[\![\pi_{\widehat{A},k-1}(u\bar{b})]\!](\perp_{k-1}) = [\![\pi_{\widehat{A},k-1}(u)]\!](\perp_{k-1}) = \text{CONTENU}_{k-1}([\![u\bar{b}]\!](\perp_k)).$$

□

Lemme 7.13. *Soit $k > 1$, nous avons*

$$\pi_{\widehat{A},k-1}(\mathcal{V}_{\boldsymbol{A}}^k) = \mathcal{V}_{\boldsymbol{A}}^{k-1} \tag{A}$$
$$\pi_{\widehat{A},k-1}(\mathcal{M}_{\boldsymbol{A}}^k) = \mathcal{M}_{\boldsymbol{A}}^{k-1}. \tag{B}$$

Démonstration. Le lemme ci-dessus est une application des lemmes 6.24 et 7.12.

(A)D'après le lemme 7.12, pour tout $u \in \mathcal{V}_{\boldsymbol{A}}^k$, $\pi_{\widehat{A},k-1}(u) \in \mathcal{V}_{\boldsymbol{A}}^{k-1}$. Réciproquement, pour tout $u \in \mathcal{V}_{\boldsymbol{A}}^{k-1}$ il existe $v \in \mathcal{V}_{\boldsymbol{A}}^k$ tel que $\pi_{\widehat{A},k-1}(v) = u$. Il suffit de considérer $v = u$.

(B)D'après le lemme 6.24, un mot u appartient à $\mathcal{M}_{\boldsymbol{A}}^k$ si et seulement si $u \in \mathcal{V}_{\boldsymbol{A}}^k$ et $u \in \mathcal{T}_{A_{1,k}}$. Nous avons ainsi $\pi_{\widehat{A},k-1}(u) \in \mathcal{V}_{\boldsymbol{A}}^{k-1}$ et $\pi_{\widehat{A}_{1,k-1}}(u) \in \mathcal{T}_{A_{1,k-1}}$. C'est-à-dire $\pi_{\widehat{A},k-1}(u) \in \mathcal{M}_{\boldsymbol{A}}^{k-1}$. Réciproquement, pour tout $u \in \mathcal{M}_{\boldsymbol{A}}^{k-1}$ il existe $v \in \mathcal{M}_{\boldsymbol{A}}^k$ tel que $\pi_{\widehat{A},k-1}(v) = u$. Il suffit de prendre $v = u$. Ceci conclut la preuve du lemme ci-dessus. □

Lemme 7.14. *Soit $k > 1$. Tout mot $u \in \mathcal{M}_{\boldsymbol{A}}^k$ tel que $|u|_{A_k} \neq 0$ admet une décomposition*

$$u = xay\bar{a}z \quad \textit{avec } a \in A_k, y \in \mathcal{T}_{A_{1,k-1}} \textit{ et } xz \in \mathcal{M}_{\boldsymbol{A}}^k.$$

Démonstration. Soit $u \in \mathcal{M}_{\boldsymbol{A}}^k$. D'après le lemme 7.11, $\pi_{\widehat{A_k}}(u) \in \mathcal{D}_{A_k}$. Le mot u admet donc une décomposition

$$u = xay\bar{a}z \quad \text{avec } a \in A_k, y \in \widehat{A}_{1,k-1}^* \text{ et } \pi_{\widehat{A_k}}(xz) \in \mathcal{D}_{A_k}.$$

Il nous reste à montrer que le mot y appartient à $\mathcal{T}_{A_{1,k-1}}$ et que xz appartient à $\mathcal{M}_{\boldsymbol{A}}^k$.

Soit la k-pile $\Omega = [\![x]\!](\perp_k)$. Considérons sa factorisation $\Omega = c[\omega]\Omega_0$. Nous avons alors

$$[\![xa]\!](\perp_k) = \text{COPIE}_a(\Omega) = a[\omega]c[\omega]\Omega_0.$$

Si y n'appartient pas à $\mathcal{T}_{A_{1,k-1}}$, c'est-à-dire si $\rho(y) \neq \varepsilon$, alors forcément $[\![xay]\!](\perp_k) \neq [\![xa]\!](\perp_k)$; c'est-à-dire

$$[\![xay]\!](\perp_k) = a[[\![y]\!](\omega)]c[\omega]\Omega_0 = a[\omega']c[\omega]\Omega_0 \quad \text{avec } \omega' \neq \omega.$$

Puisque ω' est différent de ω, par définition de l'instruction $\overline{\text{COPIE}}$, nous avons

$$[\![xay\bar{a}]\!](\perp_k) = \overline{\text{COPIE}_a}(c[\omega']a[\omega]\Omega_0) = \textit{indéfini}.$$

Ceci contredit l'hypothèse que $u \in \mathcal{M}_{\boldsymbol{A}}^k$. Nous avons donc

$$y \in \mathcal{T}_{A_{1,k-1}}$$

et par conséquent $[\![xay]\!](\perp_k) = [\![xa]\!]$ (puisque $\rho(y) = \varepsilon$). Ceci implique aussi que $[\![xay\bar{a}]\!] = \Omega$. Puisque par hypothèse, $[\![u]\!](\perp_k) = \perp_k$, il en découle que $[\![z]\!](\Omega) = \perp_k$ et par conséquent $[\![xz]\!](\perp_k) = \perp_k$, c'est-à-dire $xz \in \mathcal{M}_{\boldsymbol{A}}^k$. Ceci conclut la preuve du lemme ci-dessus. □

7.4.3 Caractérisations des générateurs de Maslov

Nous prouvons maintenant que les générateurs de Maslov sont des itérations du langage $\widetilde{\mathcal{D}}$ par des projections. Considérons en guise d'exemple simple la séquence $\boldsymbol{A} = A_1, A_2, A_3$ avec $A_1 = \{a\}, A_2 = \{b\}$ et $A_3 = \{c\}$. Le mot $u = aab\bar{a}c\bar{a}aca\bar{b}\bar{a}\bar{a}$ appartient à $\mathcal{M}^3_{\boldsymbol{A}}$. Nous pouvons observer que

$$\begin{aligned} u &= aab\bar{a}c\bar{a}aca\bar{b}\bar{a}\bar{a} \in \widetilde{\mathcal{D}}_{A_3, A_2 \cup A_1}; \\ \pi_{\widehat{\boldsymbol{A}},2}(u) &= aab\bar{a}\bar{a}aa\bar{b}\bar{a}\bar{a} \in \widetilde{\mathcal{D}}_{A_2, A_1}; \\ \pi_{\widehat{\boldsymbol{A}},1}(u) &= aa\bar{a}\bar{a}aa\bar{a}\bar{a} \in \widetilde{\mathcal{D}}_{A_1, \emptyset} = \mathcal{D}_{A_1}. \end{aligned}$$

L'appartenance à $\widetilde{\mathcal{D}}$ est respectée à chaque niveau. Commençons par prouver que chaque $\mathcal{M}^k_{\boldsymbol{A}}$ est un sous-ensemble de $\widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$.

Lemme 7.15. *Pour tout $k > 1$: $\mathcal{M}^k_{\boldsymbol{A}} \subseteq \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$.*

Démonstration. Soit un mot $u \in \mathcal{M}^k_{\boldsymbol{A}}$. Nous prouvons par induction structurelle sur le nombre de symbole de $\widehat{A_k}$ dans les mots de $\mathcal{M}^k_{\boldsymbol{A}}$. Remarquons aussi que chaque mot de $\mathcal{M}^k_{\boldsymbol{A}}$ contient un nombre pair de symbole de $\widehat{A_k}$.

Affirmation 1. Pour tout entier i pair, et $u \in \mathcal{M}^k_{\boldsymbol{A}}$, si $u_{|\widehat{A_k}|} = i$, alors $u \in \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$.

Preuve de l'affirmation 1. Par induction sur i :

Cas initial. Le cas initial $i = 0$ est vrai d'après le lemme 6.20. En effet, si u ne contient pas de symbole de $\widehat{A_k}$, alors $\pi_{\widehat{A}_{1,k-1}}(u) = u$. Puisque d'après le lemme 6.20, $\mathcal{M}^{k-1}_{\boldsymbol{A}} \subseteq \mathcal{T}_{A_{1,k-1}}$, nous obtenons alors $u \in \mathcal{T}_{A_{1,k-1}}$, c'est-à-dire $u \in \widetilde{\mathcal{D}}_{\emptyset, A_{1,k-1}}$, c'est-à-dire $u \in \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$.

Pas d'induction. Supposons que l'affirmation ci-dessus soit vraie pour un certain i. Montrons qu'elle est vraie pour $i + 2$.

Soit alors un mot $u \in \mathcal{M}^k_{\boldsymbol{A}}$ contenant $i + 2$ symboles de $\widehat{A_k}$. D'après le lemme 7.14, le mot u admet une factorisation

$$u = xay\bar{a}z \quad \text{avec } a \in A_k, y \in \mathcal{T}_{A_{1,k-1}} \text{ et } xz \in \mathcal{M}^k_{\boldsymbol{A}}.$$

Le mot $ay\bar{a}$ appartient donc à $\widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$, c'est-à-dire $\widetilde{\rho}_{A_k, A_{1,k-1}}(ay\bar{a}) = \varepsilon$; et par hypothèse d'induction, nous avons aussi $\widetilde{\rho}_{A_k, A_{1,k-1}}(xz) = \varepsilon$. Nous avons donc $\widetilde{\rho}_{A_k, A_{1,k-1}}(u) = \varepsilon$ et donc $u \in \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$. □

L'affirmation ci-dessus induit que tout mot de $\mathcal{M}^k_{\boldsymbol{A}}$ appartient à $\widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$. Ceci conclut la preuve du lemme ci-dessus. □

Lemme 7.16. *Pour tout $k > 1$: $\mathcal{M}^k_{\boldsymbol{A}} \subseteq \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}} \cap \pi^{-1}_{\widehat{A},k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}})$.*

Démonstration. D'après le lemme 7.13, nous avons $\mathcal{M}^k_{\boldsymbol{A}} = \mathcal{M}^k_{\boldsymbol{A}} \cap \pi^{-1}_{\widehat{A},k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}})$ et d'après le lemme 7.15, nous avons $\mathcal{M}^k_{\boldsymbol{A}} \subseteq \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$. Nous obtenons ainsi

$$\mathcal{M}^k_{\boldsymbol{A}} \;=\; \mathcal{M}^k_{\boldsymbol{A}} \cap \pi^{-1}_{\widehat{A},k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}}) \;\subseteq\; \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}} \cap \pi^{-1}_{\widehat{A},k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}}).$$

□

Lemme 7.17. *Pour tout $k > 1$: $\widetilde{\mathcal{D}}_{A_k,A_{1,k-1}} \cap \pi^{-1}_{\widehat{A},k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}}) \subseteq \mathcal{M}^k_{\boldsymbol{A}}$.*

Démonstration. Nous montrons le lemme ci-dessus par induction structurelle sur le nombre de symbole de $\widehat{A_k}$ dans un mot.

Affirmation 1. Pour tout entier pair $l \geq 0$, et pour tout mot $u \in \widehat{A}_{1,k}$ tel que $|u|_{a\in\widehat{A_k}} = l$; si $u \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$ et $\pi_{\widehat{A}_{1,k-1}}(u) \in \mathcal{M}^{k-1}_{\boldsymbol{A}}$ alors $u \in \mathcal{M}^k_{\boldsymbol{A}}$.

Preuve de l'affirmation 1. Par induction sur l.

Cas initial. Le cas initial est $|u|_{a\in\widehat{A_k}} = 0$. Dans ce cas, $\pi_{\widehat{A}_{1,k-1}}(u) = u$. Par conséquent, si le mot $\pi_{\widehat{A}_{1,k-1}}(u)$ appartient à $\mathcal{M}^{k-1}_{\boldsymbol{A}}$, alors $u \in \mathcal{M}^{k-1}_{\boldsymbol{A}}$ et ainsi $u \in \mathcal{M}^k_{\boldsymbol{A}}$.

Pas d'induction. Supposons maintenant l'affirmation vraie pour un certain l et montrons qu'elle est vraie pour $l+2$.

Soit un mot u de $\widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$ tel que $\pi_{\widehat{A}_{1,k-1}}(u)$ appartient à $\mathcal{M}^{k-1}_{\boldsymbol{A}}$ et tel que $|u|_{a\in\widehat{A_k}} = l+2$. Puisque u appartient à $\widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$ et que $|u|_{\widehat{A_k}} \neq 0$ il admet une factorisation

$$u = u_1 a v \bar{a} u_2 \qquad \text{avec } a \in A_k;\ \ v \in \mathcal{T}_{A_{1,k-1}};\ \ u_1 v u_2 \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}};\ \ \text{et } |u_1 v u_2|_{\widehat{A_k}} = l.$$

De plus, puisque $a \in A_k$, nous avons $\pi_{\widehat{A}_{1,k-1}}(u_1 v u_2) = \pi_{\widehat{A}_{1,k-1}}(u)$ et par hypothèse d'induction, nous avons alors $\pi_{\widehat{A}_{1,k-1}}(u_1 v u_2) \in \mathcal{M}^{k-1}_{\boldsymbol{A}}$. Encore une fois par hypothèse d'induction, le mot $u_1 v u_2$ appartient à $\mathcal{M}^k_{\boldsymbol{A}}$ puisque celui-ci appartient à $\widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$ et que $\pi_{\widehat{A}_{1,k-1}}(u_1 v u_2) \in \mathcal{M}^{k-1}_{\boldsymbol{A}}$.

Soit alors la k-pile $\Omega = [\![u_1]\!](\perp_k)$. Considérons une factorisation $b[\omega]\Omega_0$ de celle-ci. Soit aussi $\Omega_1 = [\![v]\!](\Omega)$. Puisque le mot v appartient à $\widehat{A}_{1,k-1}$, la suite d'instruction $[\![v]\!]$ est donc de niveau $k-1$ et par définition des instructions COPIE et $\overline{\text{COPIE}}$, nous avons alors

$$[\![v]\!](\Omega) = b[[\![v]\!](\omega)]\Omega_0.$$

Puisque $u_1 v u_2 \in \mathcal{M}^k_{\boldsymbol{A}}$, la pile $[\![v]\!](\Omega)$ est définie. De plus, $\rho(v) = \varepsilon$ et par conséquent

$$[\![v]\!](\Omega) = \Omega = b[\omega]\Omega_0.$$

C'est à dire $[\![v]\!](\omega) = \omega$. Finalement, puisque $u_1 v u_2 \in \mathcal{M}^k_{\boldsymbol{A}}$, nous avons $[\![u_1 v u_2]\!](\perp_k) = \perp_k$ et donc

$$[\![u_2]\!]([\![u_1 v]\!](\perp_k)) = [\![u_2]\!](\Omega) = \perp_k.$$

Nous pouvons vérifier que le mot $u_1 a v \bar{a} u_2$ appartient à $\mathcal{M}^k_{\boldsymbol{A}}$ comme suit :

$$\perp_k \xrightarrow{u_1} b[\omega]\Omega_0 \xrightarrow{a} a[\omega]b[\omega]\Omega_0 \xrightarrow{v} a[\omega]b[\omega]\Omega_0 \xrightarrow{\bar{a}} b[\omega]\Omega_0 \xrightarrow{u_2} \perp_k.$$

Le mot $u_1 a v \bar{a} u_2$ appartient donc à $\mathcal{M}^k_{\boldsymbol{A}}$. Ceci prouve l'affirmation ci-dessus. □

L'affirmation ci-dessus prouve le lemme. □

Proposition 7.18. *$\mathcal{M}^1_{\boldsymbol{A}} = \mathcal{D}_{A_1}$; pour tout $k > 1$: $\mathcal{M}^k_{\boldsymbol{A}} = \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k-1}} \cap \pi^{-1}_{\widehat{A},k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}})$.*

Démonstration. Par définition, $\mathcal{M}^1_{\boldsymbol{A}} = \mathcal{D}_{A_1}$. Les deux lemmes précédents montrent les deux sens de l'égalité $\mathcal{M}^k_{\boldsymbol{A}} = \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}} \cap \pi^{-1}_{A,k-1}(\mathcal{M}^{k-1}_{\boldsymbol{A}})$. □

Corollaire 7.19. *Un mot u appartient à $\mathcal{M}^k_{\boldsymbol{A}}$ si et seulement si $u \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$ et pour tout $l < k$: $\pi_{\widehat{\boldsymbol{A}},l}(u) \in \widetilde{\mathcal{D}}_{A_l,A_{1,l-1}}$.*

7.5 Dominance du générateur de Maslov

Nous montrons dans cette section que pour tout $k \geq 1$, la famille ε-DYCK^k est incluse dans LANG^k. Pour ce faire, il nous suffira de montrer que tout langage de ε-DYCK^k peut être obtenu par transduction rationnelle depuis un langage $\mathcal{M}^k_{\boldsymbol{A}}$. On parle de dominance rationnelle dans ce cas. Montrer l'inclusion de ε-DYCK^k dans LANG^k par un automate présume de savoir coder les mots des langages de ε-DYCK^k par des suites d'instructions de $\mathcal{M}^k_{\boldsymbol{A}}$. Ce codage est précisément la transduction rationnelle que nous décrivons ici. Les automates ne sont donc pas nécessaires.

Lemme 7.20. *Soit $f : \widehat{A_2}^* \longrightarrow \widehat{A_1}^*$ un homomorphisme ε-sûr et $\mu : \widehat{A_2} \longrightarrow \widehat{A_2 \cup A_1}^*$ l'homomorphisme tel que pour tout $a \in A_2$:*

$$\mu(a) = af(a); \qquad \mu(\bar{a}) = f(\bar{a})\bar{a}.$$

Alors $\mu(\mathcal{D}_{A_2}) \subseteq \widetilde{\mathcal{D}}_{A_2,A_1}$.

Démonstration. En effet, le langage $\mu(\mathcal{D}_{A_2})$ est décrit par la grammaire

$$S \longrightarrow \varepsilon + SS + \sum_{a \in A_2} af(a)Sf(\bar{a})\bar{a}.$$

Puisque l'homomorphisme f est ε-sûr, nous avons $\rho(f(a)f(\bar{a})) = \varepsilon$ et par conséquent

$$\widetilde{\rho}_{A_2,A_1}(af(a)f(\bar{a})\bar{a}) = \varepsilon.$$

Le langage décrit par cette équation est donc clairement un sous-ensemble de $\widetilde{\mathcal{D}}_{A_2,A_1}$. □

Proposition 7.21. *Pour tout $k \geq 1$ et $X \in \varepsilon$-DYCK^k, il existe une séquence d'alphabets $\boldsymbol{A}$ et un homomorphisme ε-sûr h_k tels que $X = h_k^{-1}(\mathcal{M}^k_{\boldsymbol{A}})$.*

Démonstration. Par induction sur k.

Cas initial. L'affirmation est trivialement vraie pour $k = 1$ puisque $\mathcal{M}^1_{\boldsymbol{A}} = \mathcal{D}_{A_1}$.

Pas d'induction. Supposons l'affirmation vraie pour un certain $k \geq 1$. Soit alors un langage $X_k \subseteq \widehat{\Sigma_k}^*$ de ε-DYCK^k et un homomorphisme $h_k : \widehat{\Sigma_k}^* \longrightarrow \widehat{A}^*_{1,k}$ tel que $X_k = h_k^{-1}(\mathcal{M}^k_{\boldsymbol{A}})$. Soit maintenant un homomorphisme ε-sûr $g : \widehat{\Sigma_{k+1}}^* \longrightarrow \widehat{\Sigma_k}^*$ et le langage $X_{k+1} = \mathcal{D}_{\Sigma_{k+1}} \cap g^{-1}(X_k)$ appartenant à ε-DYCK^{k+1}. Nous montrons l'existence d'un homomorphisme ε-sûr h_{k+1} tel que

$$X_{k+1} = h_{k+1}^{-1}(\mathcal{M}^{k+1}_{\boldsymbol{A}'}).$$

Prenons $A_{k+1} = \Sigma_{k+1}$. Soit $\boldsymbol{A}'$ l'extension de $\boldsymbol{A}$ par A_{k+1}. Soit aussi l'homomorphisme ε-sûr $h_{k+1} : \widehat{A}^*_{k+1} \longrightarrow \widehat{A'}^*_{1,k+1}$ tel que pour tout $\alpha \in A_{k+1}$:

$$h_{k+1}(\alpha) = \alpha h_k \circ g(\alpha); \quad h_{k+1}(\bar{\alpha}) = h_k \circ g(\bar{\alpha})\bar{\alpha}.$$

Remarque 7.22. $\pi_{\widehat{A_{k+1}}} \circ h_{k+1} = \mathrm{id}$ *et* $\pi_{\widehat{\boldsymbol{A}},k} \circ h_{k+1} = h_k \circ g$.

Affirmation 1. ${h_{k+1}}^{-1}(\widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}) = \mathcal{D}_{A_{k+1}}$.

Preuve de l'affirmation 1. Il s'agit de montrer que pour tout $u \in \widehat{A_{k+1}}^*$: $u \in \mathcal{D}_{A_{k+1}}$ si et seulement si $h_{k+1}(u) \in \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$.

Soit alors un mot $u \in \widehat{A_{k+1}}^*$. Si $u \in \mathcal{D}_{A_{k+1}}$, alors d'après le lemme 7.20, $h_{k+1}(u) \in \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$. Si $h_{k+1}(u) \in \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$, alors, par définition de $\widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$, nous avons $\pi_{\widehat{A_{k+1}}} \circ h_{k+1}(u) \in \mathcal{D}_{A_{k+1}}$. Ce qui d'après la remarque 7.22 implique que $u \in \mathcal{D}_{A_{k+1}}$. Ceci prouve l'affirmation ci-dessus. □

D'après la proposition 7.18, $\mathcal{M}^{k+1}_{\boldsymbol{A}'} = \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}} \cap \pi^{-1}_{\widehat{\boldsymbol{A}},k}(\mathcal{M}^A_k)$. Il s'en suit que

$$\begin{aligned}
{h_{k+1}}^{-1}(\mathcal{M}^{k+1}_{\boldsymbol{A}}) &= {h_{k+1}}^{-1}(\widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}} \cap \pi^{-1}_{\widehat{\boldsymbol{A}},k}(\mathcal{M}^k_{\boldsymbol{A}})) \\
&= {h_{k+1}}^{-1}(\widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}) \cap {h_{k+1}}^{-1}(\pi^{-1}_{\widehat{\boldsymbol{A}},k}(\mathcal{M}^A_k)) && \text{distributivité} \\
&= \mathcal{D}_{A_{k+1}} \cap g^{-1}({h_k}^{-1}(\mathcal{M}^k_{\boldsymbol{A}})) && \text{affirmation 1 et remarque 7.22} \\
&= \mathcal{D}_{A_{k+1}} \cap g^{-1}(X) = X_{k+1} && \text{hypothèse d'induction.}
\end{aligned}$$

□

Corollaire 7.23. *Pour tout $k \geq 1$: ε-DYCK$^k \subseteq$ LANG^k.*

Démonstration. D'après la proposition 7.21, tout langage $X \in \varepsilon$-DYCK^k peut s'exprimer comme $f^{-1}(\mathcal{M}^k_{\boldsymbol{A}})$. Puisque tout langage $\mathcal{M}^k_{\boldsymbol{A}}$ appartient à LANG^k et que LANG^k est une famille agréable (Théorème 6.1), il s'en suit que tout langage $X \in \varepsilon$-DYCK^k appartient à LANG^k. □

Corollaire 7.24. *Pour tout $k \geq 1$: $\mathscr{C}(\varepsilon$-DYCK$^k) \subseteq$ LANG^k.*

7.6 Dominance des langages de Dyck

Nous montrons dans cette section que pour tout $k \geq 1$, la famille ε-DYCK^k domine rationnellement la famille LANG^k. Encore une fois, il nous suffit de montrer que chaque langage $\mathcal{M}^k_{\boldsymbol{A}}$ est l'image par transduction rationnelle d'un langage de ε-DYCK^k. Nous utilisons pour ce faire notre caractérisation des générateurs de Maslov comme itérations du langage $\widetilde{\mathcal{D}}$. Reprenons notre exemple simple. Soit $A_1 = \{a\}$ et $A_2 = \{b\}$. Le mot $u = ab\bar{a}a\bar{b}\bar{a}$ appartient à $\mathcal{M}^2_{\boldsymbol{A}}$ et n'appartient pas à $\mathcal{D}_{A_2 \cup A_1}$. Pour le représenter comme un mot de Dyck, nous pouvons remplacer le facteur bilatéral $\bar{a}a$ par $a'\bar{a}'$. On obtient ainsi le mot

$$aba'\bar{a}'\bar{b}\bar{a}.$$

Le fait de substituer un symbole (bien choisi) par un symbole de *polarité* inverse appartenant à une copie de l'alphabet nous permet non seulement, de transformer des mots de $\widetilde{\mathcal{D}}$ en mots de $\mathcal{D}$, mais aussi de conserver l'information sur le symbole d'origine. Nous introduisons formellement cette notion dans ce qui suit.

7.6.1 Démarquages et d-projections

Marqueurs. Nous appelons *marqueur* un symbole β qui n'appartient à aucun alphabet défini dans le contexte. Pour tout alphabet A et marqueur β, $\beta(A)$ désigne l'alphabet $\{\beta\} \times A$.

Si l'alphabet A est marqué par plusieurs symboles, disons β_1 et β_2, les éléments de $\beta_2(\beta_1(A))$ sont notés $(\beta_2\beta_1, a)$ plutôt que $(\beta_2, (\beta_1, a))$.

Définition 7.25. *Soit A_1 et A_2 deux alphabets disjoints et β un marqueur. Nous définissons les homomorphismes* symétriques *suivants :*

— *(démarquage)* $\lambda_{A_2,A_1,\beta} : (A_2 \cup \widehat{A_1 \cup \beta(A_1)})^* \longrightarrow (\widehat{A_2 \cup A_1})^*$ *tel que*

$$\begin{cases} \lambda_{A_2,A_1,\beta}(a) = a & \forall a \in A_2 \cup A_1, \\ \lambda_{A_2,A_1,\beta}((\beta, a)) = \bar{a} & \forall (\beta, a) \in \beta(A_1). \end{cases}$$

— *(d-projection)*[6] $\sigma_{A_2,A_1,\beta} : (A_2 \cup \widehat{A_1 \cup \beta(A_1)})^* \longrightarrow \widehat{A_1}^*$ *tel que* $\sigma_{A_2,A_1,\beta} = \pi_{\widehat{A_1}} \circ \lambda_{A_2,A_1,\beta}$ *:*

$$\begin{cases} \sigma_{A_2,A_1,\beta}(a) = \varepsilon & \forall a \in A_2, \\ \sigma_{A_2,A_1,\beta}(a) = a & \forall a \in A_1, \\ \sigma_{A_2,A_1,\beta}((\beta, a)) = \bar{a} & \forall (\beta, a) \in \beta(A_1). \end{cases}$$

Nous prouvons quelques propriétés des fonctions de démarquages et de d-projections.

Lemme 7.26. *Soient A_1 et A_2 deux alphabets disjoints et β un marqueur. Alors*

$$\widetilde{\mathcal{D}}_{A_2,A_1} = \lambda_{A_2,A_1,\beta}(\mathcal{D}_{A_2 \cup A_1 \cup \beta(A_1)}).$$

Démonstration. Le langage $\mathcal{D}_{A_2 \cup A_1 \cup \beta(A_1)}$ est généré par la grammaire

$$S \longrightarrow \varepsilon + SS + \sum_{a \in A_2 \cup A_1 \cup \beta(A_1)} aS\bar{a}.$$

$\lambda_{A_2,A_1,\beta}(\mathcal{D}_{A_2 \cup A_1 \cup \beta(A_1)})$ est alors décrit par la grammaire

$$S' \longrightarrow \varepsilon + S'S' + \sum_{a \in A_2 \cup A_1 \cup \beta(A_1)} \lambda_{A_2,A_1,\beta}(a) S' \lambda_{A_2,A_1,\beta}(\bar{a})$$

c'est à dire

$$S' \longrightarrow \varepsilon + S'S' + \sum_{a \in A_2 \cup A_1 \cup \overline{A_1}} aS'\bar{a}$$

et cette dernière décrit le langage $\widetilde{\mathcal{D}}_{A_2,A_1}$. □

Le but des marquages est donc de pouvoir décrire des réductions mixtes (ou simplement bilatérales) comme des réductions unilatérales moyennant une duplication de l'alphabet : Le démarquage λ permet d'obtenir un mot bilatéralement réductible depuis sa description sous forme de mot marqué unilatéralement réductible.

Une conséquence du lemme 7.26 est le lemme suivant.

Lemme 7.27. *Soient A_1 et A_2 deux alphabets disjoints et β un marqueur. Alors*

$$\mathcal{T}_{A_1} = \sigma_{A_2,A_1,\beta}(\mathcal{D}_{A_2 \cup A_1 \cup \beta(A_1)}).$$

6. Pour « projection démarquée ».

Démonstration. D'après le lemme 7.26, nous avons $\widetilde{\mathcal{D}}_{A_2,A_1} = \lambda_{A_2,A_1,\beta}(\mathcal{D}_{A_2\cup A_1\cup\beta(A_1)})$; et par conséquent

$$\pi_{\widehat{A}_1}(\widetilde{\mathcal{D}}_{A_2,A_1}) = \pi_{\widehat{A}_1}\circ\lambda_{A_2,A_1,\beta}(\mathcal{D}_{A_2\cup A_1\cup\beta(A_1)}).$$

Par définition, $\pi_{\widehat{A}_1}(\widetilde{\mathcal{D}}_{A_2,A_1}) = \widetilde{\mathcal{D}}_{\emptyset,A_1} = \mathcal{T}_{A_1}$ et $\pi_{\widehat{A}_1}\circ\lambda_{A_2,A_1,\beta} = \sigma_{A_2,A_1,\beta}$. Nous obtenons ainsi

$$\mathcal{T}_{A_2,A_1} = \sigma_{A_2,A_1,\beta}(\mathcal{D}_{A_2\cup A_1\cup\beta(A_1)}).$$

□

7.6.2 Séquence d'alphabets marqués

Afin de marquer les mots de $\mathcal{M}^k_{\boldsymbol{A}}$ sous forme de mots appartenant à des langages de ε-DYCK^k, nous avons besoin de considérer des marquages sur plusieurs niveaux. Nous introduisons alors ici la notion de séquences d'alphabets marqués.

Considérons dans toute cette section une séquence de marqueurs $\boldsymbol{\beta} = \beta_1, \beta_2, \ldots$. Celle-ci sera toujours considérée assez grande.

Définition 7.28 (Séquence d'alphabets marqués)**.** *Étant donné une séquence d'alphabets disjoints $\boldsymbol{A}$, nous désignons par $A_k^{\boldsymbol{\beta}}$ l'alphabet défini comme suit :*

$$A_1^{\boldsymbol{\beta}} = A_1\,; \qquad \text{pour tout } k \geq 1 : A_{k+1}^{\boldsymbol{\beta}} = A_k^{\boldsymbol{\beta}} \cup \beta_k(A_k^{\boldsymbol{\beta}}) \cup A_{k+1}.$$

Exemple 7.29. *Si $A_1 = \{a\}, A_2 = \{b\}, A_3 = \{c\}$, alors*

$$A_1^{\boldsymbol{\beta}} = \{a\}, \quad A_2^{\boldsymbol{\beta}} = \{a, (\beta_1, a), b\}, \quad A_3^{\boldsymbol{\beta}} = \{a, (\beta_1, a), b, (\beta_2, a), (\beta_2\beta_1, a), (\beta_2, b), c\}.$$

Dans ce qui suit, nous considérons une séquence d'alphabet $\boldsymbol{A}$ fixée. Celle-ci sera toujours considérée assez grande.

Définition 7.30 (Séquence de démarquages)**.** *Soit $\boldsymbol{\lambda} = (\lambda_k)_{k\geq 1}$ la séquence d'homomorphismes ε-sûrs sur $\boldsymbol{A}$ telle que pour tout $k \geq 1$: λ_k est le démarquage $\lambda_{A_{k+1},A_k^{\boldsymbol{\beta}},\beta_k}$. (Définition 7.25) : c.à.d l'homomorphisme symétrique $\lambda_k : \widehat{A_{k+1}^{\boldsymbol{\beta}}}^* \longrightarrow \widehat{A_{k+1}\cup A_k^{\boldsymbol{\beta}}}^*$ tel que*

$$\begin{cases} \lambda_k(a) = a, & \forall a \in A_{k+1}; \\ \lambda_k(a) = a, & \forall a \in A_k^{\boldsymbol{\beta}}; \\ \lambda_k((\beta_k, a)) = \bar{a}, & \forall (\beta_k, a) \in \beta_k(A_k^{\boldsymbol{\beta}}). \end{cases}$$

Définition 7.31 (Séquence de d-projections)**.** *Soit $\boldsymbol{\sigma} = (\sigma_k)_{k\geq 1}$ la séquence d'homomorphismes ε-sûrs sur $\boldsymbol{A}$ telle que pour tout $k \geq 1$: σ_k est la d-projection $\sigma_{A_{k+1},A_k^{\boldsymbol{\beta}},\beta_k}$ (Définition 7.25) : c.à.d l'homomorphisme symétrique $\sigma_k : \widehat{A_{k+1}^{\boldsymbol{\beta}}}^* \longrightarrow \widehat{A_k^{\boldsymbol{\beta}}}^*$ tel que $\sigma_k = \pi_{\widehat{A_k^{\boldsymbol{\beta}}}} \circ \lambda_k$:*

$$\begin{cases} \sigma_k(a) = \varepsilon, & \forall a \in A_{k+1}; \\ \sigma_k(a) = a, & \forall a \in A_k^{\boldsymbol{\beta}}; \\ \sigma_k((\beta_k, a)) = \bar{a}, & \forall (\beta_k, a) \in \beta_k(A_k^{\boldsymbol{\beta}}). \end{cases}$$

Remarque 7.32. *Conformément à notre convention sur les suites d'homomorphismes, $\sigma_{1,k}$ désigne la composition $\sigma_1 \circ \sigma_2 \cdots \circ \sigma_k$; de même, $\lambda_{1,k} = \lambda_1 \circ \lambda_2 \cdots \circ \lambda_k$.*

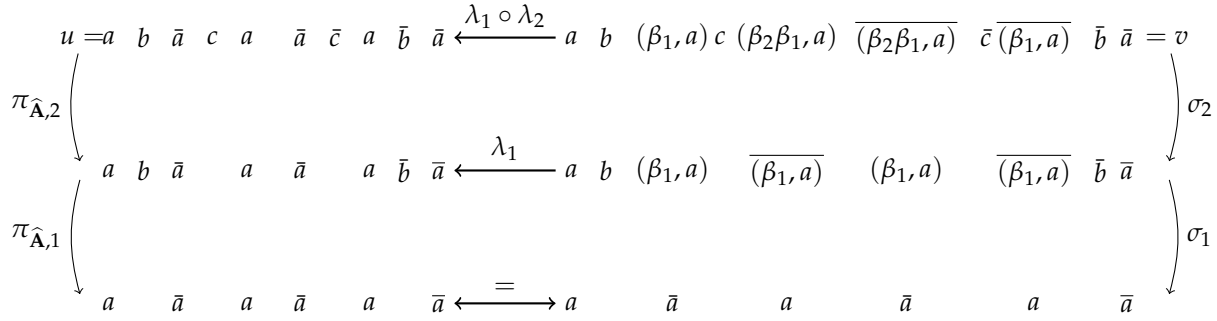


FIGURE 7.2 – À gauche : un mot u de $\mathcal{M}^3_{\boldsymbol{A}}$ avec $A_1 = \{a\}, A_2 = \{b\}, A_3 = \{c\}$. À droite : un mot v de $\mathcal{D}^3_{\boldsymbol{A}}$ correspondant. Nous remarquons que certains symboles sont marqués plusieurs fois afin de préserver l'appartenance au langage de Dyck sur tous les niveaux.

Exemple 7.33. *Supposons $A_1 = \{a\}, A_2 = \{b\}, A_3 = \{c\}$. L'image d'un mot $u \in \widehat{A_3^{\boldsymbol{\beta}}}^*$ par $\lambda_{1,2}$ est illustrée comme suit.*

$$\begin{aligned}
u &= a \quad b \quad (\beta_1,a) \quad c \quad (\beta_2\beta_1,a) \quad \overline{(\beta_2\beta_1,a)} \quad \bar{c} \quad \overline{(\beta_1,a)} \quad \bar{b} \quad \bar{a} \\
\lambda_2(u) &= a \quad b \quad (\beta_1,a) \quad c \quad \overline{(\beta_1,a)} \quad (\beta_1,a) \quad c \quad \overline{(\beta_1,a)} \quad \bar{b} \quad \bar{a} \\
\lambda_1 \circ \lambda_2(u) &= a \quad b \quad \bar{a} \quad c \quad a \quad \bar{a} \quad c \quad a \quad \bar{b} \quad \bar{a}
\end{aligned}$$

Définition 7.34 (Itération par d-projections)**.** *« Le » langage de Dyck d'ordre k sur $\boldsymbol{A}$ est $\mathcal{D}^k_{\boldsymbol{A}} = D_{k,\boldsymbol{\sigma}}$.*

Exemple 7.35. *Si $A_1 = \{a\}, A_2 = \{b\}, A_3 = \{c\}$. Alors le mot u de l'exemple 7.33 appartient à $\mathcal{D}^3_{\boldsymbol{A}}$ comme illustré ci-dessous.*

$$\begin{aligned}
u &= a \quad b \quad (\beta_1,a) \quad c \quad (\beta_2\beta_1,a) \quad \overline{(\beta_2\beta_1,a)} \quad \bar{c} \quad \overline{(\beta_1,a)} \quad \bar{b} \quad \bar{a} \\
\sigma_2(u) &= a \quad b \quad (\beta_1,a) \quad \overline{(\beta_1,a)} \quad (\beta_1,a) \quad \overline{(\beta_1,a)} \quad \bar{b} \quad \bar{a} \\
\sigma_1 \circ \sigma_2(u) &= a \quad \bar{a} \quad a \quad \bar{a} \quad a \quad \bar{a}
\end{aligned}$$

7.6.3 Preuve de la dominance

Nous montrons ici (Proposition 7.44) que pour tout $k \geq 1$: $\mathcal{M}^k_{\boldsymbol{A}} = \lambda_{1,k-1}(\mathcal{D}^k_{\boldsymbol{A}})$ comme illustré sur la figure 7.2. Les deux sens de cette équivalences sont montrés dans les lemmes 7.38 et 7.43.

Lemme 7.36. *Pour tout $k \geq 1$, $\lambda_{1,k}(\mathcal{D}_{A^{\boldsymbol{\beta}}_{k+1}}) = \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$.*

Démonstration. Ceci découle de façon similaire au lemme 7.26 du fait que pour tout $a \in \widehat{A_k^{\boldsymbol{\beta}}}$: $\lambda_{1,k}(a) = \overline{\lambda_{1,k}((\beta_k,a))}$ et que de plus $\lambda_{1,k}(a) \in \widehat{A}_{1,k}$. Par définition, le langage $\widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$ est généré par la grammaire algébrique

$$S \longrightarrow \varepsilon + SS + \sum_{a \in A_{k+1}} aS\bar{a} + \sum_{a \in \widehat{A}_{1,k}} aS\bar{a}$$

Par définition des fonctions λ_i, la grammaire ci-dessus est équivalente à

$$S \longrightarrow \varepsilon + SS + \sum_{a \in A_{k+1}} \lambda_{1,k}(a) S \lambda_{1,k}(\bar{a}) + \sum_{a \in A_k^{\boldsymbol{\beta}} \cup \beta_k(A_k^{\boldsymbol{\beta}})} \lambda_{1,k}(a) S \lambda_{1,k}(\bar{a}).$$

Le langage $\mathcal{D}_{A_{k+1}^{\boldsymbol{\beta}}}$ étant généré par la grammaire

$$S' \longrightarrow \varepsilon + S'S' + \sum_{a \in A_{k+1} \cup A_k^{\boldsymbol{\beta}} \cup \beta_k(A_k^{\boldsymbol{\beta}})} aS'\bar{a}$$

Il s'en suit que $\lambda_{1,k}(\mathcal{D}_{A_{k+1}^{\boldsymbol{\beta}}}) = \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$. □

Lemme 7.37. *Pour tout $k > 1$ et $u \in \widehat{A_k^{\boldsymbol{\beta}}}^*$: $\lambda_{1,k} \circ \sigma_k^{-1}(u) = \pi_{\widehat{\boldsymbol{A}},k}^{-1} \circ \lambda_{1,k-1}(u)$.*

($\pi_{\widehat{\boldsymbol{A}},k}$ étant la projection de $\widehat{A}_{1,k+1}$ dans $\widehat{A}_{1,k}$.)

Démonstration. Puisque les homomorphismes σ_k et $\lambda_{1,k-1}$ sont alphabétiques, il suffit de montrer que pour tout $a \in \widehat{A_k^{\boldsymbol{\beta}}} \cup \{\varepsilon\}$: $\lambda_{1,k} \circ \sigma_k(a) = \pi_{\widehat{\boldsymbol{A}},k}^{-1} \circ \lambda_{1,k-1}(a)$. En effet, pour tout $a \in \widehat{A_k^{\boldsymbol{\beta}}}^*$:

$$\pi_{\widehat{\boldsymbol{A}},k}^{-1} \circ \lambda_{1,k-1}(a) = \widehat{A_{k+1}}^* \lambda_{1,k-1}(a) \widehat{A_{k+1}}^*.$$

De même,

$$\begin{aligned}
\lambda_{1,k} \circ \sigma_k^{-1}(a) = \lambda_{1,k}\big(\widehat{A_{k+1}}^* \{a, \overline{(\beta_k, a)}\} \widehat{A_{k+1}}^*\big) &= \lambda_{1,k-1} \circ \lambda_k\big(\widehat{A_{k+1}}^* \{a, \overline{(\beta_k, a)}\} \widehat{A_{k+1}}^*\big) \\
&= \lambda_{1,k-1}\big(\widehat{A_{k+1}}^* \{a\} \widehat{A_{k+1}}^*\big) \\
&= \widehat{A_{k+1}}^* \lambda_{1,k-1}(a) \widehat{A_{k+1}}^*.
\end{aligned}$$

De plus,

$$\lambda_{1,k} \circ \sigma_k^{-1}(\varepsilon) \;=\; \widehat{A_{k+1}}^* \;=\; \pi_{\widehat{\boldsymbol{A}},k}^{-1} \circ \lambda_{1,k-1}(\varepsilon).$$

□

Lemme 7.38. *Pour tout $k \geq 1$, $\lambda_{1,k-1}(\mathcal{D}_{\boldsymbol{A}}^k) \subseteq \mathcal{M}_{\boldsymbol{A}}^k$.*

Démonstration. L'affirmation est trivialement vraie pour $k = 1$. Prouvons par induction que pour tout $k \geq 1$: $\lambda_{1,k-1}(\mathcal{D}_{\boldsymbol{A}}^k) \subseteq \mathcal{M}_{\boldsymbol{A}}^k$. D'après le lemme 7.37, $\lambda_k \circ \sigma_k^{-1} = \pi_{\widehat{\boldsymbol{A}},k}^{-1} \circ \lambda_{1,k-1}$ et d'après le lemme 7.36 : $\lambda_{1,k}(\mathcal{D}_{A_{k+1}^{\boldsymbol{\beta}}}) = \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}}$. Nous obtenons ainsi

$$\begin{aligned}
\lambda_{1,k}(\mathcal{D}_{\boldsymbol{A}}^{k+1}) = \lambda_{1,k}(\mathcal{D}_{A_{k+1}^{\boldsymbol{\beta}}} \cap \sigma_k^{-1}(\mathcal{D}_{\boldsymbol{A}}^k)) &\subseteq \lambda_{1,k}(\mathcal{D}_{A_{k+1}^{\boldsymbol{\beta}}}) \cap \lambda_{1,k}(\sigma_k^{-1}(\mathcal{D}_{\boldsymbol{A}}^k)) \\
&= \lambda_{1,k}(\mathcal{D}_{A_{k+1}^{\boldsymbol{\beta}}}) \cap \pi_{\widehat{\boldsymbol{A}},k}^{-1}(\lambda_{1,k-1}(\mathcal{D}_{\boldsymbol{A}}^k)) \qquad (P_1) \\
&= \widetilde{\mathcal{D}}_{A_{k+1},A_{1,k}} \cap \pi_{\widehat{\boldsymbol{A}},k}^{-1}(\mathcal{M}_{\boldsymbol{A}}^k) = \mathcal{M}_{\boldsymbol{A}}^{k+1} \qquad (P_2)
\end{aligned}$$

((P_1) = d'après le lemme 7.37 ; (P_2) = d'après le lemme 7.36, par hypothèse d'induction et d'après la proposition 7.18.)

□

Afin de montrer le sens inverse, nous avons besoin de montrer que pour tout mot u de $\mathcal{M}^k_{\boldsymbol{A}}$, il existe un mot marqué v dans $\mathcal{D}^k_{\boldsymbol{A}}$ correspondant. Considérons l'ensemble de marquages pour un mot défini comme suit.

Définition 7.39 (Ensemble de marquages)**.** *Pour tout $k \geq 1$, soit la fonction MARKS_k qui à tout mot $u \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$ associe le langage $\text{MARKS}_k(u)$ défini comme suit :*

— *si $u = \varepsilon$, $\text{MARKS}_k(u) = \{\varepsilon\}$;*

— *sinon, si u admet une factorisation $u = xy$ avec $x, y \in \widetilde{\mathcal{D}}_{A_k,A_{k-1}} - \{\varepsilon\}$,*

$$\text{MARKS}_k(u) = \text{MARKS}_k(x)\text{MARKS}_k(y);$$

— *sinon, si u admet une factorisation $u = ax\bar{a}$ avec $a \in A_k$, $x \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$,*

$$\text{MARKS}_k(u) = a\text{MARKS}_k(x)\bar{a};$$

— *sinon, si u admet une factorisation $u = ax\bar{a}$ avec $a \in A_l$ avec $l < k$, $x \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$,*

$$\text{MARKS}_k(u) = \{a'v\bar{a'} \mid v \in \text{MARKS}_k(x), a' = (\beta_{i,1} \cdots \beta_{i,n}, a), n \geq 0\};$$

— *sinon, si u admet une factorisation $u = \bar{a}xa$ avec $a \in A_l$ avec $l < k$, $x \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$,*

$$\text{MARKS}_k(u) = \{a'v\bar{a'} \mid v \in \text{MARKS}_k(x), a' = (\beta_{i,1} \cdots \beta_{i,n}, a), n \geq 0\}.$$

Puisque la fonction MARKS_k substitue tout facteur bilatéral par un facteur unilatéral, nous pouvons admettre le lemme suivant.

Lemme 7.40. *Pour tout $k \geq 1$ et $u \in \widetilde{\mathcal{D}}_{A_k,A_{1,k-1}}$: $\text{MARKS}_k(u) \subseteq \mathcal{D}_{A_k^\beta}$.*

Considérons aussi le cas particulier suivant.

Définition 7.41. *Pour toute paire d'alphabets A, B et marqueur γ, soit la fonction $\text{MARK}_{A,B,\beta}$ qui à tout mot $u \in \widetilde{\mathcal{D}}_{A,B}$ associe le mot $\text{MARK}_{A,B,\gamma}(u)$ défini comme suit :*

— *si $u = \varepsilon$, $\text{MARK}_{A,B,\gamma}(u) = \varepsilon$;*

— *sinon, si u admet une factorisation $u = xy$ avec $x, y \in \widetilde{\mathcal{D}}_{A,B} - \{\varepsilon\}$,*

$$\text{MARK}_{A,B,\gamma}(u) = \text{MARK}_{A,B,\gamma}(x)\text{MARK}_{A,B,\gamma}(y);$$

— *sinon, si u admet une factorisation $u = ax\bar{a}$ avec $a \in A \cup B$, $x \in \widetilde{\mathcal{D}}_{A,B}$,*

$$\text{MARK}_{A,B,\gamma}(u) = a\text{MARK}_{A,B,\gamma}(x)\bar{a};$$

— *sinon, si u admet une factorisation $u = \bar{b}xb$ avec $b \in B$, $x \in \widetilde{\mathcal{D}}_{A,B}$,*

$$\text{MARK}_{A,B,\gamma}(u) = (\gamma, b)\text{MARK}_{A,B,\gamma}(x)\overline{(\gamma, b)};$$

Du fait de la définition de la fonction $\text{MARK}_{A,B,\gamma}$, nous pouvons aussi admettre le lemme suivant.

Lemme 7.42. *Soient A, B deux alphabets, β un marqueur et $u \in \widetilde{\mathcal{D}}_{A,B}$. Alors*

$$\lambda_{A,B,\gamma}(\text{MARK}_{A,B,\gamma})(u) = u; \qquad \sigma_{A,B,\gamma}(\text{MARK}_{a,B,\gamma})(u) = \pi_{\widehat{B}}(u).$$

Lemme 7.43. *Pour tout $k \geq 1$, $\mathcal{M}^k_{\boldsymbol{A}} \subseteq \lambda_{1,k-1}(\mathcal{D}^k_{\boldsymbol{A}})$.*

Démonstration. Soit $k \geq 1$ et $u \in \mathcal{M}^k_{\boldsymbol{A}}$. D'après le lemme 7.13, pour tout $i \in \{1,\dots,k\}$: $\pi_{\widehat{\boldsymbol{A}},i}(u) \in \mathcal{M}^i_{\boldsymbol{A}}$. Pour tout $i \in \{1,\dots,k\}$, soit alors

$$u_i = \pi_{\widehat{\boldsymbol{A}},i}(u).$$

D'après la proposition 7.18, chaque u_i appartient à $\mathcal{M}^i_{\boldsymbol{A}}$. Nous définissons pour tout $i \in \{1,\dots,k\}$, un mot v_i tel que

$$v_i \in \mathcal{D}^i_{\boldsymbol{A}} \quad \text{et } \lambda_{1,i-1}(v_i) = u_i.$$

comme suit :

1. Définir $v_1 = u_1$.
2. Pour tout i de 2 à k :
   — si $|u_i|_{A_i} = 0$, alors nous avons $u_i = u_{i-1}$; et dans ce cas, définir $v_i = v_{i-1}$.
   — Sinon, considérer l'unique décomposition $u_i = x_1\alpha_1x_2\alpha_2\cdots x_n\alpha_nx_{n+1}$ telle que chaque $\alpha_j \in \widehat{A_i}$ et $x_1x_2\cdots x_{n+1} = u_{i-1}$.
   Soient $y_1, y_2, \dots, y_{n+1}$ tels que $v_{i-1} = y_1y_2\cdots y_{n+1}$ et $x_j = \lambda_{1,i-1}(x_j)$ pour tout $j \in \{1,\dots,n+1\}$. Soit $w_i = y_1\alpha_1y_2\alpha_2\cdots y_n\alpha_ny_{n+1}$. Définir $v_i = \textsc{mark}_{A_i,A^{\boldsymbol{\beta}}_{i-1},\beta_k}(w_i)$.

Validité de la construction. Montrons que chaque v_i appartient à $\mathcal{D}^i_{\boldsymbol{A}}$ et que $\lambda_{1,i-1}(v_i) = u_i$.

Affirmation 1. Pour tout $1 < i \leq k$, $\sigma_{1,i-1}(v_i) = u_i$ *et* $\sigma_{i-1}(v_i) = v_{i-1}$.

Preuve de l'affirmation 1. Par induction sur i.

Cas initial. Par construction, nous avons $v_1 = u_1$. D'après le lemme 7.42, l'affirmation est aussi vraie pour $i = 2$ puisque par construction, $v_2 = \textsc{mark}_{A_2,A_1,\beta_1}(u_2)$.

Pas d'induction. Supposons l'affirmation vraie pour un certain i. Puisque $\lambda_{1,i} = \lambda_{1,i-1} \circ \lambda_i$ et que le mot $v_{i+1} = \textsc{mark}_{A_{i+1},A^{\boldsymbol{\beta}}_i,\beta_i}(w_{i+1})$ avec $w_{i+1} = y_1\alpha_iy_2\alpha_2\cdots y_n\alpha_ny_{n+1}$ avec chaque $\alpha_j \in \widehat{A_{i+1}}$ et $u_{i+1} = \lambda_{1,i-1}(y_1)\alpha_i\lambda_{1,-1}(y_2)\cdots\lambda_{1,i-1}\alpha_n\lambda_{1,i-1}(y_{n+1})$, d'après le lemme 7.42, on obtient

$$v_{i+1} = \lambda_{1,i-1} \circ \lambda_i \circ \textsc{mark}_{A_{i+1},A^{\boldsymbol{\beta}}_i,\beta_i}(w_{i+1}) = \lambda_{1,i-1}(w_{i+1}) = u_{i+1}.$$

De plus, d'après le lemme 7.42,

$$\sigma_i(v_{i+1}) = \sigma_i \circ \textsc{mark}_{A_{i+1},A^{\boldsymbol{\beta}}_i,\beta_i}(w_{i+1}) = \pi_{\widehat{A^{\boldsymbol{\beta}}_i}}(w_i) = v_i.$$

Ceci conclut la preuve de l'affirmation ci-dessus. □

Il est aussi clair par construction que chaque $v_i \in \textsc{marks}_i(u_i)$. D'après le corollaire 7.19, chaque $u_i \in \widetilde{\mathcal{D}}_{A_i,A_{1,i-1}}$ et par conséquent, d'après le lemme 7.40, chaque v_i appartient à $\mathcal{D}_{A^{\boldsymbol{\beta}}_i}$. Combiné à l'affirmation 1, ceci implique que chaque v_i appartient à $\mathcal{D}^i_{\boldsymbol{A}}$. □

Proposition 7.44. *Pour tout* $k \geq 1$, $\mathcal{M}^k_{\boldsymbol{A}} = \lambda_{1,k-1}(\mathcal{D}^k_{\boldsymbol{A}})$.

Démonstration. Les lemmes 7.38 et 7.43 montrent les deux sens de l'égalité. □

D'après le théorème 6.2, $\textsc{Lang}^k = \mathscr{C}(\mathcal{M}^k)$ où $\mathcal{M}^k$ est le langage $\mathcal{M}^k_{\boldsymbol{A}}$ où chaque alphabet A_i de la séquence $\boldsymbol{A}$ est de taille 2. Soit alors $\mathcal{D}^k$ le langage $\mathcal{D}^k_{\boldsymbol{A}}$ où chaque alphabet A_i de $\boldsymbol{A}$ est de taille 2. En conjonction avec le corollaire 7.24, la proposition 7.44 implique alors

Corollaire 7.45. *Pour tout* $k \geq 1$ *:*

$$\mathscr{C}(\mathcal{D}^k) = \mathscr{C}(\varepsilon\text{-}\textsc{Dyck}^k) = \mathscr{C}(\mathcal{M}^k) = \textsc{Lang}^k.$$

7.7 Théorèmes de représentation

Nous résumons dans cette section plusieurs théorèmes de représentation des familles LANG^k par transductions rationnelles en y ajoutant quelques uns suivant la même syntaxe que le théorème de Chomsky–Schützenberger.

Proposition 7.46. *Soit $k \geq 1$. Un langage $L \subseteq \Sigma^*$ appartient à LANG^k si et seulement si il existe une séquence d'alphabets $\boldsymbol{A}$, un langage régulier R et un homomorphisme f tels que $L = f(R \cap \mathcal{M}^k_{\boldsymbol{A}})$.*

Démonstration. Le sens ($\Leftarrow$) découle du théorème 6.2. Pour le sens ($\Rightarrow$), considérons l'automate à k-pile $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \Delta, F)$ tel que $\mathscr{L}(\mathcal{A}) = L$. Soit le langage régulier R_0 sur l'alphabet $\widehat{Q} \cup \widehat{\Sigma} \cup \widehat{A}_{1,k}$ tel que

$$R_0 = \{\bar{p}a\bar{a}\alpha q \mid (p, a, \alpha, q) \in \Delta\}.$$

Soit aussi $A'_k = A_k \cup \Sigma \cup F$ et $\boldsymbol{A}'$ la suite $A_1, A_2, \ldots, A_{k-1}, A'_k$. Nous pouvons naturellement émettre l'affirmation suivante.

Affirmation 1. Un mot $u \in \Sigma^*$ appartient à L si et seulement si il existe un mot $w \in q_0 R_0^* \overline{F}$ de la forme $w = q_0\, \overline{p_1} a_1 \overline{a_1} \alpha_1 q_1\, \overline{p_2} a_2 \overline{a_2} \alpha_2 q_2 \cdots \overline{p_n} a_n \overline{a_n} \alpha_n q_n\, \overline{q_f}$ tel que $u = a_1 a_2 \cdots a_n$ et

$$\forall i \in \{1, \ldots, n\} : p_i = q_{i-1}; \quad q_f = q_n; \quad \alpha_1 \alpha_2 \cdots \alpha_n \in \mathcal{M}^k_{\boldsymbol{A}}. \tag{P}$$

L'affirmation ci-dessus traduit le fait qu'un mot appartient à L si et seulement si il admet un calcul acceptant dans $\mathcal{A}$. Nous souhaitons montrer qu'un mot $w \in q_0 R_0^* \overline{F}$ satisfait (P) si et seulement si il appartient à $\mathcal{M}^k_{\boldsymbol{A}'}$.

D'après la proposition 7.18, un mot appartient à $\mathcal{M}^k_{\boldsymbol{A}'}$ si et seulement si il appartient à $\widetilde{\mathcal{D}}_{A'_k, A_{1,k-1}}$ et sa projection sur le niveau $k-1$ appartient à $\mathcal{M}^{k-1}_{\boldsymbol{A}}$. Ceci nous permet de prouver l'affirmation suivante.

Affirmation 2. Soit $w = q_0\, \overline{p_1} a_1 \overline{a_1} \alpha_1 q_1\, \overline{p_2} a_2 \overline{a_2} \alpha_2 q_2 \cdots \overline{q_n} a_n \overline{a_n} \alpha_n q_n\, \overline{q_f} \in q_0 R_0^* \overline{F}$. Le mot w satisfait la propriété (P) si et seulement si $w \in \mathcal{M}^k_{\boldsymbol{A}'}$.

Preuve de l'affirmation 2. Si $w \in \mathcal{M}^k_{\boldsymbol{A}'}$, d'après la proposition 7.18, $w \in \widetilde{\mathcal{D}}_{A'_k, A_{1,k-1}}$ et par conséquent le mot w est nécessairement de la forme

$$w = q_0\, \overline{q_0} a_1 \overline{a_1} \alpha_1 q_1\, \overline{q_1} a_2 \overline{a_2} \alpha_2 q_2 \cdots \overline{q_{n-1}} a_n \overline{a_n} \alpha_n q_n\, \overline{q_n}.$$

Ipso facto, $\widetilde{\rho}_{A'_k, A_{1,k-1}}(w) = \widetilde{\rho}_{A_k, A_{1,k-1}}(\alpha_1 \alpha_2 \cdots \alpha_n)$. Par conséquent nous avons $\alpha_1 \alpha_2 \cdots \alpha_n \in \widetilde{\mathcal{D}}_{A_k, A_{1,k-1}}$. D'après le lemme 7.13, nous avons aussi $\pi_{\widehat{A},k-1}(\alpha_1 \alpha_2 \cdots \alpha_n) = \pi_{\widehat{A'},k-1}(w) \in \mathcal{M}^{k-1}_{\boldsymbol{A}}$. C'est-à-dire $\alpha_1 \alpha_2 \cdots \alpha_n \in \mathcal{M}^k_{\boldsymbol{A}}$ (Proposition 7.18).

Réciproquement, si w satisfait (P), alors il est de la forme

$$w = q_0\, \overline{q_0} a_1 \bar{a_1} \alpha_1 q_1\, \overline{q_1} a_2 \overline{a_2} \alpha_2 q_2 \cdots \overline{q_{n-1}} a_n \overline{a_n} \alpha_n q_n\, \overline{q_n}$$

avec $\alpha_1 \alpha_2 \cdots \alpha_n \in \mathcal{M}^k_{\boldsymbol{A}}$ puisque le calcul décrit par w est acceptant. Ce qui induit de façon similaire que $w \in \mathcal{M}^k_{\boldsymbol{A}'}$. □

En combinant les deux affirmations précédentes, nous obtenons qu'un mot u appartient à L si et seulement si il existe un mot $w \in q_0 R_0^* \overline{F}$ tel que $w \in \mathcal{M}^k_{\boldsymbol{A}'}$ et $\pi_\Sigma(w) = u$. C'est-à-dire, $L = \pi_\Sigma(q_0 R_0^* \overline{F} \cap \mathcal{M}^k_{\boldsymbol{A}'})$. □

Corollaire 7.47. *Soit $k \geq 1$ et L un langage. L appartient à LANG^k si et seulement si il existe un langage régulier R, un homomorphisme f et une séquence d'alphabets $\boldsymbol{A}$ tels que $L = f(R \cap \mathcal{D}^k_{\boldsymbol{A}})$.*

Démonstration. Le sens ($\Leftarrow$) découle du fait que $\mathcal{D}^k_{\boldsymbol{A}}$ appartient à LANG^k (Corollaire 7.23) et que LANG^k est une famille agréable (théorème 6.1).

($\Rightarrow$) D'après la proposition 7.46, tout langage $L \in \text{LANG}^k$ peut s'exprimer comme $L = h(R \cap \mathcal{M}^k_{\boldsymbol{A}})$. D'après la proposition 7.44, $\mathcal{M}^k_{\boldsymbol{A}} = \lambda_{1,k-1}(\mathcal{D}^k_{\boldsymbol{A}})$. Nous obtenons ainsi

$$L = h(R \cap \lambda_{1,k-1}(\mathcal{D}^k_{\boldsymbol{A}})) = h \circ \lambda_{1,k-1}(\lambda^{-1}_{1,k-1}(R) \cap \mathcal{D}^{-1}_{\boldsymbol{A}}).$$

La famille des réguliers étant une famille agréable, le langage $\lambda^{-1}_{1,k-1}(R)$ est régulier. □

En conclusion, les familles LANG^k admettent les théorèmes de représentation par transductions rationnelles suivants :

Théorème 7.1. *Soit $k \geq 1$ et L un langage. Les affirmations suivantes sont équivalentes.*

1. *Le langage L appartient à LANG^k.*
2. *Il existe une séquence d'alphabets disjoints $\boldsymbol{A}$ de taille k (dans laquelle chaque alphabet A_i de A est de taille 2) et une transduction rationnelle τ telle que $L = \tau(\mathcal{M}^k_{\boldsymbol{A}})$.*
3. *Il existe une séquence d'alphabets disjoints $\boldsymbol{A}$ de taille k (dans laquelle chaque alphabet A_i de A est de taille 2) et une transduction rationnelle τ telle que $L = \tau(\mathcal{D}^k_{\boldsymbol{A}})$.*
4. *Il existe un langage $X \in \varepsilon\text{-DYCK}^k$ et une transduction rationnelle τ telle que $L = \tau(X)$.*
5. *Il existe une séquence d'alphabets disjoints $\boldsymbol{A}$ de taille k, un langage régulier R et un homomorphisme f tels que $L = f(R \cap \mathcal{M}^k_{\boldsymbol{A}})$.*
6. *Il existe une séquence d'alphabets disjoints $\boldsymbol{A}$ de taille k, un langage régulier R et un homomorphisme f tels que $L = f(R \cap \mathcal{D}^k_{\boldsymbol{A}})$.*
7. *Il existe un langage $X \in \varepsilon\text{-DYCK}^k$, un langage régulier R et un homomorphisme f tels que $L = f(R \cap X)$.*

Démonstration. Corollaires 7.45 et 7.47 et proposition 7.46. □

7.8 Machines à séquences d'homomorphismes

Les théorèmes de représentation ont souvent aidé à concevoir des formalismes de reconnaissances pour des familles de langages. En particulier, dans [HN81], Hirose et Nasu montrent qu'une famille de langages quelconque admet un certain type de théorèmes de représentation si et seulement si celle-ci admet une grammaire universelle[7]. Dans [Sor14], Sorokin prouve un théorème de représentation pour les frontières des forêts algébriques simples et s'en sert pour concevoir une classe de machine permettant de reconnaître ces langages. Il s'agit de machines munies de deux piles, les mouvements de celles-ci sont synchronisés et restreints à une classe de « synchronisation ».

Nous menons une étude similaire pour les cônes rationnels générés par des itérations du langage de Dyck (de façon générale et pas uniquement pour les langages de Dyck « d'ordres supérieurs »). Nous montrons ici que ces familles de langages sont reconnaissables par des automates à plusieurs piles pour lesquelles le comportement de ces piles peut être spécifié par une séquence d'homomorphismes.

7. Une grammaire algébrique $\mathcal{G}$ est dite universelle gauche pour une famille de langage $\mathcal{F}$, vis-à-vis d'une famille $\mathcal{F}_2$ si ses dérivations terminales gauches pour lesquelles les suites de productions effectuées appartiennent à $\mathcal{F}_2$ est précisément des langages de $\mathcal{F}$

Définition 7.48 (Classes de langages de Dyck itérés)**.** *Soit $\mathcal{H}$ une classe (quelconque) d'homomorphismes. Pour tout $k \geq 1$, soit $\mathcal{H}$-DYCK^k l'ensemble de langages de la forme $D_{k,\boldsymbol{g}}$ où $\boldsymbol{g}$ une séquence d'homomorphismes de $\mathcal{H}$.*

Définition 7.49 (Machines à séquences (syntaxe))**.** *Soit $\mathcal{H}$ une classe d'homomorphismes et k un entier positif. Une* machine à séquences *sur $\mathcal{H}$ et de niveau k est une structure $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ où*

- *Σ est l'alphabet de travail ;*
- *Q est l'ensemble fini d'états, $q_0 \in Q$ est l'état initial et $F \subseteq Q$ est l'ensemble d'états acceptants ;*
- *$\boldsymbol{A}$ est une séquence d'alphabets de pile, de taille k ;*
- *$\boldsymbol{g}$ est une séquence d'homomorphismes $g_1, g_2, \ldots, g_{k-1}$ sur $\boldsymbol{A}$ telle que chaque g_i appartient à $\mathcal{H}$;*
- *$\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (A_k \cup \{\varepsilon\}) \times Q$ est la relation de transition.*

Définition 7.50 (Machines à séquences (sémantique))**.** *Soit $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ une* machine à séquences *sur $\mathcal{H}$ et de niveau k.*

Pour tout $i \leq k$ et $a \in A_i$, $[\![a]\!]$ désigne ici l'action COPIE$_a$ *sur la i-ème pile et $[\![\bar{a}]\!] = \overline{[\![a]\!]} = \overline{\text{COPIE}_a}$. Pour tout mot $u = a_1 a_2 \cdots a_n \in \widehat{A_l}^*$: $[\![u]\!] = [\![a_n]\!] \circ \cdots \circ [\![a_2]\!] \circ [\![a_1]\!]$.*

Configurations. Une configuration de $\mathcal{A}$ est un tuplet $c = (q, u, \omega_k, \omega_{k-1}, \ldots, \omega_1)$ où $q \in Q$ est l'état courant, $u \in \Sigma^*$ est le mot restant à lire sur la bande de lecture, et chaque $\omega_i \in {A_i}^* \perp$, pour $i \in \{1, \ldots, k\}$ est le contenu de la i-ème pile. La relation $\underset{\mathcal{A}}{\vdash}$ décrit une étape de calcul. Soient deux configurations $c = (q, u, \omega_k, \omega_{k-1}, \cdots, \omega_1)$ et $c' = (q', u', \omega'_k, \omega'_{k-1}, \cdots, \omega'_1)$. Nous avons $c \underset{\mathcal{A}}{\vdash} c'$ si et seulement si il existe une transition $(q, a, \alpha, q') \in \Delta$ telle que $u = au'$, $\omega'_k = [\![\alpha]\!](\omega_k)$ et $\omega'_l = [\![g_{l,k-1}(\alpha)]\!](\omega_l)$ pour tout $l \in \{1, \ldots, k-1\}$.

La relation $\underset{\mathcal{A}}{\vdash}^*$ désigne la clôture réflexive et transitive de $\underset{\mathcal{A}}{\vdash}$.

Langages reconnus. Le langage reconnu par $\mathcal{A}$ est

$$\mathscr{L}(\mathcal{A}) = \{u \in \Sigma^* \mid \exists q_f \in F : (q_0, u, \underbrace{\perp, \cdots, \perp}_{k \text{ fois}}) \underset{\mathcal{A}}{\vdash}^* (q_f, \varepsilon, \underbrace{\perp, \cdots, \perp}_{k \text{ fois}})\}.$$

Notations. Nous décrivons un calcul par une suite de transitions effectuées e.g.

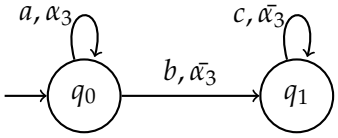
$$q_0 \xrightarrow{a_1, \alpha_1} q_1 \xrightarrow{a_2, \alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n, \alpha_n} q_{n+1}$$

avec chaque $(q_i, a_{i+1}, \alpha_{i+1}, q_{i+1}) \in \Delta$.

Exemple 7.51. *Soit la machine à séquence de niveau 3 $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ avec $A_1 = \{\alpha_1\}$, $A_2 = \{\alpha_2\}$, $A_3 = \{\alpha_3\}$, la séquence $\boldsymbol{g}$ telle que*

$$g_2(\alpha_3) = \alpha_2\alpha_2, \quad g_2(\bar{\alpha_3}) = \bar{\alpha_2}, \quad g_1(\alpha_2) = \alpha_1, \quad g_1(\bar{\alpha_1}) = \bar{\alpha_1}\bar{\alpha_1}.$$

Nous avons alors $g_1 \circ g_2(\alpha_3) = \alpha_1\alpha_1$ et $g_1 \circ g_2(\bar{\alpha_3}) = \bar{\alpha_1}\bar{\alpha_1}$. Les ensembles Q, Σ, Δ et F sont décrits par le graphe de transitions suivant.



Le calcul sur le mot aabc par la suite de transitions $q_0 \xrightarrow{a,\alpha_3} q_0 \xrightarrow{a,\alpha_3} q_0 \xrightarrow{b,\bar{\alpha_3}} q_1 \xrightarrow{c,\bar{\alpha_3}} q_1$ *est comme suit*

$$\begin{aligned}(q_0, aabc, \bot, \bot, \bot) &\underset{\mathcal{A}}{\vdash} (q_0, abc, \alpha_3\bot, \alpha_2\alpha_2\bot, \alpha_1\alpha_1\bot)\\ &\underset{\mathcal{A}}{\vdash} (q_0, bc, \alpha_3\alpha_3\bot, \alpha_2\alpha_2\alpha_2\alpha_2\bot, \alpha_1\alpha_1\alpha_1\alpha_1\bot)\\ &\underset{\mathcal{A}}{\vdash} (q_1, c, \alpha_3\bot, \alpha_2\alpha_2\alpha_2\bot, \alpha_1\alpha_1\bot)\\ &\underset{\mathcal{A}}{\vdash} (q_1, \varepsilon, \bot, \alpha_2\alpha_2\bot, \bot).\end{aligned}$$

Considérons dans toute la suite un entier k fixé, et une classe d'homomorphismes $\mathcal{H}$.

Définition 7.52 (Langages reconnaissables par des machines à séquence)**.** *Soit* $\text{REC}^k_{\mathcal{H}}$ *la famille de langages reconnaissables par machines des machines à séquences sur* $\mathcal{H}$ *et de niveau* k*.*

Lemme 7.53 (Langage de comportement)**.** *Soit* $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ *une machine à séquence sur* $\mathcal{H}$ *de niveau* k*. Un calcul*

$$q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_n$$

avec $q_n \in F$ *est acceptant si et seulement si* $\alpha_1\alpha_2\cdots\alpha_n \in D_{k,\boldsymbol{g}}$*.*

Démonstration. En effet, le calcul susmentionné est acceptant ssi $[\![\alpha_1\alpha_2\cdots\alpha_n]\!](\bot) = \bot$ et pour tout $l \in \{1, \ldots, k-1\}$: $[\![g_{l,k-1}(\alpha_1\alpha_2\cdots\alpha_n)]\!](\bot) = \bot$; c'est-à-dire ssi $\alpha_1\alpha_2\cdots\alpha_n \in \mathcal{D}_{A_k}$ et pour tout $l \in \{1, \ldots, k-1\}$: $g_{l,k-1}(\alpha_1\alpha_2\cdots\alpha_n) \in \mathcal{D}_{A_l}$; c'est-à-dire ssi $\alpha_1\alpha_2\cdots\alpha_n \in D_{k,\boldsymbol{g}}$. □

Lemme 7.54. *Soit* $\boldsymbol{g}$ *une séquence d'homomorphismes de taille* $k-1$ *telle que chaque* $g_i \in \mathcal{H}$*. Le langage* $D_{k,\boldsymbol{g}}$ *appartient à* $\text{REC}^k_{\mathcal{H}}$*.*

Démonstration. En effet, conformément au lemme 7.53, le langage $D_{k,\boldsymbol{g}}$ est reconnu par la machine $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ telle que

$$\Sigma = A_k, \quad Q = F = \{q_0\}, \quad \text{et } \Delta = \{(q_0, a, a, q_0) \mid a \in \widehat{A_k}\}.$$

□

Lemme 7.55. *La famille* $\text{REC}^k_{\mathcal{H}}$ *est effectivement fermée par transductions rationnelles.*

Démonstration. Conformément au théorème de Nivat, il nous suffit de prouver que $\text{REC}^k_{\mathcal{H}}$ est fermé par intersection avec des réguliers, homomorphismes alphabétiques et images inverses d'homomorphismes alphabétiques (le calcul des homomorphismes et du langage régulier sont de plus effectifs depuis un transducteur fini). La fermeture par ces opérations est une propriété standard des familles de langages reconnaissables par machines abstraites (voir [Gin75]). Celle-ci est indépendante du type de structure de mémoire utilisée.

Fermeture par homomorphismes alphabétiques. Soit $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ une machine à séquence sur $\mathcal{H}$ de niveau k et $f : \Sigma^* \longrightarrow {\Sigma_2}^*$ un homomorphisme alphabétique. La machine $\mathcal{B} = (Q, \Sigma_2, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta', F)$ telle que $\mathscr{L}(\mathcal{B}) = f(\mathscr{L}(\mathcal{A}))$ est construite en remplaçant chaque production $(q, a, \alpha, p) \in \Delta$ par $(q, f(a), \alpha, p)$.

Fermeture par intersection avec des réguliers. Soit $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ une machine à séquence sur $\mathcal{H}$ de niveau k et $R \subseteq \Sigma^*$ un langage régulier. Il existe donc un automate fini $\mathcal{A}_R = (Q_R, \Sigma, q_{0,R}, \Delta_R, F_R)$ tel que $\mathscr{L}(\mathcal{A}_R) = R$. Nous supposons sans perte de généralité que pour tout $q \in Q_R$, il existe une ε-transition (q, ε, q) (ceci nous permet de synchroniser les deux machines). La machine $\mathcal{B} = (Q', \Sigma, q_0', \boldsymbol{A}, \boldsymbol{g}, \Delta', F)$ telle que $\mathscr{L}(\mathcal{B}) = \mathscr{L}(\mathcal{A}) \cap R$ est construite par un produit entre les $\mathcal{A}$ et $\mathcal{A}_R$ comme suit :

$$Q' = Q \times Q_R; \quad q_0' = (q_0, q_{0,R});$$

et pour tout $q, q' \in Q, a \in \Sigma \cup \{\varepsilon\}, \alpha \in A_k \cup \{\varepsilon\}, p, p' \in Q_R$:

$$((q,p), a, \alpha, (q', p')) \in \Delta' \quad \Leftrightarrow \quad (q, a, \alpha, q') \in \Delta \text{ et } (q', a, p') \in \Delta_R.$$

Il existe alors dans $\mathcal{B}$ un calcul acceptant

Fermeture par images inverses d'homomorphismes alphabétiques. Soit une machine à séquence sur $\mathcal{H}$ de niveau k, $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ et $f : {\Sigma_2}^* \longrightarrow \Sigma^*$ un homomorphisme alphabétique. La machine $\mathcal{B} = (Q, \Sigma_2, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta', F)$ telle que $\mathscr{L}(\mathcal{B}) = f^{-1}(\mathscr{L}(\mathcal{A}))$ est construite en substituant chaque production (q, a, α, q) par l'ensemble de production $\{(q, b, \alpha, p) \mid f(b) = a\}$, et en ajoutant pour tout $q \in Q$ et $b \in f^{-1}(\varepsilon)$ la transition (q, b, ε, p). On obtient ainsi que pour tout calcul $q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_n$ de $\mathcal{B}$ est acceptant si et seulement si il existe un calcul acceptant pour $f(a_1 a_2 \cdots a_n)$ dans $\mathcal{A}$. □

Proposition 7.56. $\mathscr{C}(\mathcal{H}\text{-DYCK}^k) \subseteq \text{REC}^k_{\mathcal{H}}$.

Démonstration. D'après les lemmes 7.54 et 7.55, $\mathcal{H}\text{-DYCK}^k \subseteq \text{REC}^k_{\mathcal{H}}$ et $\text{REC}^k_{\mathcal{H}}$ est fermé par transductions rationnelles. Il en découle la proposition ci-dessus. □

Proposition 7.57. $\text{REC}^k_{\mathcal{H}} \subseteq \mathscr{C}(\mathcal{H}\text{-DYCK}^k)$.

Démonstration. Similairement à la preuve du théorème 6.2 ; Soit $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ une machine à séquence sur $\mathcal{H}$ de niveau k, il existe une transduction τ telle que $\mathscr{L}(\mathcal{A}) = \tau(D_{k,\boldsymbol{g}})$. Celle-ci est décrite par le transducteur $\mathcal{T} = (Q, \widehat{A}_{1,k}, \Sigma, q_0, \Delta', F)$ tel que pour tout $q, p \in Q, a \in \Sigma \cup \{\varepsilon\}, \alpha \in \widehat{A}_{1,k} \cup \{\varepsilon\}$:

$$(q, \alpha | a, p) \in \Delta' \quad \Leftrightarrow \quad (q, a, \alpha, p) \in \Delta.$$

Il existe alors dans $\mathcal{T}$ un calcul

$$q_0 \xrightarrow{\alpha_1|a_1} q_1 \xrightarrow{\alpha_2|a_2} q_2 \cdots q_{n-1} \xrightarrow{\alpha_n|a_n} q_n$$

si et seulement si il existe dans $\mathcal{A}$ un calcul

$$q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_n.$$

De plus, ce dernier calcul est acceptant ssi $q_n \in F$ et $\alpha_1 \alpha_2 \cdots \alpha_n \in D_{k,\boldsymbol{g}}$ (Lemme 7.54). Il en découle que $\mathscr{L}(\mathcal{A}) = \tau(D_{k,\boldsymbol{g}})$. □

Corollaire 7.58. $\text{REC}^k_{\mathcal{H}} = \mathscr{C}(\mathcal{H}\text{-DYCK}^k)$.

7.8.1 Applications aux théorèmes de représentation

D'après le corollaire 7.45, nous avons l'équivalence qui suit :

$$\textsc{Lang}^k = \mathscr{C}(\varepsilon\text{-}\textsc{Dyck}^k) = \mathscr{C}(\mathcal{D}^k).$$

En conjonction avec le corollaire 7.58, nous obtenons le théorème ci-dessous.

Théorème 7.2. *Soit $k \geq 1$ et L un langage. Les affirmations suivantes sont équivalentes.*

1. $L \in \textsc{Lang}^k$.
2. *L est reconnu par un automate à séquence de niveau k sur une séquence d'homomorphismes ε-sûrs (symétriques, symétriques et alphabétiques).*
3. *L est reconnu par un automate à séquence de niveau k sur la séquence de d-projections $\boldsymbol{\sigma}$.*

Sorokin [Sor14] et Kanazawa [Kan14b] ont prouvé que pour tout $m \geq 1$, les frontières d'arbres algébriques simples de rang $m-1$ sont des images par transduction rationnelle de langages de la forme $\mathcal{D}_{nm} \cap \mathrm{W}_{n,m}^{-1}(\mathcal{D}_{nm})$ où n est un entier positif et $\mathrm{W}_{n,m}$ est un homomorphisme particulier. Pour tout $m \geq 1$, soit alors $\boldsymbol{W}_m$ l'ensemble des homomorphismes $\mathrm{W}_{n,m}$ avec $n \geq 1$. En conjonction avec le corollaire 7.58, nous obtenons le théorème ci-dessous.

Théorème 7.3. *[Sor14] Soit $m \geq 1$ et L un langage. Les affirmations suivantes sont équivalentes*

1. *L est la frontière d'une forêt algébrique simple de rang $(m-1)$.*
2. $L \in \textsc{Rec}^2_{\boldsymbol{W}_m}$.

La classe d'automate introduite dans [Sor14], est équivalente à celle de $\textsc{Rec}^2_{\boldsymbol{W}_m}$.

7.8.2 Transductions à séquences (simulations des piles)

Nous introduisons ici les notions de transducteurs et transductions à séquences. Celle-ci sont des machines à séquences munies d'un ruban d'entrée et de sortie et telles que le mot lu sur le ruban d'entrée est restreint à l'image par une séquence d'homomorphismes d'un langage de Dyck itéré par cette même séquence d'homomorphismes. Cette restriction permet à ce que si le mot lu en entrée appartient à un langage de Dyck itéré, alors celui-ci permet de simuler le comportement de plusieurs piles et le mot de sortie appartient alors à un langage de niveau supérieur.

Définition 7.59 (Transducteur à séquences (syntaxe))**.** *Soit $\mathcal{H}$ une classe d'homomorphismes et k un entier positif. Un* transducteur à séquences *sur $\mathcal{H}$ et de niveau k est une structure $\mathcal{T} = (Q, \Sigma_1, \Sigma_2, q_0, \boldsymbol{A}, f, \boldsymbol{g}, \Delta, F)$ où*

- *$\widehat{\Sigma_1}$ est l'alphabet d'entrée ;*
- *Σ_2 est l'alphabet de sortie ;*
- *Q est l'ensemble fini d'états, $q_0 \in Q$ est l'état initial et $F \subseteq Q$ est l'ensemble d'états acceptants ;*
- *$\boldsymbol{A}$ est une séquence d'alphabets de pile, de taille k ;*
- *$\boldsymbol{g}$ est une séquence d'homomorphismes $g_1, g_2, \ldots, g_{k-1}$ sur $\boldsymbol{A}$ telle que chaque g_i appartient à $\mathcal{H}$;*
- *f est un homomorphisme de $\widehat{A_1}$ dans $\widehat{\Sigma_1}^*$ appartenant à $\mathcal{H}$;*
- *$\Delta \subseteq Q \times (\Sigma_1^* \mid \Sigma \cup \{\varepsilon\}) \times (A_k \cup \{\varepsilon\}) \times Q$ est la relation de transition ; de plus toute transition est de la forme $(q, (f \circ g_{1,k-1}(\alpha) \mid a), \alpha, p)$.*

Définition 7.60 (Transducteur à séquences (sémantique))**.** *Soit $\mathcal{T} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ un* transducteur à séquences *sur $\mathcal{H}$ et de niveau k.*

Configurations. *Une configuration de $\mathcal{T}$ est un uplet $c = (q, u, v, \omega_k, \omega_{k-1}, \dots, \omega_1)$ où $q \in Q$ est l'état courant, $u \in \Sigma_1^*$ est le mot restant à lire sur la bande de lecture, $v \in \Sigma_2^*$ est le mot écrit sur la bande d'écriture et chaque $\omega_i \in {A_i}^* \perp$, pour $i \in \{1, \dots, k\}$ est le contenu de la i-ème pile. La relation $\underset{\mathcal{T}}{\vdash}$ décrit une étape de calcul. Soient $c = (q, u, v, \omega_k, \omega_{k-1}, \cdots, \omega_1)$ et $c' = (q', u', v', \omega'_k, \omega'_{k-1}, \cdots, \omega'_1)$; $c \underset{\mathcal{T}}{\vdash} c'$ si et seulement si il existe une transition $(q, x|a, \alpha, q') \in \Delta$ telle que $v = av'$, $u' = ux$, $\omega'_k = [\![\alpha]\!](\omega_k)$ et $\omega'_l = [\![g_{l,k-1}(\alpha)]\!](\omega_l)$ pour tout $l \in \{1, \dots, k-1\}$.*

La relation $\underset{\mathcal{T}}{\vdash}^*$ désigne la clôture réflexive et transitive de $\underset{\mathcal{T}}{\vdash}$.

Relation décrite. *La relation décrite par $\mathcal{T}$ est*

$$\mathcal{R}(\mathcal{T}) = \{(u, v) \in \Sigma_1^* \times \Sigma_2^* \mid \exists q_f \in F : (q_0, u, \varepsilon, \underbrace{\perp, \cdots, \perp}_{k \text{ fois}}) \underset{\mathcal{T}}{\vdash}^* (q_f, \varepsilon, v, \underbrace{\perp, \cdots, \perp}_{k \text{ fois}})\}.$$

Notations. Nous décrivons un calcul par une suite de transitions effectuées e.g.

$$q_0 \xrightarrow{(x_1|a_1),\alpha_1} q_1 \xrightarrow{(x_2|a_2),\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{(x_n|a_n),\alpha_n} q_{n+1}$$

avec chaque $(q_i, (x_{i+1}|a_{i+1}), \alpha_{i+1}, q_{i+1}) \in \Delta$.

Remarque 7.61. *Par définition de la relation de transitions Δ, pour tout calcul*

$$q_0 \xrightarrow{x_1|a_1,\alpha_1} q_1 \xrightarrow{x_2|a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{x_n|a_n,\alpha_n} q_{n+1},$$

nous avons $x_1 x_2 \cdots x_n = f \circ g_{1,k-1}(\alpha_1 \alpha_2 \cdots \alpha_n)$.

Puisqu'un transducteur à séquence est aussi une machine à séquence, d'après le lemme 7.53, le calcul susmentionné est acceptant si et seulement si $q_{n+1} \in F$ et $\alpha_1 \alpha_2 \cdots \alpha_n \in D_{k,\boldsymbol{g}}$. Le domaine de la transduction décrite est alors inclus dans $f \circ g_{1,k-1}(D_{k,\boldsymbol{g}})$.

Définition 7.62 (Classes de transductions à séquences généralisées)**.** *Soit $\text{TRANS}^k_{\mathcal{H}}$ la classe de transductions descriptibles par des transducteurs à séquences de niveau k sur $\mathcal{H}$.*

Théorème 7.4. *Soient $k, l \geq 1$ et L un langage. Le langage L appartient à $\text{REC}^{k+l}_{\mathcal{H}}$ si et seulement si il existe une transduction $\tau \in \text{TRANS}^k_{\mathcal{H}}$ et un langage $X \in \mathcal{H}\text{-DYCK}^l$ tels que $L = \tau(X)$.*

Démonstration. ($\Rightarrow$) Soit $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ une machine à séquence de niveau $k + l$ sur $\mathcal{H}$ reconnaissant L avec

$$\boldsymbol{A} = A_1, A_2, \dots, A_{k+l+1} \qquad \text{et} \quad \boldsymbol{g} = g_1, g_2, \dots, g_{k+l-1}.$$

Soit $\boldsymbol{B} = A_{l+1}, A_{l+2}, \cdots, A_{k+l}$ la séquence des k derniers alphabets de $\boldsymbol{A}$. Soit $\boldsymbol{h} = g_1, g_2, \dots, g_{l-1}$ la séquence des $l - 1$ premiers homomorphismes de $\boldsymbol{g}$ et

$$\boldsymbol{f} = g_{l+1}, g_{l+2}, \dots, g_{k+l-1}$$

la séquence des $k - 1$ derniers homomorphismes de $\boldsymbol{g}$. Nous avons alors

$$h_{1,l-1} \circ g_l \circ f_{1,k-1} = g_{1,k+l-1}.$$

Nous construisons le transducteur $\mathcal{T}$ tel que $L = \mathcal{R}(\mathcal{T})(D_{l,\boldsymbol{h}})$ comme suit :

$\mathcal{T} = (Q, A_l, \Sigma, q_0, \boldsymbol{B}, g_l, \boldsymbol{f}, \Delta', F)$ où Δ' est tel que pour tout $q, p \in Q, \alpha \in A_{k+l} \cup \{\varepsilon\}, a \in \Sigma \cup \{\varepsilon\}$:

$$(q, (g_l \circ f_{1,k-1}(\alpha) \mid a), \alpha, p) \in \Delta' \quad \Leftrightarrow \quad (q, a, \alpha, p) \in \Delta.$$

Par construction, tout calcul acceptant de $\mathcal{T}$ est de la forme

$$C_1 = q_0 \xrightarrow{(x_1|a_1),\alpha_1} q_1 \xrightarrow{(x_2|a_2),\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{(x_n|a_n),\alpha_n} q_f$$

avec $q_f \in F$, chaque $x_i = g_l \circ f_{1,k-1}(\alpha_i)$ et $\alpha_1\alpha_2\cdots\alpha_n \in D_{k,\boldsymbol{f}}$ (puisque $\mathcal{T}$ est une machine à séquence) et

$$C_2 = q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_f$$

est un calcul dans $\mathcal{A}$. Si $x_1x_2\cdots x_n = g_l \circ g_{1,k-1}(\alpha_1\alpha_2\alpha_n) \in D_{l,\boldsymbol{h}}$, alors on obtient $\alpha_1\alpha_2\cdots\alpha_n \in D_{k,\boldsymbol{f}}$ et $g_l \circ g_{1,k-1}(\alpha_1\alpha_2\alpha_n) \in D_{l,\boldsymbol{h}}$ c'est à dire $\alpha_1\alpha_2\cdots\alpha_n \in D_{k+l,\boldsymbol{g}}$. Le calcul C_2 est donc acceptant dans $\mathcal{A}$. Nous pouvons en déduire que $\mathcal{R}(\mathcal{T})(D_{l,\boldsymbol{h}}) \subseteq L$.

Réciproquement, tout calcul acceptant dans $\mathcal{A}$ est de la forme

$$C_2 = q_0 \xrightarrow{a_1,\alpha_1} q_1 \xrightarrow{a_2,\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{a_n,\alpha_n} q_f$$

avec $q_f \in F$, $\alpha_1\alpha_2\cdots\alpha_n \in D_{k+l,\boldsymbol{g}}$ et tel que

$$C_1 = q_0 \xrightarrow{(x_1|a_1),\alpha_1} q_1 \xrightarrow{(x_2|a_2),\alpha_2} q_2 \cdots q_{n-1} \xrightarrow{(x_n|a_n),\alpha_n} q_f$$

avec chaque $x_i = g_l \circ f_{1,k-1}(\alpha_i)$. Puisque $\alpha_1\alpha_2\cdots\alpha_n \in D_{k+l,\boldsymbol{g}}$, nous avons

$$\alpha_1\alpha_2\cdots\alpha_n \in D_{k,\boldsymbol{f}} \tag{A}$$
$$g_l \circ f_{1,k-1}(\alpha_1\alpha_2\cdots\alpha_n) \in D_{l,h}. \tag{B}$$

(A) implique que le calcul C_1 est acceptant et (B) implique que $x_1x_2\cdots x_n \in D_{l,h}$. Il en découle que $L \subseteq \mathcal{R}(\mathcal{T})(D_{l,\boldsymbol{h}})$.

($\Leftarrow$) Il nous suffit ici de faire la construction inverse. Etant donné un transducteur de niveau k $\mathcal{T} = (Q, A_l, \Sigma, q_0, \boldsymbol{B}, g_l, \boldsymbol{f}, \Delta', F)$ et une séquence d'homomorphisme $\boldsymbol{h}$ de taille $l-1$, la machine de niveau $k+l$ $\mathcal{A}$ telle que $\mathscr{L}(\mathcal{A}) = \mathcal{R}(\mathcal{T})(D_{l,\boldsymbol{h}})$ est $\mathcal{A} = (Q, \Sigma, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ où $\boldsymbol{g}$ est l'extension de $\boldsymbol{h}$ par g_l et $\boldsymbol{f}$; les alphabets de $\boldsymbol{A}$ sont conformes à $\boldsymbol{h}$ et g_l et ses k derniers alphabets sont ceux de $\boldsymbol{B}$; et la relation Δ est telle que pour tout $q, p \in Q, \alpha \in A_{k+l} \cup \{\varepsilon\}, a \in \Sigma \cup \{\varepsilon\}$:

$$(q, (g_l \circ f_{1,k-1}(\alpha) \mid a), \alpha, p) \in \Delta' \quad \Leftrightarrow \quad (q, a, \alpha, p) \in \Delta.$$

Nous obtenons ainsi la même relation entre $\mathcal{A}$ et $\mathcal{T}$ que dans le sens ($\Rightarrow$). □

Définition 7.63 (Transductions ε-sûres d'ordres supérieurs)**.** *Pour tout $k \geq 1$, soit ε-TRANS^k la classe de transductions décrites par des transducteurs à séquences de niveau k sur des séquences d'homomorphismes ε-sûrs.*

Un corollaire du théorème 7.4 est alors le suivant :

Corollaire 7.64. *Soient $k, l \geq 1$. Un langage L appartient à* LANG^{k+l} *si et seulement si il existe une transduction $\tau \in \varepsilon$-TRANS^k et un langage $X \in \varepsilon$-DYCK^l tels que $L = \tau(X)$.*

Théorèmes de représentation pour les transductions à séquences

Définition 7.65. *Nous disons qu'une classe d'homomorphisme $\mathcal{H}$ est fermée par composition* droite *avec des projections si pour tout homomorphisme $g : \widehat{A_2}^* \longrightarrow \widehat{A_1}^*$, et pour tout alphabets B tel que $A_2 \subseteq B$, l'homomorphisme $g \circ \pi_{A_2}$ appartient à $\mathcal{H}$; où π_{A_2} est la projection de B dans A_2.*

Théorème 7.5. *Soit une classe d'homomorphisme $\mathcal{H}$ fermée par composition droite avec des projections, $k \geq 1$ et τ une transduction. Les affirmations suivante sont équivalentes.*

1. *La transduction τ appartient à $\text{TRANS}^k_{\mathcal{H}}$.*
2. *Il existe un langage régulier R, une séquence d'homomorphismes $\boldsymbol{g}$ de $\mathcal{H}$ et de taille $k-1$, un homomorphisme f de $\mathcal{H}$ et un homomorphisme h tels que*

$$\tau = \{(f \circ g_{1,k-1}(u), h(u)) \mid u \in R \cap D_{k,\boldsymbol{g}}\}.$$

Démonstration. $(2 \Rightarrow 1)$ Il suffit de considérer la machine à séquence reconnaissant le langage $R \cap D_{k,\boldsymbol{g}}$. Le langage $D_{k,\boldsymbol{g}}$ est reconnu par une machine à séquence de niveau k dans laquelle toute production est de la forme (q, α, α, q). D'après la construction de la fermeture par intersection, le langage $R \cap D_{k,\boldsymbol{g}}$ est reconnu par une machine dans laquelle chaque transition est de la forme (q, α, α, p). Il suffit alors de substituer chaque transition (q, α, α, p) par $(q, (f \circ g_{1,k-1}(\alpha) \mid h(\alpha)), \alpha, p)$ pour obtenir un transducteur de niveau k sur $\mathcal{H}$.

$(1 \Rightarrow 2)$ Soit $\mathcal{T} = (Q, \Sigma_1, \Sigma_2, q_0, \boldsymbol{A}, f, \boldsymbol{g}, \Delta, F)$ un transducteur de niveau k sur $\mathcal{H}$ décrivant τ. Supposons que la séquence $\boldsymbol{g}$ est $\boldsymbol{g} = g_1, g_2, \ldots, g_{k-1}$. Soit $\mathcal{A} = (Q, \Sigma_2 \cup \widehat{\Delta}, q_0, \boldsymbol{A}, \boldsymbol{g}, \Delta, F)$ la machine à séquence obtenue en transformant toute transition $\delta = (q, (f \circ g_{1,k-1}(\alpha) \mid a), \alpha, p) \in \Delta$ en $(q, \delta\bar{\delta}\alpha, \alpha, p)$.

Soit l'homomorphisme $\mu = g_k \circ \pi_{A_k}$ où π_{A_k} est la projection de $\widehat{\Delta \cup A_k}$ dans A_k. Soit la séquence $\boldsymbol{g}' = g_1, g_2, \ldots, g_{k-1}, \boldsymbol{\mu}$. Nous pouvons affirmer que pour tout mot u de la forme

$$u = \delta_1\bar{\delta_1}\alpha_1\delta_2\bar{\delta_2}\alpha_2 \cdots \delta_n\bar{\delta_n}\alpha_n$$

tel que chaque $\delta_i \in \Delta$ et chaque $\alpha_i \in \widehat{A_k} \cup \{\varepsilon\}$, nous avons $u \in \mathcal{D}_{A_k \cup \Delta}$ si et seulement si $\pi_{A_k}(u) = \alpha_1\alpha_2 \cdots \alpha_n \in \mathcal{D}_{A_k}$ et par conséquent, $u \in D_{k,\boldsymbol{g}'}$ si et seulement si $\pi_{A_k}(u) \in D_{k,\boldsymbol{g}}$.

Soit aussi l'homomorphisme $h : \widehat{A_k \cup \Delta}^* \longrightarrow \Sigma_2^*$ tel que

$$h(a) = \varepsilon \text{ pour tout } a \in \widehat{A}_k \cup \overline{\Delta}, \quad h(\delta) = a \text{ pour } \delta = (q, (x \mid a), \alpha, p) \in \Delta.$$

Il est alors clair que $\tau = \{(f \circ g'_{1,k-1}(u), h(u)) \mid u \in \mathscr{L}(\mathcal{A})\}$. Il nous reste à prouver que $\mathscr{L}(\mathcal{A})$ peut s'exprimer comme l'intersection d'un langage régulier avec $D_{k,\boldsymbol{g}'}$.

Affirmation 1. Il existe un langage régulier R tel que $\mathscr{L}(\mathcal{A}) = R \cap D_{k,\boldsymbol{g}'}$.

Preuve de l'affirmation 1. En effet, soit $\mathcal{A}_R = (Q, \widehat{A_k \cup \Delta}, q_0, \Delta_R, F)$ l'automate fini sous-jacent à $\mathcal{A}$. Par construction, tout calcul de $\mathcal{A}_R$ est de la forme

$$C_{\mathcal{A}_R} = q_0 \xrightarrow{\delta_1\bar{\delta_1}\alpha_1} q_1 \xrightarrow{\delta_2\bar{\delta_2}\alpha_2} \cdots q_{n-1} \xrightarrow{\delta_n\bar{\delta_n}\alpha_n} q_n.$$

Un calcul de $\mathcal{A}$, de la forme

$$q_0 \xrightarrow{\delta_1\bar{\delta_1}\alpha_1,\alpha_1} q_1 \xrightarrow{\delta_2\bar{\delta_2}\alpha_2,\alpha_2} \cdots q_{n-1} \xrightarrow{\delta_n\bar{\delta_n}\alpha_n,\alpha_n} q_n.$$

est acceptant si et seulement si $q_n \in F$ et $\alpha_1\alpha_2 \cdots \alpha_n \in D_{k,\boldsymbol{g}}$, c'est-à-dire si et seulement si $\delta_1\bar{\delta_1}\alpha_1\delta_2\bar{\delta_2}\alpha_2 \cdots \delta_n\bar{\delta_n}\alpha_n$ appartient à R et à $D_{k,\boldsymbol{g}'}$. Il en découle l'affirmation ci-dessus. □

Ceci conclut la preuve du théorème ci-dessus. □

7.9 Approfondissements

Afin d'alléger la lecture de ce chapitre, nous avons volontairement omis certaines notions qui n'étaient pas immédiatement nécessaires à notre étude. Nous les abordons ici

7.9.1 Sur les langages de Dyck d'ordres supérieurs

Une propriétés intéressante des langages de Dyck « algébriques » est que pour tout $k \geq 1$, le langage $\mathcal{D}_k$ peut s'exprimer comme l'image inverse par un homomorphisme du langage $\mathcal{D}_2$. Nous retrouvons cette même notion pour les langages de ε-DYCK^k. Nous montrons que pour tout $k \geq 1$, il existe un langage $X \in \varepsilon$-DYCK^k tel que tout langage $Y \in \varepsilon$-DYCK^k peut s'exprimer comme l'image inverse de X par un homomorphisme symétrique. Rappelons premièrement que

— (Définition 7.28) pour toute séquence d'alphabets $\boldsymbol{A}$, l'alphabet $A_k^{\boldsymbol{\beta}}$ est défini comme suit :

$$A_1^{\boldsymbol{\beta}} = A_1\,; \qquad \text{pour tout } k \geq 1 : A_{k+1}^{\boldsymbol{\beta}} = A_k^{\boldsymbol{\beta}} \cup \beta_k(A_k^{\boldsymbol{\beta}}) \cup A_{k+1}.$$

— (Définition 7.31) La séquence de d-projections $\boldsymbol{\sigma}$ sur une séquence d'alphabets $\boldsymbol{A}$ est définie comme suit : Pour tout $k \geq 1 : \sigma_k : \widehat{A_{k+1}^{\boldsymbol{\beta}}}^* \longrightarrow \widehat{A_k^{\boldsymbol{\beta}}}^*$ est tel que :

$$\begin{cases} \sigma_k(a) = \varepsilon, & \forall a \in A_{k+1}; \\ \sigma_k(a) = a, & \forall a \in A_k^{\boldsymbol{\beta}}; \\ \sigma_k((\beta_k, a)) = \bar{a}, & \forall (\beta_k, a) \in \beta_k(A_k^{\boldsymbol{\beta}}). \end{cases}$$

Le langage $\mathcal{D}_{\boldsymbol{A}}^k$ est le langage $D_{k,\boldsymbol{\sigma}}$.

— (Définition 3.1) Un homomorphisme $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ est dit ε-sûr unilatéral si pour tout $u \in \widehat{A}^* : u \in \mathcal{D}_A \Rightarrow f(u) \in \mathcal{D}_B$. L'homomorphisme f est de plus dit fortement ε-sûr unilatéral si « $u \in \mathcal{D}_A \Leftrightarrow f(u) \in \mathcal{D}_B$ » ; c'est-à-dire $\mathcal{D}_A = f^{-1}(\mathcal{D}_B)$.

— (Lemme 4.25) Si un homomorphisme $f : \widehat{A}^* \longrightarrow \widehat{B}^*$ est ε-sûr unilatéral et que A et B sont disjoints, alors la concaténation disjointe $g : \widehat{A}^* \longrightarrow \widehat{(A \cup B)}^*$ telle que pour tout $a \in A : g(a) = af(a)$ et $g(\bar{a}) = f(\bar{a})\bar{a}$ est fortement ε-sûr unilatéral.

Nous montrons premièrement que pour tout langage $X \in \varepsilon$-DYCK^k, il existe une séquence d'alphabet $\boldsymbol{A}$ et un homomorphisme symétrique f tels que $X = f^{-1}(\mathcal{D}_{\boldsymbol{A}}^k)$.

Proposition 7.66. *Pour tout $k \geq 1$ et $X \in \varepsilon$-DYCK^k, il existe une séquence $\boldsymbol{A}$ et un homomorphisme ε-sûr f_k tel que $X = f_k^{-1}(\mathcal{D}_{\boldsymbol{A}}^k)$.*

Démonstration. Nous prouvons ceci par induction sur k.

Cas initial. Pour $k = 1$, l'affirmation est trivialement vraie. Il suffit de considérer f_k comme l'identité.

Pas d'induction. Supposons l'affirmation vraie pour un certain k. Soit alors C_k un alphabet, $\boldsymbol{A}$ une séquence de taille k, $X \subseteq \widehat{C_k}^*$ un langage de ε-DYCK^k et $f_k : \widehat{\boldsymbol{A}_k^{\boldsymbol{\beta}}}^* \longrightarrow \widehat{C_k}^*$ un homomorphisme ε-sûr tel que $X = f_k^{-1}(\mathcal{D}_{\boldsymbol{A}}^k)$.

Nous montrons que pour tout homomorphisme ε-sûr $h : \widehat{C_{k+1}}^* \longrightarrow \widehat{C_k}^*$ il existe une séquence $\boldsymbol{A}'$ de taille $k+1$ et un homomorphisme ε-sûr $f_{k+1} : \widehat{\boldsymbol{A}'^{\boldsymbol{\beta}}_k}^* \longrightarrow \widehat{C_{k+1}}^*$ tels que

$$\mathcal{D}_{C_{k+1}} \cap h^{-1}(X) = {f_{k+1}}^{-1}(\mathcal{D}^{k+1}_{\boldsymbol{A}'}). \tag{7.1}$$

Soit $\boldsymbol{A}'$ la séquence $\boldsymbol{A}$ suivie de C_{k+1} i.e. $\boldsymbol{A}' = A_1, \cdots, A_k, C_{k+1}$ où $\boldsymbol{A} = A_1, \cdots, A_k$. Pour simplifier la lecture, soit B l'alphabet

$$B = A_k^{\boldsymbol{\beta}} \cup \beta_k(A_k^{\boldsymbol{\beta}}).$$

L'alphabet $A'^{\boldsymbol{\beta}}_{k+1}$ est alors égal à $C_{k+1} \cup B$. Nous prouvons les affirmations suivantes.

Affirmation 1. Pour tout $a \in C_{k+1}$: il existe $u_a, u_{\bar{a}} \in \widehat{B}^*$ tels que

$$\sigma_k(u_a) = f_k(h(a)); \quad \sigma_k(u_{\bar{a}}) = f_k(h(\bar{a})); \quad u_a u_{\bar{a}} \in \mathcal{D}_{\boldsymbol{A}'^{\boldsymbol{\beta}}_{k+1}}.$$

Preuve de l'affirmation 1. Par fermeture par composition des homomorphismes ε-sûrs, l'homomorphisme $f_k \circ h$ est ε-sûr. De ce fait, pour tout $a \in C_{k+1}$, le mot $f_k(h(a\bar{a}))$ appartient à $\mathcal{T}_{A_k^{\boldsymbol{\beta}}}$ et d'après le lemme 7.26, celui-ci peut s'exprimer comme $\sigma_k(u)$ avec $u \in \mathcal{D}_B$. Les mots u_a et $u_{\bar{a}}$ forment une décomposition d'un tel mot u telle que $\sigma_k(u_a) = f_k(h(a))$ et $\sigma_k(u_{\bar{a}}) = f_k(h(\bar{a}))$. Cette décomposition est possible puisque σ_k est alphabétique. □

Fixons pour tout $a \in C_{k+1}$ de tels mots u_a et $u_{\bar{a}}$. Soit alors $f_{k+1} : \widehat{C_{k+1}}^* \longrightarrow \widehat{\boldsymbol{A}'^{\boldsymbol{\beta}}_{k+1}}^*$ tel que pour tout $a \in C_{k+1}$:

$$f_{k+1}(a) = a u_a \quad \text{et} \quad f(\bar{a}) = u_{\bar{a}} \bar{a}.$$

Affirmation 2. L'homomorphisme f_{k+1} est fortement ε-sûr unilatéral .

Preuve de l'affirmation 2. Par définition des mots u_a, pour tout $a \in C_{k+1}$: $u_a u_{\bar{a}} \in \mathcal{D}_B$. L'homomorphisme qui associe à tout symbole $a \in C_{k+1}$ le mot u_a et à $\bar{a}$ le mot $u_{\bar{a}}$ est donc ε-sûr unilatéral ; et l'homomorphisme f_{k+1} est une concaténation disjointe de l'identité avec ce dernier. D'après le lemme 4.25, f_{k+1} est fortement ε-sûr unilatéral. □

Affirmation 3. $\sigma_k \circ f_{k+1} = f_k \circ h$.

Preuve de l'affirmation 3. Par définition de f_{k+1} : pour tout $a \in C_{k+1}$: $f_{k+1}(a) = a u_a$ et $f_{k+1}(\bar{a}) = u_{\bar{a}} \bar{a}$. Alors, $\sigma_k(f_{k+1}(a)) = \sigma_k(a)\sigma_k(u_a) = f_k(h(a))$ par définitions de σ_k et de u_a. De même, $\sigma_k(f_{k+1}(\bar{a})) = f_k(h(\bar{a}))$. □

Par les trois affirmations ci-dessus, nous obtenons

$$\begin{aligned}
\mathcal{D}_{C_{k+1}} \cap h^{-1}(X) &= f_{k+1}^{-1}(\mathcal{D}_{A'^{\boldsymbol{\beta}}_k}) \cap h^{-1}(X) && \text{d'après l'affirmation 2} \\
&= f_{k+1}^{-1}(\mathcal{D}_{A'^{\boldsymbol{\beta}}_k}) \cap h^{-1}(f_k^{-1}(\mathcal{D}^k_{\boldsymbol{A}})) && \text{par hypothèse d'induction} \\
&= f_{k+1}^{-1}(\mathcal{D}_{A'^{\boldsymbol{\beta}}_k}) \cap f_{k+1}^{-1}(\sigma_k^{-1}(\mathcal{D}^k_{\boldsymbol{A}})) && \text{d'après l'affirmation 3} \\
&= f_{k+1}^{-1}(\mathcal{D}_{A'^{\boldsymbol{\beta}}_k} \cap \sigma_k^{-1}(\mathcal{D}^k_{\boldsymbol{A}})) && \text{par distributivité} \\
&= f_{k+1}^{-1}(\mathcal{D}^{k+1}_{\boldsymbol{A}'}) && \text{par définition.}
\end{aligned}$$

□

Le langage $\mathcal{D}^k$ désigne le langage $\mathcal{D}^k_{\boldsymbol{B}}$ tel que chaque alphabet B_i de $\boldsymbol{B}$ est de taille 2. Nous montrons maintenant que tout langage $\mathcal{D}^k_{\boldsymbol{A}}$ peut s'exprimer comme l'image inverse par un homomorphisme symétrique du langage $\mathcal{D}^k$.

Proposition 7.67. *Pour toute séquence d'alphabets $\boldsymbol{A}$ et pour tout $k \geq 1$: il existe un homomorphisme f_k tel que $\mathcal{D}^k_{\boldsymbol{A}} = f_k^{-1}(\mathcal{D}^k)$.*

Démonstration. Il s'agit comme attendu du codage binaire symétrique « bin » tel que pour tout k et pour tout $a_i \in A_k$:

$$\mathrm{bin}(a_k) = 0_k 1_k{}^i 0_k; \qquad \mathrm{bin}((\beta_k, a_i)) = (\beta_k, 0_k)(\beta_k, 1_k)^i(\beta_k, 0_k).$$

Remarquons alors que pour tout k et pour tout $a \in \widehat{A^{\boldsymbol{\beta}}_k}^*$: $\sigma_{k-1} \circ \mathrm{bin}(a) = \mathrm{bin} \circ \sigma_{k-1}(a)$. C'est-à-dire $\sigma_{k-1} \circ \mathrm{bin} = \mathrm{bin} \circ \sigma_{k-1}$. Il est de plus bien connu que l'homomorphisme bin est fortement ε-sûr unilatéral. Nous pouvons alors prouver le lemme par induction.

Cas initial. Le cas $k = 1$ est vrai du fait que bin est fortement ε-sûr unilatéral.

Pas d'induction. Supposons le lemme vrai pour un certain k : $\mathcal{D}^k_{\boldsymbol{A}} = \mathrm{bin}^{-1}(\mathcal{D}^k)$. Alors il est aussi vrai pour le cas $k+1$ puisque bin est fortement ε-sûr unilatéral ; c'est-à-dire pour tout $u \in \widehat{A^{\boldsymbol{\beta}}_{k+1}}^*$: $u \in \mathcal{D}_{A^{\boldsymbol{\beta}}_{k+1}}$ si et seulement si $\mathrm{bin}(u) \in \mathcal{D}_{V^{\boldsymbol{\beta}}_{k+1}}$. De plus, par hypothèse d'induction $\sigma_k(u) \in \mathcal{D}^k_{\boldsymbol{A}}$ si et seulement si $\mathrm{bin} \circ \sigma_k(u) = \sigma_k \circ \mathrm{bin}(u) \in \mathcal{D}^k$. En somme, $u \in \mathcal{D}^{k+1}_{\boldsymbol{A}}$ si et seulement si $\mathrm{bin}(u) \in \mathcal{D}^{k+1}$. □

7.9.2 Langages ε-sûrs de niveaux supérieurs

Puisque nous avons une notion de transductions ε-sûres d'ordres supérieurs, il va naturellement avec la notion de langages ε-sûres d'ordres supérieurs qui décrivent les domaines des transductions ε-sûres.

Définition 7.68 (Langages ε-sûrs d'ordres supérieurs)**.** *Soit $k \geq 1$, un langage ε-sûr de niveau k est un langage de la forme*

$$X = f \circ g_{1,k-1}(R \cap D_{k,\boldsymbol{g}})$$

où R est un langage régulier, f est un homomorphisme ε-sûr, et $\boldsymbol{g} = g_1, g_2, \ldots, g_{k-1}$ est une séquence d'homomorphismes ε-sûrs.

Nous désignons par ε-LANG^k l'ensemble de langages ε-sûrs de niveau k/

De part leurs définitions, nous pouvons admettre la proposition suivante

Proposition 7.69. *Soit $k \geq 1$. Un langage X appartient à ε-LANG^k si et seulement si il est reconnaissable par une machine à séquences de niveau k sur une séquence d'homomorphisme ε-sûrs et dans laquelle toute transition est de la forme $(q, f \circ g_{1,k-1}(\alpha), \alpha, p)$.*

Nous avons aussi le lemme suivant

Lemme 7.70. *Chaque famille ε-LANG^k est fermée par homomorphismes ε-sûrs et intersection avec des langages réguliers.*

La fermeture par homomorphismes ε-sûrs découle de la définition même de ε-LANG^k et la fermeture par intersection avec des langages réguliers vient du fait qu'un langage

$$f \circ g_{1,k-1}(R \cap D_{k,\boldsymbol{g}}) \cap R'$$

s'exprime de façon équivalente comme

$$f \circ g_{1,k-1}(R \cap (f \circ g_{1,k-1})^{-1}(R') \cap D_{k,\boldsymbol{g}}).$$

D'après le théorème 7.1, chaque famille LANG^k est le cône généré par ε-DYCK^k. Il s'en suit aussi le lemme suivant :

Lemme 7.71. *Pour tout $k \geq 1$: ε-LANG$^k \subseteq$ LANG^k.*

Nous avons montré que langage est indexé si et seulement si il est l'image par un homomorphisme de l'intersection entre un langage algébrique ε-sûr et le langage de Dyck. Nous montrons dans ce qui suit que chaque famille LANG^k admet une caractérisation similaire.

Considérons premièrement le lemme suivant :

Lemme 7.72. *Soient $k, l \geq 1$. Un langage X appartient à ε-LANG^{k+l} si et seulement si il peut s'exprimer comme $X = f(Y \cap Z)$ où $Y \in \varepsilon$-LANG^k et $Z \in \varepsilon$-DYCK^l.*

Démonstration. ($\Rightarrow$) Soit $\boldsymbol{g} = g_1, g_2, \ldots, g_{k+l-1}$ une séquence d'homomorphismes ε-sûrs, f un homomorphisme ε-sûr et R un langage régulier. Soit $\boldsymbol{\mu} = g_1, g_2, \ldots, g_{l-1}$ la séquence des $l-1$ premiers homomorphismes de $\boldsymbol{g}$. Le langage $f \circ g_{1,k+l-1}(R \cap D_{k+l,\boldsymbol{g}})$ peut alors s'exprimer de façon équivalente comme

$$f \circ g_{1,l-1} \circ g_{l,k+l-1}(R \cap D_{k,\boldsymbol{g}} \cap g^{-1}_{l,k+l-1}(D_{l,\boldsymbol{\mu}})) = f \circ g_{1,l-1}(g_{l,k+l-1}(R \cap D_{k,\boldsymbol{g}}) \cap D_{l,\boldsymbol{\mu}}))$$

($\Leftarrow$) Soient $\boldsymbol{g} = g_1, g_2, \ldots, g_{k-1}$ une séquence d'homomorphismes ε-sûrs de taille $k-1$, f un homomorphisme ε-sûr, R un langage régulier et $Y = f \circ g_{1,k-1}(R \cap D_{k,\boldsymbol{g}})$.

Soit aussi $\boldsymbol{h} = h_1, h_2, \ldots, h_{l-1}$ une séquence d'homomorphismes ε-sûrs et

$$\boldsymbol{\mu} = h_1, h_2, \ldots, h_{l-1}, f, g_1, g_2, \ldots, g_{k-1}$$

l'extension de $\boldsymbol{h}$ par f et $\boldsymbol{g}$. Le langage $Y \cap D_{l,\boldsymbol{h}}$ s'exprime alors comme

$$\begin{aligned} f \circ g_{1,k-1}(R \cap D_{k,\boldsymbol{g}}) \cap D_{l,\boldsymbol{h}} &= f \circ g_{1,k-1}(R \cap D_{k,\boldsymbol{g}} \cap (f \circ g_{1,k-1})^{-1}(D_{l,\boldsymbol{h}})) \\ &= f \circ g_{1,k-1}(R \cap D_{k+l,\boldsymbol{\mu}}). \end{aligned}$$

Ce langage appartient donc à ε-LANG^{k+l}. □

Conjointement à la fermeture par transduction rationnelle des familles LANG^k, le lemme 7.72 implique que

Lemme 7.73. *Pour tous $k, l \geq 1$ tout langage de la forme $f(Y \cap Z)$ où $Y \in \varepsilon$-LANG^k, $Z \in \varepsilon$-DYCK^l et f est un homomorphisme appartient à* LANG^{k+l}.

Nous montrons maintenant la réciproque. Considérons premièrement le lemme suivant :

Lemme 7.74. *Pour tout $k \geq 1$, il existe un langage $L_k \in \varepsilon$-LANG^k et un homomorphisme f tel que $f(L_k) = \lambda_{1,k}(\mathcal{D}^k_{\boldsymbol{A}})$.*

Démonstration. Le lemme est trivialement vrai pour $k = 1$. Nous montrons par construction qu'il est vrai pour $k > 1$.

Soit $k \geq 1$. Nous considérons ici un nouveau marqueur γ. Soit $\Sigma_{k+1} = A_{k+1}^{\boldsymbol{\beta}}$ et pour tout $i \in \{1, \ldots, k\}$, soient

$$\Gamma_i = A_{i+1,k+1} \cup \gamma(A_{i+1,k+1}), \quad \Sigma_i = \Gamma_i \cup A_i^{\boldsymbol{\beta}}.$$

Considérons aussi pour tout $i \in \{1, \ldots, k\}$, l'homomorphisme $g_i : \widehat{\Sigma_{i+1}}^* \to \widehat{\Sigma}_i^*$ tel que :

$$a \in A_{i+1} \mapsto a\bar{a}, \;\; a \in \bar{A}_{i+1} \mapsto \gamma(a)\overline{\gamma(a)}, \quad a \in \widehat{\Gamma}_{i+1} \mapsto a, \quad a \in \widehat{A_{i+1}^{\boldsymbol{\beta}}} \mapsto \sigma_{i+1}(a).$$

Considérons pour tout $i \in \{0, \ldots, k+1\}$ le langage $L_i \subseteq \widehat{\Sigma_{k+1-i}}^*$ défini comme suit :

$$L_0 = \widehat{\Sigma_{k+1}}^*; \qquad L_i = g_{k+1-i}(L_{i-1} \cap \mathcal{D}_{\Sigma_{k+2-i}}) \text{ pour } 0 < i \leq k; \qquad \text{et } L_{k+1} = \mathcal{D}_{\Sigma_1} \cap L_k.$$

Affirmation 1. Pour tout $i \in \{1, \ldots, k+1\}$:

$$L_{k+1} = g_{1,i-1}(D_{j,\boldsymbol{g}} \cap L_{k-\ell+1}). \tag{7.2}$$

Preuve de l'affirmation 1. L'affirmation est trivialement vraie pour $i = 1$. Supposons la vraie pour un $i < k+1$ et montrons qu'elle est vraie pour $i+1$. Par hypothèse d'induction, nous avons

$$\begin{aligned} L_k + 1 &= g_{1,i-1}(D_{i,\boldsymbol{g}} \cap L_{k+1-i}) = g_{1,i-1}(D_{i,\boldsymbol{g}} \cap g_i(\mathcal{D}_{\Sigma_{i+1}} \cap L_{k-i})) \\ &= g_{1,i-1} \circ g_i(\mathcal{D}_{\Sigma_i+1} \cap {g_i}^{-1}(D_{i,\boldsymbol{g}}) \cap L_{k-i}) = g_{1,i}(D_{i+1,\boldsymbol{g}} \cap L_{k-i}). \end{aligned}$$

□

Puisque $L_0 = \widehat{\Sigma}_{k+1}^*$, en prenant $i = k+1$ dans (7.2) nous obtenons :

$$L_{k+1} = g_{1,k}(D_{k+1,\boldsymbol{g}}). \tag{7.3}$$

Pour simplifier les expressions qui suivent, nous désignons par B_i l'alphabet $A_i^{\boldsymbol{\beta}}$. Remarquons maintenant que par définition, pour tout j, $\pi_{\widehat{\Sigma}_{j+1},\widehat{B_{j+1}}} \circ {g_j}^{-1} = {\sigma_j}^{-1} \circ \pi_{\widehat{\Sigma}_j,\widehat{B}_j}$. De même, pour tout $i \in \{1, \ldots, \leq k\}$,

$$\sigma_{i,k}^{-1} \circ \pi_{\widehat{\Sigma}_i,\widehat{B}_i} = \pi_{\widehat{\Sigma_{k+1}},\widehat{B_{k+1}}} \circ {g_{i,k}}^{-1}$$

Et puisque $\Sigma_{k+1} = B_{k+1}$:

$$\sigma_{i,k}^{-1} \circ \pi_{\widehat{\Sigma}_i,\widehat{B}_i} = {g_{i,k}}^{-1}.$$

Combiné au fait que $\pi_{\widehat{\Sigma}_j,\widehat{B}_j}(\mathcal{D}_{\Sigma_j}) = \mathcal{D}_{B_j}$, pour tout $1 \leq j \leq k+1$, nous obtenons :

$$\begin{aligned} \mathcal{D}_{\boldsymbol{A}}^{k+1} &= \mathcal{D}_{B_{k+1}} \cap \bigcap_{1 \leq i \leq k} \sigma_i^{-1}(\mathcal{D}_{B_i}) = \mathcal{D}_{B_{k+1}} \cap \bigcap_{1 \leq i \leq k} \sigma_i^{-1}(\pi_{\widehat{\Sigma}_i,\widehat{B}_i}(\mathcal{D}_{\Sigma_i})) \\ &= \mathcal{D}_{\Sigma_{k+1}} \cap \bigcap_{1 \leq i \leq k} {g_{i,k}}^{-1}(\mathcal{D}_{\Sigma_i})) = D_{k+1,\boldsymbol{g}}. \end{aligned} \tag{7.4}$$

Considérons maintenant l'homomorphisme $f : \widehat{\Sigma_1}^* \to \widehat{A}_{1,k+1}$ défini comme suit

$$a \in A_{1,k+1} \cup \overline{A}_1 \mapsto a, \quad a \in \overline{A_{2,k+1}} \cup \overline{\gamma(A_{1,k})} \mapsto \varepsilon, \quad : \gamma(\alpha) \in \gamma(A_{2,k}) \mapsto \bar{\alpha}.$$

Nous avons ainsi $f \circ g_{1,k} = \lambda_{1,k}$. Par conséquent

$$\begin{aligned} f(L_{k+1}) &= f \circ g_{1,k}(D_{k+1,\boldsymbol{g}}) && \text{d'après (7.3)} \\ &= \lambda_{1,k}(D_{k+1,\boldsymbol{g}}) \\ &= \lambda_{1,k}(\mathcal{D}^{k+1}_{\boldsymbol{A}}) && \text{d'après (7.4).} \end{aligned}$$

Ceci conclut la preuve du lemme. Remarquons toutefois que l'homomorphisme f n'est pas ε-sûr. □

Théorème 7.6. *Soient $k, l \geq 1$. Un langage L appartient à LANG^{k+l} si et seulement si il peut s'écrire comme $f(X \cap Y)$ avec $X \in \varepsilon\text{-LANG}^k$, $Y \in \varepsilon\text{-DYCK}^l$ et f est un homomorphisme.*

Démonstration. ($\Leftarrow$) D'après le lemme 7.73.

($\Rightarrow$)Soit $L \in \text{LANG}^k$. D'après les propositions 7.46 et 7.44, le langage L peut s'écrire comme $f(R \cap \lambda_{1,k-1}(\mathcal{D}^k_{\boldsymbol{A}}))$ pour un certain homomorphisme f, langage régulier R et séquence d'alphabets $\boldsymbol{A}$. D'après le lemme 7.74, L peut alors s'écrire comme $L = f(R \cap h(M))$ où h est un homomorphisme et $M \in \varepsilon\text{-LANG}^k$. Le langage L peut alors s'écrire comme $L = f \circ h(M')$ avec $M' = M \cap h^{-1}(R)$ qui par fermeture par intersection avec des réguliers de $\varepsilon\text{-LANG}^{k+l}$ (Lemme 7.70) appartient à $\varepsilon\text{-LANG}^{k+l}$. D'après le lemme 7.72, le langage M' peut s'exprimer comme $\mu(X \cap Y)$ avec $X \in \varepsilon\text{-LANG}^k$ et $Y \in \varepsilon\text{-DYCK}^l$. Ceci conclut la preuve du théorème. □

Chapitre 8

Les Langages Les Plus Durs

8.1 Introduction

Le langage algébrique le plus dur. La complexité de la reconnaissance des langages algébriques a fait l'objet de plusieurs études. Dans [Gre73], Greibach montre l'existence d'un langage algébrique L_0 tel que un langage L est algébrique si et seulement si il peut s'écrire comme $L = \$^{-1} \cdot f^{-1}(L_0)$ où f est un homomorphisme (qui peut être effectivement calculé depuis une grammaire en forme normale de Greibach le générant) [8]. Ainsi, un mot u appartient à L si et seulement si $f(\$u)$ appartient à L_0. Le problème de l'appartenance d'un mot *($u \in L$?)* pour tout langage algébrique se réduit alors à celui de L_0 et la complexité en temps de l'appartenance à L_0 est la plus petite borne supérieur de la complexité du problème de l'appartenance pour les langages algébriques. Le langage L_0 susmentionné est alors comme « le langage algébrique le plus dur de Greibach ». Il existe plusieurs preuves de ce résultat dans la littérature [GS80, Sto] et d'extensions à d'autre familles de langages (voir par exemple [Okh16]). La question générale que nous abordons ici est : *quelles propriétés suffisent à ce qu'une famille ait un tel langage ?*

Généralisations du langage algébrique le plus dur. La famille des langages algébriques est un cône fidèle principal [CS63, BN73, Okh12] fermé par substitutions. Nous montrons ici que tout cône fidèle principal $\mathcal{F}$ fermé par substitutions algébriques admet un langage L_0 tel que tout langage $L \in \mathcal{F}$ peut être décrit $L = \$^{-1} \cdot f^{-1}(L_0)$. De plus, si un langage de $\mathcal{F}$ contient le mot vide, alors $\mathcal{F}$ est fermé par quotient gauche et tout langage pouvant se représenter comme $\$^{-1} \cdot f^{-1}(L_0)$ appartient $\mathcal{F}$. Il en est de même si $\mathcal{F}$ est fermé par substitutions et contient le langage $X = \{\varepsilon\} \cup \{x\bar{x} \mid x \in 01^+0 \cup 0'1'^+0'\}$. Un corollaire de ce résultat est que pour tout $k > 1$, la famille q-LANG^k, de langages reconnaissables par automates à k-pile « quasi-réels » admet un tel langage L_0.

L'ingrédient principal de notre preuve est que pour tout alphabet A, il existe une transduction algébrique fidèle et continue τ_0 telle que $\mathrm{dom}(\tau_0) \subseteq A^*$ et telle que toute transduction rationnelle fidèle τ telle que $\mathrm{dom}(\tau) \subseteq A^*$ peut s'écrire comme $\tau = \$^{-1} \circ f^{-1} \circ \tau_0$ où f est un homomorphisme. La transduction τ_0 mentionnée ci-dessus associe à tout mot u un ensemble de codages de tous les calculs possibles d'un automate fini sur u et utilise le langage X mentionné plus haut pour décrire ce codage. Notre preuve utilise des arguments similaires à ceux de la preuve du langage le plus dur de Greibach donnée dans [ABB97].

8. Originalement énoncé comme $M = f^{-1}(L_0)$ ou $M = f^{-1}(L_0 - \varepsilon)$.

Travaux connexes. Dans [YW84], Yokomori et Wood prouvent que toute famille agréable fermée par substitutions algébriques contient un langage L_0 tel que tout langage peut s'exprimer comme $gf^{-1}(L_0)$ où g et f sont des homomorphismes. Dans [Tur78] Turakainen prouve un résultat similaire au notre utilisant la présence d'autres langages algébriques linéaires comme condition pour la caractérisation par homomorphismes inverses.

8.2 Préliminaires

Rappels : Cônes rationnels fidèles.

- Une transduction $\tau : A^* \times B^*$ est dite *croissante* si pour tout $(u, v) \in \tau : |u| \leq |v|$. Elle est *fidèle* si pour tout $v \in B^*$, l'ensemble $\tau^{-1}(v)$ est fini. Elle est dite *continue* si pour tout couple $(u, v) \in \tau : v = \varepsilon \Rightarrow u = \varepsilon$.
  Toute transduction croissante est alors fidèle et continue.
- (Théorème 1.6) Une transduction τ est rationnelle fidèle et continue si et seulement si elle peut s'exprimer comme $\tau = \{(g(u), h(u)) \mid u \in R\}$ où R est un langage régulier, g un homomorphisme et h un homomorphisme strictement alphabétique.
  Nous aurons besoin de composer des transductions, nous présentons les transductions sous forme fonctionnelles ($\tau : A^* \longrightarrow 2^{B^*}$) et nous employons la notation d'Eilenberg : La transduction τ susmentionnée sera écrite $\tau = h \circ \cap R \circ g^{-1}$.
- Un cône fidèle est une famille de langage fermée par transductions rationnelles *fidèles et continues*.

Rappels : Transductions algébriques. (Théorème 2.6) Une transduction $\tau : A^* \longrightarrow 2^{B^*}$ est algébrique si et seulement si elle peut s'exprimer comme $\tau = h \circ \cap K \circ g^{-1}$ où h et g sont des homomorphismes et K est un langage algébrique.

Équivalence de transductions. Deux transductions τ_1 et τ_2 sont dites équivalentes si $\mathrm{dom}(\tau_1) = \mathrm{dom}(\tau_2)$ et pour tout $u \in \mathrm{dom}(\tau_1) : \tau_1(u) = \tau_2(u)$.

Rappels : Substitutions.

- Une substitution est un morphisme $\nu : A^* \longrightarrow 2^{B^*}$ qui a tout symbole $a \in A$ associe un langage L_a. Puisque ν est un morphisme, $\nu(\varepsilon) = \{\varepsilon\}$ et pour tout $u, v \in A^* : \nu(uv) = \nu(u)\nu(v)$. De plus, pour tout langage $L \subseteq A^* : \nu(L) = \bigcup_{u \in L} \nu(u)$.
- Une substitution est dite non-effaçante si pour tout $a \in A : \nu(a)$ ne contient pas le mot vide.
- Soit $\mathcal{F}$ une famille de langage. Une $\mathcal{F}$-substitution est une substitution $\nu : A^* \longrightarrow 2^{B^*}$ telle que pour tout $a \in A : \nu(a) \in \mathcal{F}$.
  La famille $\mathcal{F}$ est simplement dite *fermée par substitution* si elle est fermée par $\mathcal{F}$-substitution.

8.3 Un Générateur de Transductions Fidèles

Dans cette section, nous montrons que pour tout alphabet A, il existe une transduction algébrique fidèle et continue $\tau_0 : A^* \longrightarrow 2^{B^*}$ telle que pour toute transduction rationnelle fidèle $\tau : A^* \longrightarrow C^*$, on peut trouver un homomorphisme $f_\tau : (C \cup \{\$\})^* \longrightarrow B^*$ tel que $\tau = \$^{-1} \circ f_\tau^{-1} \circ \tau_0$. L'idée ici est que la transduction τ_0 associe à tout mot u un ensemble de codages de tous les calculs possibles d'un automate fini sur u. L'homomorphisme f_τ, essaiera quant à lui d'associer à tout mot de la forme $\$v$ un codage des calculs d'un transducteurs

décrivant τ sur tous les mots u tels que $v \in \tau(u)$. L'intention est que « pour tous mots u et v, $v \in \tau(u) \Leftrightarrow f_\tau(\$v) \in \tau_0(u)$ ».

Définition 8.1. *Soient deux ensembles finis Q et F tels que $F \subseteq Q$ et $F \neq \emptyset$. Soit la transduction $\Phi_{Q,F} : A^* \longrightarrow 2^{(A \cup \widehat{Q})^*}$ telle que pour tout mot $u \in A^*$:*

$$\Phi_{Q,F}(u) = \{q_1\overline{q_1}u_1q_2\overline{q_2}u_2 \cdots q_n\overline{q_n}u_nq_{n+1}, \mid n \geq 0, u_1u_2 \cdots u_n = u \\ \forall i \in \{1, \ldots, n\} : u_i \in A^*, q_i \in Q \text{ et } q_{n+1} \in F\}.$$

Exemple 8.2. *Soit $Q = \{q_1, q_2\}$, $F = \{q_2\}$ et $u = aabb$. Alors*

$$q_2\overline{q_2}\ aq_1\overline{q_1}\ q_2\overline{q_2}\ abq_1\overline{q_1}\ bq_2 \in \Phi_{Q,F}(u).$$

Définition 8.3. *Soit $\Phi_{\text{bin}} : A^* \longrightarrow 2^{(A \cup \widehat{\{0,1,0',1'\}})^*}$ tel que pour tout mot $u \in A^*$:*

$$\Phi_{\text{bin}}(u) = \{x_1\overline{x_1}u_1x_2\overline{x_2}u_2 \cdots x_n\overline{x_n}u_nx_{n+1} \mid n \geq 0, u_1u_2 \cdots u_n = u, \\ \forall i \in \{1, \ldots, n\} : u_i \in A^*, x_i \in 01^+0 + 0'1'^+0', \\ x_{n+1} \in 0'1'^+0'\}.$$

La transduction $\Phi_{Q,F}$ a pour vocation de décrire les calculs acceptants d'une machine sur un mot. Celle-ci ayant Q comme ensemble d'états et F comme ensemble d'états acceptants. Chaque facteur $\overline{q_i}u_iq_{i+1}$ décrit une transition d'un calcul sur u. La transduction Φ_{bin} a pour vocation de décrire les images de toutes les possibles transductions $\Phi_{Q,F}$.

Lemme 8.4. — *Pour tous ensembles finis Q et F tels que $F \subseteq Q$ et $F \neq \emptyset$, la transduction $\Phi_{Q,F}$ est rationnelle, fidèle et continue.*
— *La transduction Φ_{bin} est algébrique, fidèle et continue.*

Démonstration. Toute transduction $\Phi_{Q,F}$ est rationnelle, et Φ_{bin} algébriques car elles peuvent toutes les deux se décrire comme $\cap X \circ \pi_A^{-1}$ avec $X = (\{q\bar{q} \mid q \in Q\}A^*)^*F$ dans le cas de la transduction $\Phi_{Q,F}$ et $X = (\{x\bar{x} \mid x \in 01^+0 \cup 0'1'^+0'\}A^*)^*0'1'^+0'$ pour Φ_{bin}. De plus, elles sont fidèles et continues puisque croissantes. □

Lemme 8.5. *Pour tout ensembles Q et F tels que $F \subseteq Q$ et $F \neq \emptyset$, il existe un homomorphisme f tel que $\Phi_{Q,F} = f^{-1} \circ \Phi_{\text{bin}}$.*

Démonstration. L'homomorphisme recherché est $f : (A \cup \widehat{Q})^* \longrightarrow (A \cup \widehat{\{0,1,0',1'\}})^*$ tel que $f(a) = a$ pour tout $a \in A$; pour tout $q_i \in Q$:

$$f(q_i) = \begin{cases} 0'1'^i0' \text{ si } q_i \in F, \\ 01^i0 \text{ sinon;} \end{cases} \quad \text{et } f(\overline{q_i}) = \overline{f(q_i)}.$$

Nous pouvons clairement affirmer que pour tout $u \in A^*$ et $v \in (A \cup \widehat{Q})^* : v \in \Phi_{Q,F}(u)$ si et seulement si $f(v) \in \Phi_{\text{bin}}(u)$; c'est à dire $\Phi_{Q,F}(u) = f^{-1}(\Phi_{\text{bin}}(u))$ pour tout mot u. □

Comme moyen de décrire des ensembles finis sous forme de mots, nous utiliserons une certaine forme d'expressions régulières communément appelées motifs.

Notation. Étant donné un alphabet A, nous dénotons par $\underline{A}$ l'alphabet $A \cup \{(,),+\}$.

Définition 8.6. *Un motif sur l'alphabet A est un mot $\omega \in \underline{A}^*$ de la forme*

$$\omega = (u_{1,1} + \cdots + u_{1,n_1})(u_{2,1} + \cdots + u_{2,n_2}) \cdots (u_{m,1} + \cdots + u_{m,n_m})$$

avec $m \geq 0$, chaque $n_i \geq 1$ et chaque $u_{i,j} \in A^*$. Nous désignons par $\mathscr{L}(\omega)$ le langage décrit par ω vu comme une expression régulière et $\mathrm{Patt}(A)$ *désigne l'ensemble de motifs sur l'alphabet A.*

Définition 8.7. *Soit* $\mathrm{ND} : A^* \longrightarrow 2^{\underline{A}^*}$ *la transduction qui a tout mot u associe*

$$\mathrm{ND}(u) = \{\omega \in \mathrm{Patt}(A) \mid u \in \mathscr{L}(\omega)\}.$$

Exemple 8.8. *Soit $u = aab$. Le mot $(a+b)(bb+a)(bbb+b)$ appartient à* $\mathrm{ND}(u)$.

Note.

— *Pour tout $L \subseteq A^*$, le langage* $\mathrm{ND}(L)$ *est alors*

$$\{\omega \in \mathrm{Patt}(A) \mid \exists u \in \mathscr{L}(\omega) : u \in L\}.$$

Celui-ci est communément appelé « version non-déterministe » de L.

— *La transduction* ND *est rationnelle [ABB97]. De plus, celle-ci est croissante, et par conséquent, fidèle et continue.*

Lemme 8.9. *Pour tout ensembles finis Q et F tels que $F \subseteq Q$ et $F \neq \emptyset$, il existe un homomorphisme h tel que* $\mathrm{ND} \circ \Phi_{Q,F} = h^{-1} \circ \mathrm{ND} \circ \Phi_{\mathrm{bin}}$.

Démonstration. Considérons l'homomorphisme f du lemme 8.5 et $h : (\underline{A} \cup \widehat{Q})^* \longrightarrow (\underline{A} \cup \widehat{\{0,1,0',1'\}})^*$ l'homomorphisme tel que pour tout $a \in A \cup \widehat{Q} : h(a) = f(a)$, et pour tout $a \in \{(,),+\} : h(a) = a$. Nous pouvons alors remarquer que pour tout $\omega \in \mathrm{Patt}(A \cup \widehat{Q})$, étant de la forme $(u_{1,1} + \cdots + u_{1,n_1}) \cdots (u_{m,1} + \cdots + u_{m,n_m})$, $h(\omega)$ est de la forme

$$(f(u_{1,1}) + \cdots + f(u_{1,n_1})) \cdots (f(u_{m,1}) + \cdots f(u_{m,n_m})).$$

Ainsi, nous avons $\mathscr{L}(h(\omega)) = f(\mathscr{L}(\omega))$. Soit alors $u \in A^*$ et $\omega \in \mathrm{Patt}(A \cup \widehat{Q})$. Nous voulons montrer que

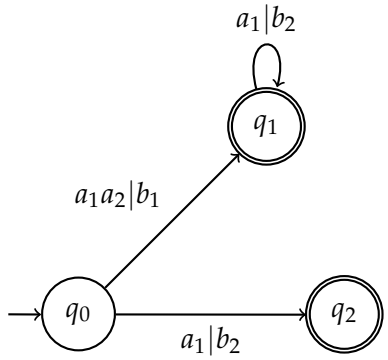
$$\omega \in \mathrm{ND}(\Phi_{Q,F}(u)) \Leftrightarrow h(\omega) \in \mathrm{ND}(\Phi_{\mathrm{bin}}(u))$$

c.-à-d. $\exists x \in \mathscr{L}(\omega)$ t.q. $x \in \Phi_{Q,F}(u) \Leftrightarrow \exists y \in f(\mathscr{L}(\omega))$ t.q. $y \in \Phi_{\mathrm{bin}}(u)$

Ce qui est vrai d'après le lemme 8.5 combiné à l'injectivité de f. □

Lemme 8.10 ([Gin75])**.** *Une transduction $\tau : A^* \longrightarrow 2^{B^*}$ est rationnelle, fidèle et continue si et seulement si elle peut être décrite par un transducteur fini $(Q, A, B, \Delta, q_0, F)$ dans lequel toute transition est de la forme $(q, u|b, p)$ avec $b \in B$, $u \in A^*$.*

Le lemme ci-dessus découle de la caractérisation homomorphique des transductions rationnelles fidèles et continues. Toute transduction rationnelle fidèle et continue peut être décrite comme $\tau = f \circ \cap R \circ g^{-1}$ avec f strictement alphabétique, g alphabétique et R est un langage régulier. Le transducteur correspondant obtenue en transformant toute transition (q, a, p) de l'automate fini reconnaissant R en $(q, g(a)|f(a), p)$. Réciproquement, toute transduction rationnelle décrite par un tel transducteur est fidèle et continue.



$$\mu : \begin{cases} b_1 \mapsto \{\overline{q_0}a_1a_2q_1\} \\ b_2 \mapsto \{\overline{q_0}a_1q_2, \overline{q_1}a_1q_1\} \end{cases}$$

$$\phi : \begin{cases} \$ \mapsto (q_0) \\ b_1 \mapsto (\overline{q_0}a_1a_2q_1) \\ b_2 \mapsto (\overline{q_0}a_1q_2 + \overline{q_1}a_1q_1) \end{cases}$$

FIGURE 8.1 – Illustration de la preuve de la proposition 8.11.

Proposition 8.11. *Pour toute transduction rationnelle fidèle et continue $\tau : A^* \longrightarrow 2^{B^*}$, il existe un homomorphisme $f : (B \cup \{\$\})^* \longrightarrow (\underline{B} \cup \widehat{\{0,1,0',1'\}})^*$ tel que $\tau = \$^{-1} \circ f^{-1} \circ \mathrm{ND} \circ \Phi_{\mathrm{bin}}$.*

La preuve ci-dessous emploie des arguments similaires à ceux de la preuve du langage le plus dur de Greibach donnée dans [ABB97]. Voir la figure 8.1 pour une illustration de la preuve.

Démonstration. Nous souhaitons montrer l'existence d'un homomorphisme f tel que pour tout $u \in A^*$ et $v \in B^*$, $v \in \tau(u)$ si et seulement si $f(\$v) \in \mathrm{ND}(\Phi_{\mathrm{bin}}(u))$. Remarquons premièrement que la proposition est vraie pour $\tau = \emptyset$. Il suffit de considérer un homomorphisme f tel que pour tout mot v, $f(v)$ n'est pas un motif. Nous pouvons donc supposer dans ce qui suit que $\tau \neq \emptyset$. De plus, d'après le lemme 8.10, il existe un transducteur fini $\mathcal{T} = (Q, A, B, q_0, \Delta, F)$ tel que $\tau = [\![\mathcal{T}]\!]$ et dans lequel toute transition de Δ est de la forme $(q, u|b, p)$ avec $b \in B$. Soit $\mu : B^* \longrightarrow 2^{(\widehat{Q} \cup A)^*}$ la substitution telle que

$$\mu(b) = \{\bar{q}up \mid (q, u|b, p) \in \Delta\} \qquad \text{pour tout } b \in B.$$

Nous émettons les affirmations suivantes.

Affirmation 1. Pour tout $u \in A^*$, $v \in B^* : v \in \tau(u) \Leftrightarrow \exists x \in q_0\mu(v) : x \in \Phi_{Q,F}(u)$.

Preuve de l'affirmation 1. On peut observer par définition de μ, qu'il existe un calcul $q_0 \xrightarrow{u_1|b_1} q_1 \xrightarrow{u_2|b_2} \cdots q_{n-1} \xrightarrow{u_n|b_n} q_n$ si et seulement si le mot $q_0\overline{q_0}u_1q_1\overline{q_1}u_2q_2 \cdots \overline{q_{n-1}}u_nq_n$ appartient à $q_0\mu(b_1b_2 \cdots b_n)$. Le mot $v_1v_2 \cdots v_n$ appartient à $\tau(u)$ si et seulement si il existe un tel calcul avec $u_1u_2 \cdots u_n = u$ et $q_n \in F$, c'est à dire, si et seulement si $q_0\overline{q_0}u_1q_1\overline{q_1}u_2q_2 \cdots \overline{q_{n-1}}u_nq_n$ appartient à $\Phi_{Q,F}(u)$. □

Considérons maintenant l'homomorphisme $\phi : (B \cup \{\$\})^* \longrightarrow (\underline{A} \cup \{\$\} \cup \widehat{Q})^*$ tel que $\phi(\$) = (q_0)$ et pour tout $b \in B$:

$$\phi(b) = (m_1 + m_2 + \cdots + m_n) \Leftrightarrow \mu(b) = \{m_1, m_2, \ldots, m_n\}.$$

Pour tout $b \in B$, nous avons $\mu(b) = \mathscr{L}(\phi(b))$. Par conséquent, nous pouvons émettre l'affir-

mation suivante

Affirmation 2. Pour tout mot $v \in B^*$: $q_0\mu(v) = \mathscr{L}(\phi(\$v))$.

En combinant les deux affirmations, nous obtenons : *pour tout $u \in A^*$ et $v \in B^*$:*

$$\begin{aligned} v \in \tau(u) &\Leftrightarrow \exists x \in q_0\mu(v) \text{ t.q. } x \in \Phi_{Q,F}(u) \\ &\Leftrightarrow \exists x \in \mathscr{L}(\phi(\$v)) \text{ t.q. } x \in \Phi_{Q,F}(u) \\ &\Leftrightarrow \phi(\$v) \in \mathrm{ND}(\Phi_{Q,F}(u)). \end{aligned}$$

C'est à dire $\tau(u) = \$^{-1} \cdot \phi^{-1}(\mathrm{ND}(\Phi_{Q,F}(u))$. Combiné au lemme 8.9, nous obtenons

$$\tau(u) = \$^{-1} \cdot \phi^{-1} \circ f^{-1}(\mathrm{ND}(\Phi_{\mathrm{bin}}(u))) = \$^{-1} \cdot h^{-1}(\mathrm{ND}(\Phi_{\mathrm{bin}}(u)))$$

pour un certain homomorphisme f et $h = f \circ \phi$. La proposition ci-dessus est ainsi prouvée. □

Exemple 8.12. *Soit $T = (Q, A, B, \Delta, q_0, F)$ le transducteur illustré sur la figure 8.1 et τ la transduction réalisée par T. Soit $u = a_1a_2a_1$ et $v = b_1b_2$. On a $v \in \tau(u)$ mais aussi*

$$\phi(\$v) = (q_0)(\overline{q_0}a_1a_2q_1)(\overline{q_0}a_1q_2 + \overline{q_1}a_1q_1) \in \mathrm{ND}(\Phi_{Q,F}(u))$$

puisque $q_0\overline{q_0}a_1a_2q_1\overline{q_1}a_1q_1 \in \Phi_{Q,F}(u)$. D'après le lemme 8.9, nous aussi

$$f \circ \phi(\$v) = (010)(\overline{010}a_1a_20'1'^20')(\overline{010}a_10'1'^30' + \overline{0'1'^20'}a_10'1'^20') \in \mathrm{ND}(\Phi_{\mathrm{bin}}(u))$$

où f décrit le codage binaire des états $q \in Q$.

Nous pouvons aussi « relacher » la contrainte sur la continuité de la proposition 8.11 comme suit.

Théorème 8.1. *Pour toute transduction rationnelle fidèle $\tau : A^* \longrightarrow 2^{B^*}$, il existe un homomorphisme $f : (B \cup \{\$\})^* \longrightarrow (\underline{B} \cup \{\widehat{0, 1, 0', 1'}\})^*$ tel que $\tau = \$^{-1} \circ f^{-1} \circ \mathrm{ND} \circ \Phi_{\mathrm{bin}}$.*

Démonstration. La preuve s'appuie sur l'observation suivante. Pour toute transduction rationnelle fidèle τ, il existe à cause même de la fidélité de τ, un nombre fini de mots x tels que $\varepsilon \in \tau(x)$. Soit X l'ensemble de ces mots. La transduction τ peut alors être décrite comme $\tau = \tau_\varepsilon \cup \tau_c$ où τ_ε est une transduction finie qui associe à chaque mot $x \in X$ le mot vide ε et $\tau_c = \tau - \tau_\varepsilon$ est la partie continue de τ. Par conséquent, τ peut être décrit par un transducteur $(Q, A, B, \Delta, q_0, F)$ doté d'un état acceptant particulier $q_\varepsilon \in F$ pour lequel il n'existe pas de transition sortante mais un nombre fini de transitions entrantes de la forme $(q_0, x|\varepsilon, q_\varepsilon)$ avec $x \in X$; et toute autre transition de Δ est de la forme $(q, u|b, p)$ avec $b \in B$. Nous pouvons alors adapter la preuve de la proposition 8.11 en définissant $\varphi(\$) = (q_0 + q_0\overline{q_0}x_1q_\varepsilon + \cdots + q_0\overline{q_0}x_{|X|}q_\varepsilon)$ avec chaque $x_i \in X$. □

Corollaire 8.13. *Pour tout alphabet A, il existe une transduction algébrique fidèle et continue τ_0 telle que pour toute transduction rationnelle fidèle τ, il existe un homomorphisme f tel que $\tau = \$^{-1} \circ f^{-1} \circ \tau_0$.*

Démonstration. Puisque que la composition d'une transduction algébrique et d'une transduction rationnelle est algébrique [Ber79], et que la fidélité et la continuité sont préservées par composition, la transduction $\tau_0 = \mathrm{ND} \circ \Phi_{\mathrm{bin}}$ est algébrique. Combiné au théorème 8.1, nous obtenons le corollaire ci-dessus. □

8.4 Généralisations du langage algébrique le plus dur

Commençons par ce corollaire de la proposition 8.11.

Corollaire 8.14. *Soit L un langage et $\mathcal{F} = \mathscr{C}^f(L)$. Si $\Phi_{\text{bin}}(L) \in \mathcal{F}$, alors il existe un langage $L_0 \in \mathcal{F}$ tel que tout langage $M \in \mathcal{F}$ peut s'exprimer comme $M = \$^{-1} \cdot f^{-1}(L_0)$. Si de plus $\mathcal{F}$ est fermé par quotient gauche, alors un langage M appartient à $\mathcal{F}$ si et seulement si il peut s'exprimer comme $M = \$^{-1} \cdot f^{-1}(L_0)$.*

Démonstration. Le langage L_0 mentionné ici est $L_0 = \text{ND}(\Phi_{\text{bin}}(L))$. Puisque tout langage $M \in \mathcal{F}$ peut s'exprimer comme $M = \tau(L)$ où τ est une transduction rationnelle fidèle et continue, alors d'après la proposition 8.11, un tel langage M peut s'exprimer comme

$$\tau(L) \;=\; \$^{-1} \cdot f^{-1}(\text{ND}(\Phi_{\text{bin}}(L))) \;=\; \$^{-1} \cdot f^{-1}(L_0)$$

pour un certain homomorphisme f. Si de plus $\mathcal{F}$ est fermé par quotient gauche, alors tout langage pouvant s'exprimer comme $\$^{-1} \cdot f^{-1}(L_0)$ appartient à $\mathcal{F}$. □

Dans ce qui suit, nous montrons des conditions supplémentaires sous lesquelles un cône fidèle principal $\mathcal{F} = \mathscr{C}^f(L)$ puisse être fermé par quotient gauche et contient $\Phi_{\text{bin}}(L)$. Commençons par la fermeture par quotient gauche.

Lemme 8.15. *Soit L un langage et $\mathcal{F} = \mathscr{C}^f(L)$. Les affirmations suivantes sont équivalentes.*

1. *$\mathcal{F}$ est fermé par quotient gauche.*
2. *Il existe un langage $M \in \mathcal{F}$ qui contient ε.*
3. *$\varepsilon \in L$.*
4. *Pour tout $M \in \mathcal{F}$: le langage $M \cup \{\varepsilon\}$ appartient à $\mathcal{F}$.*
5. *$\mathcal{F}$ est fermé par transductions rationnelles fidèles.*

Démonstration. $(1 \Rightarrow 2)$ Considérons un langage non vide $M \in \mathcal{F}$ et un mot $u \in M$. Puisque $\mathcal{F}$ est fermé par quotient gauche, le langage $u^{-1} \cdot M$ appartient à $\mathcal{F}$ et celui-ci contient le mot vide.

$(2 \Rightarrow 3)$ Puisque tout langage $M \in \mathcal{F}$ peut être décrit comme $M = \tau(L)$ où la transduction τ est continue, un tel langage M ne peut contenir ε que si L contient ε.

$(3 \Rightarrow 4)$ Soit τ une transduction rationnelle fidèle et continue et $M = \tau(L)$. Si $\varepsilon \in L$, alors $M \cup \{\varepsilon\} = \tau'(L)$ avec $\tau' = \tau \cup \{(\varepsilon, \varepsilon)\}$–qui est toujours rationnelle fidèle et continue. Par conséquent $M \cup \{\varepsilon\}$ appartient aussi à $\mathcal{F}$.

$(4 \Rightarrow 5)$ Soit τ une transduction rationnelle fidèle et M un langage de $\mathcal{F}$. Nous souhaitons montrer qu'il existe un langage M' et une transduction rationnelle fidèle et continue τ' telle que $\tau(M) = \tau'(M')$. Soit X l'ensemble fini de mots u tels que $\tau(u) = \{\varepsilon\}$. La transduction τ peut être décrite comme $\tau = \tau_\varepsilon \cup \tau_c$ où $\tau_\varepsilon = \{(u, \varepsilon) \mid u \in X\}$ et $\tau_c = \tau - \tau_\varepsilon$ est le sous-ensemble continue τ. Remarquons que si $M \cap X = \emptyset$, alors $\tau(M) = \tau_c(M)$ qui appartient à $\mathcal{F}$ puisque τ_c est continue. Si par contre $M \cap X \neq \emptyset$, alors $\tau(M) = \{\varepsilon\} \cup \tau_c(M)$. Par conséquent $\tau(M)$ est équivalent à $\tau'(M \cup \{\varepsilon\})$ avec $\tau' = \tau_c \cup \{(\varepsilon, \varepsilon)\}$–qui est rationnel fidèle et continu. Le langage $\tau(M)$ appartient alors à $\mathcal{F}$.

$(5 \Rightarrow 1)$ Le quotient gauche est une transduction rationnelle fidèle. □

Lemme 8.16. *Soit $\mathcal{F}$ un cône fidèle. Si $\mathcal{F}$ est fermé par substitutions algébriques non-effaçantes, alors pour tout langage $M \in \mathcal{F}$, le langage $\Phi_{\text{bin}}(M)$ appartient à $\mathcal{F}$*

Démonstration. Il s'agit simplement de montrer que la transduction algébrique Φ_{bin} peut se décrire comme la composition d'une transduction rationnelle fidèle et continue et d'une substitution algébrique non-effaçante. Soit $\tau : A^* \longrightarrow 2^{(A\cup\{\$,\$'\})^*}$ la transduction rationnelle fidèle et continue qui associe à tout mot u l'ensemble

$$\tau(u) = \{\$u_1\$u_2\$\cdots\$u_n\$' \mid \text{ chaque } u_i \in A^*, u_1\cdots u_n = u\}.$$

Soit alors $\nu : (A \cup \{\$, \$'\})^* \longrightarrow 2^{(A\cup\{0,1,0',1'\})^*}$, la substitution algébrique non-effaçante telle que

$$\forall a \in A : \nu(a) = \{a\}; \quad \nu(\$) = \{x\bar{x} \mid x \in 01^+0 \cup 0'1'^+0'\}; \quad \nu(\$') = 0'1'^+0'.$$

Alors pour tout mot $u \in A^* : \nu(\tau(u)) = \Phi_{\text{bin}}(u)$; c'est-à-dire $\Phi_{\text{bin}} = \nu \circ \tau$. □

Lemme 8.17. *Soit $\mathcal{F}$ un cône fidèle. Si $\mathcal{F}$ est fermé par substitutions non-effaçantes et contient le langage $\{x\bar{x} \mid x \in 01^+0 \cup 0'1'^+0'\}$, alors pour tout langage $M \in \mathcal{F}$, le langage $\Phi_{\text{bin}}(M)$ appartient à $\mathcal{F}$.*

Démonstration. Puisque tout cône fidèle contient la famille des réguliers, le langage $0'1'^+0'$ appartient à $\mathcal{F}$ ainsi que tout singleton $\{a\}$. Si de plus le langage $\{x\bar{x} \mid x \in 01^+0 \cup 0'1'^+0'\}$ appartient à $\mathcal{F}$, alors la substitution ν de la preuve du lemme 8.16 est une $\mathcal{F}$-substitution non-effaçante, *ipso facto* le lemme ci-dessus. □

8.5 Application aux langages de temps quasi-réel

Dans cette section, nous considérons pour tout $k \geq 1$, la famille q-LANG^k constituée de langages de LANG^k reconnus par des machines en temps quasi-réel : celles-ci n'ont pas le droit de faire des ε-transitions (mais chaque transition effectue plusieurs instructions sur la structure de mémoire). Nous montrons que chacune de ces familles q-LANG^k possède un langage « le plus dur ». Il nous suffira de montrer que ces famille sont des cônes fidèles principaux fermés par substitution.

Définition 8.18. *Un automate à k-pile $(Q, A, q_0, \boldsymbol{\Gamma}, \Delta, F)$ (voir Définition 6.12) est dit (de temps) quasi-réel si $\Delta \subseteq Q \times A \times \widehat{\Gamma}^*_{1,k} \times Q$.*

Pour tout $k \geq 1$, soit q-LANG^k la famille de langages reconnaissables par automates à k-pile en temps quasi-réel.

Remarque 8.19. *La famille q-LANG^0 est celle des langages réguliers, et q-LANG^1 celle des langages algébriques.*

Note. Dans [GG72], la famille $\boldsymbol{Q}^+$ est introduite comme la famille de langages reconnaissables par des machines de Turing quasi-réelles. Dans [BG70], il est montré que $\boldsymbol{Q}^+$ correspond à la famille de langages reconnaissables en temps linéaire non-déterministe.

Lemme 8.20. *Pour tout $k \geq 1$, et pour toute séquence d'alphabet $\boldsymbol{\Gamma}$ de taille k, le langage $\mathcal{M}^k_{\boldsymbol{\Gamma}} \in q$-LANG^k.*

Démonstration. En effet, tout langage $\mathcal{M}^k_{\boldsymbol{\Gamma}}$ est reconnu par un automate ayant un unique état à la fois initial et acceptant et ayant comme ensemble de transitions tous les (q, α, α, q) tels que $\alpha \in \widehat{\Gamma}_{1,k}$. □

Lemme 8.21. *Pour tout $k \geq 1$, q-LANG^k est un cône fidèle.*

Démonstration. Le fait que q-LANG^k soit un cône fidèle est une conséquence du fait que la classe d'automate à k-pile quasi-réels est une famille agréable d'accepteurs (voir [Gin75]) et est indépendant du type de structure de mémoire que nous utilisons. La preuve est constructive et se fait comme suit. Soit un automate à k-pile quasi-réel $\mathcal{A} = (Q, A, q_0, \boldsymbol{\Gamma}, \Delta, F)$.

Homomorphismes non-effaçants. Soit $f : A^* \longrightarrow B^*$ un homomorphisme non-effaçant. L'automate $\mathcal{A}'$ tel que $\mathscr{L}(\mathcal{A}') = f(\mathscr{L}(A))$ est obtenu en remplaçant toute transition (q, a, ω, p) par un ensemble de transition

$$(q, b_1, \omega, p_1), (p_1, b_2, \varepsilon, p_2), \ldots, (p_{n-1}, b_n, \varepsilon, p)$$

tel que pour tout $i \in \{1, \ldots, n\} : b_i \in B$ et $f(a) = b_1 b_2 \cdots b_n$.

Intersections avec des réguliers. Soit $R \subseteq A^*$ un langage régulier. Il existe alors un automate fini $\mathcal{A}_R = (Q_R, A, q_{0,R}, \Delta_R, F_R)$ reconnaissant R. L'automate $\mathcal{A}' = (Q', A, q'_0, \boldsymbol{\Gamma}, \Delta', F')$ tel que $\mathscr{L}(\mathcal{A}') = \mathscr{L}(\mathcal{A}) \cap R$ est un produit de $\mathcal{A}$ et $\mathcal{A}_R$:

— $Q' = Q \times Q_R$; $q'_0 = (q_0, q_{0,R})$; $F' = F \times F_R$;

— la relation Δ' est telle que pour tout $a \in A$, $(q_1, q_2), (p_1, p_2) \in Q'$ et $\omega \in \Gamma^*_{1,k}$:

$$((q_1, q_2), a, \omega, (p_1, p_2)) \in \Delta' \Leftrightarrow (q_1, a, \omega, p_1) \in \Delta \text{ et } (q_2, a, p_2) \in \Delta_R.$$

Homomorphismes inverses. Soit $f : B^* \longrightarrow A^*$ un homomorphisme. L'automate $\mathcal{A}'$ tel que $\mathscr{L}(\mathcal{A}') = f^{-1}(\mathscr{L}(\mathcal{A}))$ est construit comme suit : obtenu en remplaçant, pour tout $a_1 a_2 \cdots a_n \in f(B)$, tout ensemble de transitions de la forme

$$(q_1, a_1, \omega_1, q_2), (q_2, a_2, \omega_2, q_3), \cdots, (q_n, a_n, \omega_n, q_{n+1}) \in \Delta$$

par $(q_1, b, \omega_1 \omega_2 \cdots \omega_n, q_{n+1})$ tel que $f(b) = a_1 a_2 \cdots a_n$. De plus, pour tout état $q \in Q$ et symbole $b \in B$, ajouter (q, b, ε, q) si $f(b) = \varepsilon$. □

Lemme 8.22. *Pour tout $k \geq 1$, q-LANG$^k = \mathscr{C}^f(\mathcal{M}^k)$.*

Démonstration. Montrons premièrement que q-LANG$^k \subseteq \mathscr{C}^f(\mathcal{M}^k)$. Soit $\mathcal{A} = (Q, A, q_0, \boldsymbol{\Gamma}, \Delta, F)$ un automate à k-pile quasi-réel. D'après le théorème 6.2, le transducteur $\mathcal{T} = (Q, \widehat{\Gamma}_{1,k}, A, q_0, \Delta', F)$ tel que $\mathscr{L}(\mathcal{A}) = \mathcal{R}(\mathcal{T})(\mathcal{M}^k_{\boldsymbol{\Gamma}})$ est construit en remplaçant toute production $(q, a, \omega, p) \in \Delta$ par $(q, \omega|a, p)$. Puisque $\mathcal{A}$ est de temps quasi-réel, toute les transitions de Δ' sont alors de la forme $(q, \omega|a, p)$ avec $a \neq \varepsilon$. D'après le lemme 8.10, la transduction $\mathcal{R}(\mathcal{T})$ est fidèle (et continue). Combiné au fait que la classe de transductions rationnelles fidèle soit fermée composition et que pour toute séquence $\boldsymbol{\Gamma}$, le langage $\mathcal{M}^k_{\boldsymbol{\Gamma}}$ peut s'exprimer comme $g^{-1}(\mathcal{M}^k)$ pour un certain homomorphisme g, il s'en suit que $\mathscr{L}(\mathcal{A}) \in \mathscr{C}^f(\mathcal{M}^k)$.

D'après le lemme 8.20, pour tout $k \geq 1$, et toute séquence $\boldsymbol{\Gamma}$ de taille k, le langage $\mathcal{M}^k_{\boldsymbol{\Gamma}}$ appartient à q-LANG^k ; et d'après le lemme 8.21, q-LANG^k est un cône fidèle. Par conséquent, pour tout $k \geq 1$, $\mathscr{C}^f(\mathcal{M}^k) \subseteq q$-LANG^k. Il s'en suit que q-LANG$^k = \mathscr{C}^f(\mathcal{M}^k)$. □

Il ne nous reste qu'à montrer que chaque famille q-LANG^k est fermée par substitutions.

Lemme 8.23. *Pour tout $k \geq 1$, la famille q-LANG^k est fermée par substitutions non-effaçantes.*

Démonstration. Le lemme ci-dessus, lui aussi découle du fait que la classe d'automates à k-pile quasi-réel est une famille agréable d'accepteurs.

Soit $k \geq 1$, $\mathcal{A} = (Q, A, q_0, \boldsymbol{\Gamma}, \Delta, F)$ un automate à k-pile quasi-réel et $\mu : A^* \longrightarrow 2^{B^*}$ une substitution non-effaçante telle que $\mu(a) \in q\text{-LANG}^k$ pour tout $a \in A$. Nous pouvons supposer sans perte de généralité que $\mu(a) \neq \emptyset$ pour tout $a \in A$. Pour reconnaitre le langage $\mu(\mathscr{L}(\mathcal{A}))$, nous souhaitons substituer, pour tout $a \in A$, toute étape de calcul $(q, au, \omega) \underset{\mathcal{A}}{\vdash} (q', u, \omega')$ en $(q, wu', \omega) \underset{\mathcal{A}}{\vdash}^+ (q', u', \omega')$ avec $w \in \mu(a)$, $u' \in \mu(u)$.

Pour tout $a \in A, b, c \in B$, soit $L_{(a,b,c)} = b^{-1} \cdot \mu(a) \cdot c^{-1}$. Il est clair que

$$\mu(a) = \bigcup_{b,c \in B} bL_{(a,b,c)}c \cup \bigcup_{b \in B \cap \mu(a)} \{b\}.$$

Puisque la famille $q\text{-LANG}^k$ est un cône fidèle (Lemme 8.21), elle est aussi fermée par quotient. Chaque langage $L_{(a,b,c)}$ appartient donc à $q\text{-LANG}^k$.

Pour tout $a \in A, b, c \in B$, soit $\mathcal{A}_{(a,b,c)} = (Q_{(a,b,c)}, B, q_{0,(a,b,c)}, \boldsymbol{\Gamma}_{(a,b,c)}, \Delta_{(a,b,c)}, F_{(a,b,c)})$ l'automate à k-pile quasi-réel reconnaissant $L_{(a,b,c)}$. L'automate $\mathcal{A}' = (Q', B, q_0, \boldsymbol{\Gamma}', \Delta', F)$ tel que $\mathscr{L}(\mathcal{A}') = \mu(\mathscr{L}(A))$ est tel que

- $Q' = Q \cup \bigcup_{a \in A, b, c \in B} Q_{(a,b,c)}$;
- $\Gamma'_1 = \{\$\} \cup \gamma_1 \cup \bigcup_{a \in A, b, c \in B} \Gamma_{(a,b,c),1}$; pour tout $i > 1$, $\Gamma'_i = \gamma_i \cup \bigcup_{a \in A, b, c \in B} \Gamma_{(a,b,c),i}$;
- Δ' contient tous les $\Delta_{(a,b,c)}$. De plus, pour tout $a \in A$, $b, c \in B$, $d \in \mu(a) \cap B$, $q_f \in F_{(a,b,c)}$ et pour toute transition $(q, a, \omega, p) \in \Delta$, les transitions $(q, b, \omega, q_{0,(a,b,c)})$, (q_f, c, ε, p) et (q, d, ω, p) appartiennent à Δ'.

□

Théorème 8.2. *Pour tout $k \geq 1$, il existe un langage $L_0 \in q\text{-LANG}^k$ tel que pour tout langage L, L appartient à $q\text{-LANG}^k$ si et seulement si il existe un homomorphisme f tel que $L = \$^{-1} \cdot f^{-1}(L_0)$.*

Démonstration. Le langage $X = \{x\overline{x} \mid x \in 01^+0 + 0'1'^+0'\}$ appartient à $q\text{-LANG}^1$ puisque celui-ci est algébrique. Par conséquent il appartient à toute famille $q\text{-LANG}^k$ pour $k \geq 1$. D'après les lemmes 8.23, 8.17 et 8.20, pour tout $k \geq 1$, le langage $\Phi_{\text{bin}}(\mathcal{M}^k)$ appartient à $q\text{-LANG}^k$. La fermeture par quotient de $q\text{-LANG}^k$ (lemme 8.21) et le fait que $q\text{-LANG}^k = \mathscr{C}^f(\mathcal{M}^k)$ combiné au corollaire 8.14, nous obtenons le théorème ci-dessus. □

Conclusions et perspectives

Nous avons dans cette thèse proposé plusieurs caractérisations des langages d'ordres supérieurs et nous avons parfois établi des résultats plus généraux et pour lesquels les caractérisations des langages d'ordres supérieurs n'en étaient que des corollaires.

Dans le chapitre 3, nous avons introduit et étudié une classe de langages que nous appelons langages algébriques ε-sûrs. Ces langages sont générés par des grammaires algébriques pour lesquelles *localement*, le langage généré par chaque non-terminal est inclus dans le langage de Dyck bilatéral. Bien que cette famille soit fermée par la plupart des opérations qui préservent l'inclusion dans le langage de Dyck bilatéral, nous avons montré que cette contrainte locale : « $\forall X \in N : \mathscr{L}(X) \subseteq \mathcal{T}$ » n'est pas équivalente à la contrainte *globale* « $\mathscr{L}(\mathcal{G}) \subseteq \mathcal{T}$ ». Il est de plus indécidable de savoir si un langage satisfaisant la contrainte globale satisfait la contrainte locale. *Qu'obtiendrait-on si la contrainte locale était remplacée par* « $\forall X \in N : \mathscr{L}(X) \subseteq \mathcal{D}$ » *?* Obtiendrait-on de meilleures propriétés (de décidabilité) ?

Dans les chapitres 4 et 7, nous avons introduit une notion de « langages de Dyck d'ordres supérieurs ». Un langage de Dyck de niveau $k+1$ est un ensemble de mots de Dyck pour lesquels l'image par un homomorphisme ε-sûr donné appartient à un langage de Dyck de niveau k donné. Nous avons montré que chaque niveau $k \geq 1$ de la hiérarchie OI est la fermeture par transductions rationnelles des langages de Dyck du même niveau. Une conséquence de ce fait est qu'il existe pour tout $k \geq 1$ une classe de transductions que nous appelons ε-sûres telle que pour tout $l \geq 1$ la famille du niveau $k+l$ des ordres supérieurs est exactement l'ensemble d'images de langage de Dyck de niveau l par des transductions ε-sûres de niveau k. Une question qui se pose est alors la suivante : *existe-t-il un mode d'itération (sens d'itération et classe d'homomorphismes utilisés) permettant de générer, non pas la hiérarchie OI, mais la hiérarchie IO ?* Pouvons nous par exemple trouver une classe d'homomorphismes $\mathcal{H}$ telle que si nous considérons cette fois les langages de Dyck de niveau $k+1$ comme des langages de la forme $X \cap g^{-1}(\mathcal{D})$ où X est un langage de Dyck de niveau k et g un homomorphisme de $\mathcal{H}$, alors chaque niveau k de la hiérarchie IO est le cône rationnel généré par ces langages de Dyck de niveau k ?

Nous avons aussi introduit dans le chapitre 7 la notion de machines à séquences. Celles-ci sont des machines munies de plusieurs piles comme structure de mémoire, pour lesquelles la fonction d'écriture sur ces piles et spécifiée par une séquence d'homomorphismes et les familles de langages reconnus sont des cônes rationnels générés par des itérations de langages de Dyck. Nous pensons qu'il serait intéressant de s'interroger sur l'incidence de la classe d'homomorphismes utilisés, et des restrictions syntaxique sur les machines sous-jacentes sur les propriétés (de décidabilité) des familles de langages reconnus. On peut par exemple se demander *quelle*

classe d'homomorphismes préserve la décidabilité du problème du vide ?.

Dans le chapitre 8, nous avons montré que tout cône fidèle principal fermé par substitutions algébriques non-effaçantes, ou bien fermé par substitution non-effaçantes et contenant le langage $\{\varepsilon\} \cup \{x\bar{x} \mid x \in 01^+0\}$, admet une représentation par homomorphismes inverses. Une telle famille contient alors un langage L_0 pour lequel le problème du mot de tout langage de cette famille peut être réduit au problème du mot de L_0. *Peut-on trouver d'autres conditions suffisantes à cette caractérisation ?*

Dans le chapitre 5, nous avons étudié la définissabilité des langages indexés dans un fragment du second ordre existentiel. Nous avons considéré la classe des langages indexés doubles. Ceux-ci sont générés par des grammaires indexées pour lesquelles chaque production d'empilement et de dépilement est en forme double. Cette « double lexicalisation » et la notion de transductions algébriques ε-sûres nous ont permis de montrer que la classe des langages indexés doubles coïncide avec la famille de langages définissables dans un fragment du second ordre existentiel dans lequel deux types de quantifications existentielles sont autorisées et celles-ci sont restreintes à une classe de relations que nous appelons « matching de Dyck ». Les langages indexés quant à eux sont des images de langages définissables dans cette dernière classe par une transformation logique. *Qu'en est-il de certaines sous-classes des langages indexés tels que les langages indexés linéaires, m-adics [Kan14a] ? Et qu'en est-il du reste des ordres supérieurs ?*

Bibliographie

[ABB97] Jean-Michel Autebert, Jean Berstel, and Luc Boasson. Context-free languages and pushdown automata. In *Handbook of Formal Languages*, volume 1, Word Language Grammar, pages 111–174. Springer-Verlag, Berlin, 1997.

[AD77] André Arnold and Max Dauchet. Un theoreme de chomsky-schützenberger pour les forets algebriques. *CALCOLO*, 14 :161–184, Jun 1977.

[Aho67] Alfred V. Aho. *Indexed grammars, an extension of context free grammars*. PhD thesis, Princeton, 1967.

[AM04] Rajeev Alur and Parthasarathy Madhusudan. Visibly pushdown languages. In *Symposium on Theory Of Computation proceedings*, pages 202–211. ACM, 2004.

[AM09] Rajeev Alur and Parthasarathy Madhusudan. Adding nesting structure to words. *Journal of ACM*, 56(3) :16 :1–16 :43, 2009.

[B$\ddot{\text{o}}$60] Julius R. Büchi. Weak secondorder arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(16) :66–92, 1960.

[B$\ddot{\text{o}}$90] Julius R. Büchi. *On a Decision Method in Restricted Second Order Arithmetic*, pages 425–435. Springer New York, New York, NY, 1990.

[BB00] Jean Berstel and Luc Boasson. Xml grammars. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science*, pages 182–191, London, UK, UK, 2000. Springer-Verlag.

[BB02] Jean Berstel and Luc Boasson. *Balanced Grammars and Their Languages*, pages 3–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[BCR09] Alberto Bertoni, Christian Choffrut, and Roberto Radicioni. *The Inclusion Problem of Context-Free Languages : Some Tractable Cases*, pages 103–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[Ber79] Jean Berstel. *Transductions and Context-Free Languages*. Teubner, Stuttgart, 1979.

[Ber92] Jean Berstel. *Axel Thue's papers on repetitions in words : a translation*. 1992.

[BG70] Ronald V. Book and Sheila A. Greibach. Quasi-realtime languages. *Mathematical systems theory*, 4(1) :97–111, Mar 1970.

[BHPS60] Yehoshua Bar-Hillel, Micha A. Perles, and Eli Shamir. On formal properties of simple phrase structure grammars. Technical Report 4, Office of Naval Research, Information Systems Branch, 1960.

[BN73] Luc Boasson and Maurice Nivat. Sur diverses familles de langages fermées par transduction rationnelle. *Acta Informatica*, 2(2) :180–188, 1973.

[BV98] Hans-Jörg Burtschichk and Heribert Vollmer. Lindström quantifiers and leaf language definability. *International Journal of Foundations of Computer Science*, 09(03) :277–294, 1998.

[Car05] Arnaud Carayol. *Regular Sets of Higher-Order Pushdown Stacks*, pages 168–179. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[Cho56] Noam Chomsky. Three models for the description of language. *IRE Trans. Info. Theory*, 1 :113–124, 1956.

[Chu37] Alonzo Church. Review : A. m. turing, on computable numbers, with an application to the entscheidungsproblem. *Journal of Symbolic Logic*, 2(1) :42–43, 03 1937.

[CS63] Noam Chomsky and Marcel Paul Schützenberger. The algebraic theory of context-free languages. In *Computer Programming and Formal Systems*, pages 118–161, Amsterdam, 1963. North-Holland.

[Dam82] Werner Damm. The IO- and OI-hierarchies. *TCS : Theoretical Computer Science*, 20, 1982.

[DG86] Werner Damm and Andreas Goerdt. An automata-theoretical characterization of the oi-hierarchy. *Information and Control*, 71(1) :1 – 32, 1986.

[Don70] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4(5) :406 – 451, 1970.

[DPS79] Jürgen Duske, Rainer Parchmann, and Johann Specht. A homomorphic characterization of indexed languages. *Elektronische Informationsverarbeitung und Kybernetik*, 15(4) :187–195, 1979.

[Elg61] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1) :21–51, 1961.

[Eng83] Joost Engelfriet. Iterated pushdown automata and complexity classes. In *STOC : ACM Symposium on Theory of Computing (STOC)*, 1983.

[Fis68] Michael J. Fischer. Grammars with macro-like productions. In *9th Annual Symposium on Switching and Automata Theory*, pages 131–142. IEEE Computer Society, 1968.

[Fra05] Séverine Fratani. *Automates à piles de piles ... de piles*. PhD thesis, Université Bordeaux 1, 2005.

[FS06] Séverine Fratani and Géraud Sénizergues. Iterated pushdown automata and sequences of rational numbers. *Annals of Pure and Applied Logic*, 141(3) :363 – 411, 2006.

[Gaz88] Gerald Gazdar. Applicability of indexed grammars to natural languages. In *Natural Language Parsing and Linguistic Theories*, pages 69–94. D. Reidel, Dordrecht, 1988.

[GF80] Sheila A. Greibach and Emily P. Friedman. Superdeterministic pdas : A subcase with a decidable inclusion problem. *Journal of ACM*, 27(4) :675–700, 1980.

[GG67] Seymour Ginsburg and Sheila A. Greibach. Abstract families of languages. In *Proceedings of the 8th Annual Symposium on Switching and Automata Theory (SWAT 1967)*, pages 128–139, Washington, DC, USA, 1967. IEEE Computer Society.

[GG72] Seymour Ginsburg and Sheila A. Greibach. Multitape afa. *Journal of ACM*, 19(2) :193–221, April 1972.

[GH67] Seymour Ginsburg and Michael A. Harrison. Bracketed context-free languages. *Journal of Compututation and System Sciences*, 1(1) :1–23, April 1967.

[Gin75] Seymour Ginsburg. *Algebraic and Automata-Theoretic Properties of Formal Languages*. Elsevier Science Inc., New York, NY, USA, 1975.

[Gre67] Sheila A. Greibach. A note on pushdown store automata and regular systems. *Proceedings of the American Mathematical Society*, 18(2) :263–268, 1967.

[Gre68] Sheila A. Greibach. A note on undecidable properties of formal languages. *Mathematical systems theory*, 2(1) :1–6, Mar 1968.

[Gre73] Sheila A. Greibach. The hardest context-free language. *SIAM Journal of Computing*, 2(4), December 1973.

[GS80] Haim Gaifman and Eli Shamir. Roots of the hardest context-free language and other constructs. *sigact*, 12 :45–51, Fall 1980.

[Gue83] Inène Guessarian. Pushdown tree automata. *Mathematical systems theory*, 16(1) :237–263, Dec 1983.

[HN81] Sadaki Hirose and Masakazu Nasu. Left universal context-free grammars and homomorphic characterizations of languages. *Information and Control*, 50(2) :110 – 118, 1981.

[Hop69] John E. Hopcroft. On the equivalence and containment problems for context-free languages. *Mathematical Systems Theory*, 3(2) :119–124, 1969.

[HOY85] Sadaki Hirose, Satoshi Okawa, and Masaaki Yoneda. A homomorphic characterization of recursively enumerable languages. *Theoretical Computer Science*, 35 :261 – 269, 1985.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[Kan14a] Makoto Kanazawa. *A Generalization of Linear Indexed Grammars Equivalent to Simple Context-Free Tree Grammars*, pages 86–103. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[Kan14b] Makoto Kanazawa. Multidimensional trees and a Chomsky–Schützenberger–Weir representation theorem for simple context-free tree grammars. *Journal of Logic and Computation*, 2014.

[Kle56] Stephen C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1956.

[Knu67] Donald E. Knuth. A characterization of parenthesis languages. *Information and Control*, 11(3) :269 – 289, 1967.

[Lan06] Tore Langholm. A descriptive characterisation of linear languages. *Journal of Logic, Language and Information*, 15(3) :233–250, October 2006.

[Leg80] Jeannine Leguy. *Transductions rationnelles décroissantes et substitution*. PhD thesis, Université de Lille, 1980.

[Lin66] Per Lindström. First order predicate logic with generalized quantifiers. *Theoria*, 32(3) :186–195, 1966.

[LMSV99] Clemens Lautemann, Pierre McKenzie, Thomas Schwentick, and Heribert Vollmer. The descriptive complexity approach to logcfl. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, STACS'99, pages 444–454, Berlin, Heidelberg, 1999. Springer-Verlag.

[LST94] Clemens Lautemann, Thomas Schwentick, and Denis Thérien. Logics for context-free languages. In *Computer Science Logic*, pages 205–216, 1994.

[Mas74] A. N. Maslov. Hierarchy of indexed languages of arbitrary level. *Soviet Math. Dokl*, 115(14) :1170–1174, 1974.

[McN67] Robert McNaughton. Parenthesis grammars. *Journal of ACM*, 14(3) :490–500, July 1967.

[MW67] J. Mezei and J.B. Wright. Algebraic automata and context-free sets. *Information and Control*, 11(1) :3 – 29, 1967.

[Niv68] Maurice Nivat. Transductions des langages de chomsky. *Ann. de l'Inst. Fourier*, 18 :339–456, 1968. in french.

[Okh12] Alexander Okhotin. Non-erasing variants of the chomsky-schützenberger theorem. In *Developments in Language Theory*, volume 7410 of *Lecture Notes in Comput. Sci.*, pages 121–129. Springer, 2012.

[Okh16] Alexander Okhotin. *The Hardest Language for Conjunctive Grammars*, pages 340–351. Springer International Publishing, Cham, 2016.

[PD90] Rainer Parchmann and Jürgen Duske. The structure of index sets and reduced indexed grammars. *ITA*, 24 :89–108, 1990.

[Pig16] Giovanni Pighizzini. Restricted turing machines and language recognition. In *Language and Automata Theory and Applications - 10th International Conference, LATA 2016, Prague, Czech Republic, March 14-18, 2016, Proceedings*, volume 9618 of *Lecture Notes in Computer Science*, pages 42–56. Springer, 2016.

[Pin83] Jean-Eric Pin. Introduction aux langages reconnaissables. In *Actes des Journées d'Avignon d'Informatique Théorique*, 1983.

[Pos36] Emil L. Post. Finite combinatory processes—formulation. *Journal of Symbolic Logic*, 1(3) :103–105, 1936.

[Sor14] Alexey Sorokin. *Monoid Automata for Displacement Context-Free Languages*, pages 154–173. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[Sto] Julia Stoll. The hardest context-free language revised.

[Tra61] Boris Trakhtenbrot. Finite automata and the logic of monadic predicates. In *Doklady Akademii Nauk SSSR 140*, pages 326–329, 1961.

[Tur78] Paavo Turakainen. On characterization of recursively enumerable languages in terms of linear languages and vw-grammars. *Indagationes Mathematicae (Proceedings)*, 81(1) :145 – 153, 1978.

[TW68] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical systems theory*, 2(1) :57–81, 1968.

[VDH16] Heiko Vogler, Manfred Droste, and Luisa Herrmann. *A Weighted MSO Logic with Storage Behaviour and Its Büchi-Elgot-Trakhtenbrot Theorem*, pages 127–139. Springer International Publishing, Cham, 2016.

[Vog88] Heiko Vogler. The oi-hierarchy is closed under control. *Information and Computation*, 78(3) :187 – 204, 1988.

[Wei88] David J. Weir. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, University of Pennsylvania, 1988.

[YW84] Takashi Yokomori and Derick Wood. An inverse homomorphic characterization of full principal afl. *Information Sciences*, 33(3) :209 – 215, 1984.

[ZKL17] Georg Zetzsche, Dietrich Kuske, and Markus Lohrey. On boolean closed full trios and rational kripke frames. *Theory of Computing Systems*, 60(3) :438–472, Apr 2017.