

# THÈSE

PRÉSENTÉE À

**L'UNIVERSITÉ BORDEAUX I**

ÉCOLE DOCTORALE DE MATHÉMATIQUES et  
INFORMATIQUE

Par **David JANIN**

POUR OBTENIR LE GRADE DE

**DOCTEUR**

SPÉCIALITÉ : INFORMATIQUE

---

**Propriétés logiques du non déterminisme et  $\mu$ -calcul modal**

---

**Soutenu le :** 5 Janvier 1996

**Après avis de :** MM. P. Darondeau Rapporteurs  
D. Niwiński

**Devant la Commission d'examen formée de :**

A. Arnold	Professeur d'Université	.....	Examineur
B. Courcelle	Professeur d'Université	.....	Examineur
P. Darondeau	Directeur de Recherche	.....	Rapporteur
M. Dauchet	Professeur d'Université	.....	Examineur
J.-E. Pin	Directeur de Recherche	.....	Examineur
D. Niwiński	Professeur d'Université	.....	Rapporteur



*Notre première rencontre eut lieu presque par hasard, sous la neige, à Montréal, au début de l'année 1992. A l'image du Mont-Royal, recouvert, le sujet de thèse qui me lia à André Arnold fut avant tout une atmosphère : les systèmes informatiques, et une matière : la logique.*

*Cette "impression" initiale aurait pu tout aussi bien se transformer en labyrinthe cauchemardesque. C'était sans compter son enthousiasme et son expérience; c'était sans compter aussi le contexte de travail que m'offrait mois après mois le LaBRI; c'était sans compter enfin la surprise de me voir partager avec Igor Walukiewicz, de l'Université de Varsovie, une conjecture que notre rencontre et notre collaboration, qui se poursuit aujourd'hui encore, transforma en théorème.*

*Aujourd'hui, un premier travail se termine. Sa réalisation fut parfois difficile. Aux hasards des couloirs du laboratoire et des halls de conférences, des personnes m'ont offert généreusement réconforts et conseils. Ma thèse leur doit beaucoup, qu'ils en soient ici remerciés.*

*Je tiens en particulier à exprimer toute ma gratitude à André Arnold. Il m'initia aux mystères des systèmes de transitions et me fit partager sa passion pour le "théorème de Rabin". Il fut aussi le lecteur efficace, rapide et critique de mes nombreuses "ébauches de thèses" dont la dernière est présentée ici.*

*Bruno Courcelle, par ses travaux et le groupe de travail "graphes et logiques", exerça une influence profonde sur mon travail. C'est avec plaisir et honneur que je le compte parmi les membres de mon jury.*

*C'est avec une appréhension bien naturelle que j'envoyais une première version de ce mémoire à Philippe Darondeau et Damian Niwiński. Les commentaires qu'ils m'envoyèrent, les critiques et les remarques qu'ils me firent, leur propre vision de mon travail, furent autant d'aides précieuses pour la rédaction finale de ce mémoire. Je ne saurais trop les en remercier!*

*Pour les esprits romanesques, un jury de thèse en informatique théorique en France ne saurait être complet sans représentants (extérieurs au LaBRI!) de l'"Ecole Française d'Informatique Théorique" dont Boris Vian citait le grand-père fondateur. Je remercie Jean-Eric Pin et Max Dauchet, qui ont accepté de faire partie de mon jury, de remplir ce rôle. C'est avec plaisir et honneur que je leur présente mes travaux sur les arbres, les automates et la logique.*

*Vivant sans un souffle,  
Froid comme la mort,  
Jamais assoiffé, toujours buvant,  
En cotte de mailles, jamais cliquetant.*

J.R.R. Tolkien, *Bilbo le hobbit*.

*A mon épouse.*

# Résumé

Cette thèse traite de la spécification logique des comportements de programmes. Les programmes étant modélisés à l'aide de systèmes de transitions, les comportements de programmes sont définis comme des classes d'équivalence de systèmes de transitions, l'équivalence considérée étant l'équivalence de bisimulation de Park. Dans ce cadre, une formule logique spécifie une propriété de comportements lorsqu'elle admet une classe de modèles close par équivalence de bisimulation.

Nous démontrons qu'une formule de la logique monadique du second ordre est comportementale au sens précédent si et seulement si elle est équivalente à une formule du  $\mu$ -calcul modal de Kozen. Ce résultat donne un sens précis à l'assertion "la plupart des logiques de programmes peuvent être traduites en  $\mu$ -calcul modal".

Techniquement, notre approche développe la théorie des automates sur les arbres finis ou infinis, mettant en évidence le rôle unificateur que jouent les calculs de points-fixes vis à vis de la théorie des modèles d'une part et de la théorie des automates d'autre part.

# Abstract

This thesis deals with logical specifications of program's behaviors. Modeling programs with transition systems, programs behaviors are defined as equivalent classes of transition systems under Park's bisimulation equivalence. In this setting, formulas describe properties of behaviors when their class of models are closed under bisimulation equivalence.

We prove that a formula of monadic second order logic is behavioral in the previous sense if and only if it is equivalent to a formula of Kozen's modal  $\mu$ -calculus. Such a result gives a precise meaning to the statement that most logic of programs can be translated into the modal  $\mu$ -calculus.

Technically, our approach developed automata theory on finite or infinite trees. It illustrates the fundamental rôle of fixed-point calculi in between model theory on the one hand and automata theory on the other hand.



# Introduction

*Il était environ neuf heures du matin; c'était à la fin de novembre, par un temps de dégel. Le train de Varsovie filait à toute vapeur vers Pétersbourg.*  
Dostoïevski, *L'Idiot*.

On s'intéresse ici [26] aux aspects de la théorie des modèles pouvant être utilisés dans les méthodes formelles de vérification et validation de programmes informatiques.

Cette étude, qui vise donc à décrire en logique les comportements des programmes, soulève immédiatement deux problèmes: il faut tout d'abord disposer d'un *modèle* mathématique de la notion de programmes puis comprendre quel *modèle* de comportements est induit par cette notion. Cette distinction est d'autant plus cruciale qu'en général les programmes sont finis alors que les comportements, souvent, ne le sont pas.

Un modèle de programmes, aussi vieux que l'informatique mais dont la pertinence, comme l'illustre Arnold [3], semble loin d'être épuisée, est le modèle des *systèmes de transitions*.

Intuitivement, un système de transitions se compose d'un ensemble d'*états*: représentant les diverses images mémoires possibles du programme, d'un état initial et d'un ensemble d'arcs entre ces états, appelées *transitions*, représentant les effets possibles de l'exécution d'une action à partir de tel ou tel état du programme.

Théoriquement, il est possible de modéliser tout ordinateur à l'aide d'un système de transitions. En pratique cependant, une telle modélisation est irréalisable. Le nombre d'états nécessaire à une telle modélisation est en effet exponentiel en la taille de la mémoire de l'ordinateur!

Un premier remède face à cette explosion combinatoire consiste à ne conserver que certains aspects de l'image mémoire du programme à modéliser. Autrement dit, on peut faire *abstraction* d'une partie de l'information contenue dans la mémoire de l'ordinateur à un instant donné.

En général, cette approche introduit du non-déterminisme, c'est-à-dire, à partir d'un

certain état du système, la possibilité d'exécuter de plusieurs façons une même action. En effet, même si l'ordinateur sous-jacent est déterministe, l'information partielle que modélise un état peut être insuffisante pour être déterminante.

Ayant donc retenu les systèmes de transitions comme modèles de programmes, la notion, sous-jacente, de comportement reste à définir. Plus précisément: étant donné un système de transitions  $S$ , quelle notion de comportement  $C(S)$  doit-on associer au programme modélisé par  $S$ ?

En premier lieu, on souhaite ne pas tenir compte de l'éventuelle réutilisation de la mémoire. Or dans les systèmes de transitions, chaque circuit modélise précisément une telle réutilisation. La notion de *développement* d'un système, c'est à dire l'*arbre de transitions* obtenu en construisant les chemins issus de l'état initial, s'impose ici comme premier modèle de comportements.

En second lieu, la présence de non-déterminisme peut conduire à ne pas faire de différence entre deux états à partir desquels les exécutions des mêmes actions conduisent à des états décrivant les mêmes comportements. Dans cette approche, dupliquer les sous-arbres du développement d'un système de transitions ne doit pas modifier le comportement modélisé. La notion d' $\alpha$ -*expansion* d'un système, c'est à dire l'*arbre de transitions* obtenu en dupliquant  $\alpha$  fois tout sous-arbres du développement de ce système, s'impose ici comme second modèle de comportements. La figure 1 illustre les notions de développement et expansion d'un système.

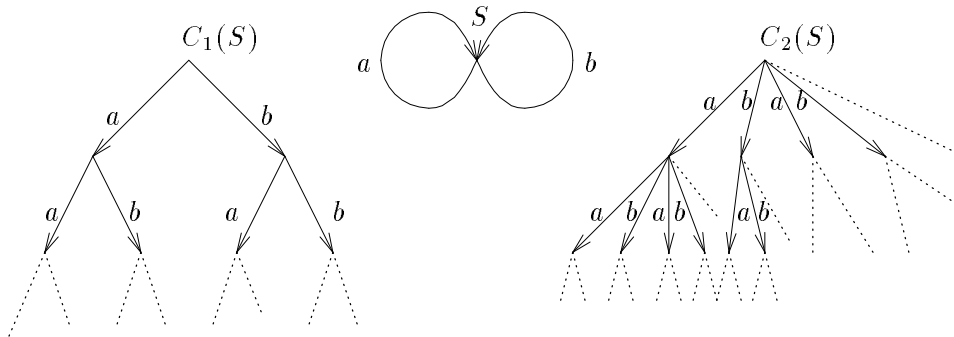


FIG. 1 - Développement  $C_1(S)$  et expansion  $C_2(S)$  d'un système  $S$ .

Les travaux de Castellani [9] montre que cette notion caractérise l'équivalence de *bisimulation* de Park [38] au sens où, pour tout système de transition  $S_1$  et  $S_2$  de cardinalité au plus  $\alpha$ , l'existence d'une relation de bisimulation entre les états initiaux de  $S_1$  et  $S_2$



équivalent à l'existence d'une  $\alpha$ -expansion commune aux systèmes  $S_1$  et  $S_2$ . Autrement dit, la notion d' $\alpha$ -expansion d'un système permet d'associer à tout système de cardinalité au plus  $\alpha$  un représentant canonique et arborescent de sa classe d'équivalence par bisimulation.

On dispose alors d'un modèle de programmes et d'un modèle de comportements associés. La spécification comportementale d'un programme est alors définie comme une propriété logique de comportement.

Un candidat naturel de logique pour ces spécifications comportementales est la logique monadique du second ordre (*LMSO*).

Elle est en effet suffisamment *expressive* pour spécifier des comportements limites, à l'infini, tels que ceux traitant du problème de l'équité. Elle est aussi suffisamment restreinte pour rester *décidable*: étant donnée une spécification de comportement exprimée en logique monadique du second ordre, on est capable de tester en un temps fini la cohérence de cette spécification, c'est-à-dire savoir si elle admet au moins un modèle.

Cependant les modèles de comportements peuvent être infinis. La construction explicite d'un modèle d'une spécification comportementale est donc exclue. Au mieux, peut-on espérer construire un système de transitions dont le comportement associé soit effectivement un modèle de la spécification.

D'un point de vue logique, cela soulève le problème suivant : étant donnée une spécification comportementale  $\varphi$ , existe-t-il une formule  $C^{-1}(\varphi)$  telle que, pour tout comportement  $C(S)$  de système de transitions  $S$ , affirmer que  $C(S)$  est un modèle de  $\varphi$  équivaut à dire que  $S$  est un modèle de  $C^{-1}(\varphi)$ ? Ce problème a été résolu par Courcelle et Walukiewicz [14] lorsque  $C(S)$  est définie comme le développement de  $S$ .

Nous traitons dans ce mémoire le cas où le comportement d'un système est l' $\omega$ -expansion de ce système.

L'outil logique qui s'impose dans ce cas là est le  $\mu$ -calcul modal, un calcul de point fixe définie par Kozen [30].

Du point de vue comportemental, ce calcul a en effet la propriété remarquable que pour toute formule  $\varphi$ , tout système de transitions  $S$ , en notant à nouveau  $C(S)$  l' $\alpha$ -expansion de  $S$ , affirmer que  $C(S)$  est un modèle de  $\varphi$  équivaut à affirmer que  $S$  est un modèle de  $\varphi$ . Autrement dit, la sémantique de ces formules est close par équivalence de bisimulation.

Nous démontrons dans ce mémoire non seulement l'existence, pour toute formule  $\varphi$  de la logique monadique du second ordre, d'une formule  $C^{-1}(\varphi)$  satisfaisant les propriétés ci-dessus, mais nous démontrons aussi que l'ensemble des formules logiques décrit par  $C^{-1}(\varphi)$  lorsque  $\varphi$  décrit l'ensemble des formules de logique monadique du second ordre

est équivalent, d'un point de vue sémantique, à l'ensemble des formules du  $\mu$ -calcul modal.

\* \* \*

Ce mémoire est composé de cinq chapitres sur le thème des propriétés logiques des comportements non déterministes. Deux annexes le complètent afin de présenter : dans la première le vocabulaire et les notations logiques utilisés, dans la seconde une présentation logique de la théorie des domaines, qui rappelle celle de Palmgren et Stoltenberg-Hansen [37], dont le contenu diffère trop du thème principal de ce mémoire pour pouvoir constituer un chapitre à part entière. Un index, placé à la fin du mémoire, doit permettre de retrouver à tout moment toutes les définitions utilisées.

Le contenu respectif de chaque chapitre est décrit ci-dessous.

Le premier chapitre traite des propriétés élémentaires du non-déterminisme. On s'intéresse plus précisément au cas où le non-déterminisme est modélisé à l'aide de structures plates, c'est à dire des ensembles amorphes dont les points représentent les diverses éventualités, et on caractérise le pouvoir expressif de la logique du premier ordre et la logique monadique du second ordre sur de telles structures.

L'intuition sous-jacente à la notion de non déterminisme conduit à s'intéresser plus particulièrement à la sémantique des formules du premier ordre *sans prédicat d'égalité*. On introduit aussi la notion de type modal d'une formule ou d'une structure. Elle nous conduit à une première définition de logiques modales caractérisant les formules sans égalité. Elle nous conduit aussi à une première définition de la notion d'équivalence de bisimulation.

La fin de ce chapitre décrit les relations entre cette logique modale et la théorie des "power-domains". Elle s'appuie sur notre présentation logique des domaines. Nous exprimons les propriétés des ensembles de points à l'aide d'ensembles de propriétés de points. Plusieurs sémantiques associées à ces ensembles de propriétés nous permettent de décrire plusieurs types de non-déterminisme : le non déterminisme angélique, le non déterminisme démoniaque, . . . Ce travail généralise et précise les travaux d'Abramsky [1, 2], de Robinson [41] et Winskel [49].

Le second chapitre traite des propriétés algébriques et logiques des systèmes de transitions en tant que modèles de programmes non-déterministes.

On met en évidence la distinction entre modèles de programmes ou de machines, et modèles de comportements via la notion d' $\alpha$ -expansion d'un système. Cette notion, qui

s'appuie sur les “abstraction homomorphisms” de Castellani [9, 6], conduit à la notion d'équivalence de bisimulation. On montre que cette équivalence de bisimulation est naturellement caractérisée, dans le cas des arbres de transitions à branchements finis, par les formules de logique du premier ordre sans prédicat d'égalité.

On rappelle ensuite la définition de la logique modale paramétrée [23], un sous-ensemble sémantique de la logique du premier ordre sans prédicats d'égalité. Le théorème de Van Benthem [7], nous montre en fait que l'ensemble des formules de logique du premier ordre dont la sémantique est close par équivalence de bisimulation est équivalent à la logique modale paramétrée.

L'étude faite dans le cas des structures plates est étendue au cas des systèmes de transitions à la fin du chapitre. On prouve un théorème d'existence de forme normale disjonctive pour toute formule de la logique modale paramétrée. Cette notion de forme normale conduit à la définition d'un nouveau combinateur modal qui se substitue avantageusement aux modalités paramétrées  $\langle a \rangle$  et  $[a]$ .

Le  $\mu$ -calcul modal, une extension de la logique modale paramétrée à l'aide de plus petits et plus grands points fixes, est présenté dans le troisième chapitre.

On présente en détail la notion de calcul de points fixes par approximations successives qui, avec la notion de signature [43], se révèle être un outil puissant d'étude de la sémantique des calculs de points fixes.

La notion de tableau pour le  $\mu$ -calcul modal, portée à notre connaissance par Walukiewicz, nous permet de caractériser de façon calculatoire la sémantique des formules du  $\mu$ -calcul. On montre en particulier que l'alternance de plus petit et plus grand point fixe se traduit naturellement en un critère infinitaire similaire aux critères de parité, appelé aussi condition de chaîne [31], sur les automates de mots ou d'arbres binaires infinis [44].

On caractérise enfin la sémantique du  $\mu$ -calcul modal à l'aide d'une notion originale d'automates sur les systèmes de transitions appelés automates modaux alternants. Elle apparaît comme une extension aux cas des systèmes de transitions des automates alternants sur arbres binaires [34]. Le combinateur modal défini au chapitre précédent permet ici de capturer la bonne notion de transition.

En l'absence de point fixe, on a montré que l'usage de la conjonction peut être largement restreint sans diminuer le pouvoir expressif du langage logique. Le quatrième chapitre étend ce résultat au  $\mu$ -calcul modal.

D'un point de vue logique, on définit la notion de forme normale disjonctive pour les

formules du  $\mu$ -calcul modal. On montre que la satisfaisabilité de ces formules est décidable en temps linéaire. Dans le cas général, le problème de la satisfaisabilité d'une formule du  $\mu$ -calcul est *EXP*-space complet. On démontre alors de façon constructive que toute formule du  $\mu$ -calcul modal est équivalente à une formule sous forme normale disjonctive. Cette notion apparaît en fait comme une généralisation aux cas des systèmes de transitions des formes normales de formules avec points fixes mises en évidence par Niwiński pour caractériser le pouvoir d'expression des automates d'arbres binaires.

En terme d'automates ce théorème d'existence de forme normale disjonctive s'interprète comme un théorème de simulation généralisant celui de Muller et Schupp [35] : il apparaît que tout automate modal alternant peut en effet être simulé par un automate modal sans alternances. Le théorème de complémentation de Rabin [40], réputé difficile [20, 33, 34, 17, 5], est une conséquence immédiate de ce résultat.

Ces résultats ont fait l'objet d'une publication commune avec Walukiewicz [27].

Le dernier chapitre traite du pouvoir expressif de la logique monadique du second ordre par rapport à celui du  $\mu$ -calcul modal.

Le début de ce chapitre introduit la notion de lexique : des arbres dont les fils de chaque sommet sont ordonnés de telle sorte qu'ils forment un mot fini ou infini. Un codage de ces lexiques en arbres binaires nous permet d'extraire des automates d'arbres binaires une notion d'automate sur les lexiques caractérisant le pouvoir d'expression de la logique monadique du second ordre.

De cette notion d'automate sur les lexiques, on extrait une notion d'automate sur les arbres de transitions dont le pouvoir d'expression est caractérisable par un sur-ensemble de la logique monadique.

Ce dernier résultat nous permet alors de démontrer, à la fin du chapitre, que toute formule de la logique monadique du second ordre sur les systèmes de transitions dont la sémantique est close par équivalence de bisimulation est équivalente à une formule de  $\mu$ -calcul modal.

Autrement dit, on démontre que le  $\mu$ -calcul modal est le plus grand ensemble de formules "préservant" l'équivalence de bisimulation dont la sémantique est définissable en logique monadique du second ordre.

# Table des matières

<b>1 Propriétés élémentaires du non-déterminisme</b>	<b>1</b>
1.1 Logique égalitaire et structures plates . . . . .	2
1.1.1 Logique du premier ordre sur structures plates . . . . .	2
1.1.2 Logique monadique du second ordre . . . . .	5
1.2 Le cas sans égalité . . . . .	6
1.2.1 Formules modales . . . . .	6
1.2.2 Caractérisation modale des formules sans égalité . . . . .	7
1.3 Propriétés d'ensembles et ensembles de propriétés . . . . .	8
1.3.1 Un vocabulaire pour les ensembles de structures . . . . .	9
1.3.2 Trois sémantiques pour les ensembles de propriétés . . . . .	11
1.3.3 Connexions avec les constructions classiques . . . . .	13
<b>2 Systèmes de transitions, bisimulation et logique</b>	<b>17</b>
2.1 Définitions et propriétés élémentaires des systèmes de transitions . . . . .	17
2.1.1 Systèmes de transitions . . . . .	18
2.1.2 Développement d'un système et arbres de transitions . . . . .	19
2.1.3 Expansion d'un système, simulation et bisimulation . . . . .	21
2.2 Caractérisation logique de la bisimulation . . . . .	25
2.2.1 Logique du premier ordre sans égalité . . . . .	25
2.2.2 Logique modale paramétrée . . . . .	27
2.3 Forme normale disjonctive pour <i>LMod</i> . . . . .	29
2.3.1 Un connecteur pour deux modalités . . . . .	29
2.3.2 Formules modales disjonctives . . . . .	30
<b>3 Un calcul de points fixes</b>	<b>33</b>
3.1 Le $\mu$ -calcul modal . . . . .	34
3.1.1 Un calcul de points fixes . . . . .	34

3.1.2	Formules bien nommées . . . . .	37
3.1.3	Calcul de point fixe par approximations . . . . .	38
3.1.4	Signatures . . . . .	40
3.2	Une sémantique calculatoire pour le $\mu$ -calcul . . . . .	42
3.2.1	Tableaux . . . . .	42
3.2.2	Sémantique des tableaux . . . . .	45
3.3	Automates modaux alternants . . . . .	50
3.3.1	Définition des automates modaux alternants . . . . .	50
3.3.2	Traces et développement d'un automate . . . . .	51
3.3.3	Formes normales arborescentes . . . . .	53
3.3.4	Automates alternants et $\mu$ -formules modales . . . . .	55
<b>4</b>	<b>Formes normales disjonctives</b>	<b>61</b>
4.1	Formes normales disjonctives . . . . .	62
4.1.1	Définitions et propriétés élémentaires . . . . .	62
4.1.2	Une équivalence de tableaux . . . . .	65
4.1.3	Existence de forme normale disjonctive . . . . .	66
4.2	Automates modaux . . . . .	70
4.2.1	Définitions des automates . . . . .	70
4.2.2	Automates modaux et $\mu$ -formules modales . . . . .	72
4.3	Restrictions au cas des arbres binaires infinis . . . . .	74
4.3.1	Définitions et résultats classiques sur les arbres binaires . . . . .	74
4.3.2	Une preuve du théorème de complémentation . . . . .	75
<b>5</b>	<b>Complétude expressive du mu-calcul</b>	<b>77</b>
5.1	Présentation lexicographique des arbres . . . . .	78
5.1.1	Lexiques et arbres de transitions induits . . . . .	78
5.1.2	Codage binaire des lexiques . . . . .	80
5.1.3	Automates sur les lexiques . . . . .	82
5.2	Propriétés induites . . . . .	88
5.2.1	Propriétés d'arbres "modulo" une présentation lexicographique . . . . .	88
5.2.2	Extension du théorème de Parikh aux mots infinis . . . . .	89
5.2.3	Caractérisation par automates . . . . .	90
5.3	Applications . . . . .	92
5.3.1	$\Omega$ -clôture d'une formule . . . . .	92
5.3.2	Caractérisation du pouvoir expressif du $\mu$ -calcul modal . . . . .	97

<b>Annexe</b>	<b>103</b>
<b>A Structures, formules et jeux logiques</b>	<b>103</b>
A.1 Structures, vocabulaires et formules logique . . . . .	103
A.2 Formules du premier ordre, égalité et jeux . . . . .	105
<b>B Présentation logique des domaines</b>	<b>109</b>
B.1 Définitions usuelles des domaines sémantiques . . . . .	110
B.1.1 Complétudes et idéaux . . . . .	110
B.1.2 Compacité et domaine . . . . .	111
B.1.3 “Power-domains” . . . . .	113
B.2 Construction logique des domaines . . . . .	114
B.2.1 Définition . . . . .	114
B.2.2 Domaines de théories et domaines de modèles . . . . .	117
B.2.3 Connexion avec l’approche de Palmgren et Stoltenberg-Hansen . . . . .	120





# Chapitre 1

## Propriétés élémentaires du non-déterminisme

Que ce soit une conditionnelle dans un langage tel que C ou Pascal, ou bien une alternative de commandes gardées dans un langage tel que CSP ou Prolog, le non-déterminisme, au sens le plus large, est omniprésent dans les langages de programmation. Dans l'étude de la sémantique de ces langages, on représente souvent la sémantique de telles structures à l'aide d'ensembles dont chaque élément décrit la sémantique d'une des composantes de cette structure : par exemple une commande gardée.

D'un point de vue logique, on est amené à étudier des structures, dites plates, de la forme  $\langle E, \rho \rangle$  pour  $E$  un ensemble de points et  $\rho : E \rightarrow \mathcal{P}(Prop)$  une fonction associant à chaque point  $e \in E$  l'ensemble  $\rho(e)$  des prédicats  $p$  qu'il satisfait. Dans ce chapitre, on s'intéresse aux propriétés logiques de ces structures.

Pour ce faire, on explicite quelques résultats élémentaires quant au pouvoir d'expression de la logique du premier ordre sur les structures plates. On montre à nouveau [12] que la logique monadique du second ordre sur ces structures dégénère : son pouvoir d'expression est égal à celui de la logique du premier ordre.

On s'intéresse aussi aux propriétés de ces structures exprimables “sans prédicats d'égalité”. Intuitivement, si on s'intéresse aux propriétés logiques du non déterminisme, il est inutile de pouvoir décrire les éventuelles redondances sémantiques pouvant apparaître dans une liste de possibilités. Cette étude nous conduit à une définition élémentaire de l'équivalence dite de bisimulation. On définit alors un langage combinatoire de “propriétés modales” des structures plates dont le pouvoir expressif est exactement celui de la logique du premier ordre sans égalité.

## 1.1 Logique égalitaire et structures plates

Dans cette section, on caractérise le pouvoir expressif de la logique du premier ordre avec égalité et la logique monadique du second ordre sur les structures plates .

### 1.1.1 Logique du premier ordre sur structures plates

Etant donné  $Prop$  un ensemble de symboles de prédicats unaires, une *structure plate* est toute structure logique sur le vocabulaire réduit aux prédicats unaires de  $Prop$ . Plus précisément, c'est un couple

$$M = \langle E_M, \rho_M \rangle$$

formé d'un ensemble  $E_M$  appelé *univers* de  $M$  et d'une fonction  $\rho_M : E_M \rightarrow \mathcal{P}(Prop)$  associant à chaque point  $e \in E_M$  son *type*  $\rho_M(e)$  c'est à dire l'ensemble des prédicats qu'il satisfait.

**Remarque.** L'ensemble des symboles de prédicats apparaissant dans une formule  $\varphi$  est nécessairement fini. La sémantique de cette formule ne mettant en jeu que ces prédicats, on peut toujours supposer, pour étudier la sémantique des formules du premier ordre, que l'ensemble  $Prop$  des prédicats est fini.

**1.1.1.1 Définition.** Pour toute partie  $P \subseteq Prop$ , on note  $T_P(x)$  le *prédicat de typage*

$$T_P(x) = \bigwedge_{p \in P} p(x) \wedge \bigwedge_{p \notin P} \neg p(x)$$

Dans une formule  $\varphi$ , on appelle *quantification typée* de type  $P$  une sous-formule de la forme :

$$\forall x T_P(x) \Rightarrow \psi$$

ou bien

$$\exists x T_P(x) \wedge \psi$$

cette sous-formule étant *positive* dans  $\varphi$ , i.e. elle apparaît sous un nombre pair de négations. On dit alors que toutes les occurrences libres de la variable  $x$  dans  $\psi$  sont des occurrences typées de type  $P$ .

**Remarque.** Pour toute structure  $M$ , un point  $e \in E_M$  satisfait le prédicat  $T_P(x)$  ssi  $\rho_M(e) = P$ . En particulier, si  $P$  et  $Q$  sont deux types distincts, il n'existe pas de point  $e \in E_M$  satisfaisant le prédicat  $T_P(x) \wedge T_Q(x)$ ; on dit alors que les types  $P$  et  $Q$  sont disjoints.

**1.1.1.2 Proposition.** *Toute formule du premier ordre  $\varphi$  est équivalente à une formule  $\psi$  dont toutes les quantifications sont typées. On dit alors que la formule  $\psi$  est typée.*

*Preuve.* Quitte à repousser les négations au maximum à l'aide des lois de De Morgan usuelles, on peut supposer que les négations ne portent que sur les prédicats constants. On construit alors la formule  $\psi$  par induction sur la structure de  $\varphi$ , en usant des réécritures suivantes dont on vérifie facilement qu'elles préservent la sémantique :

$$\forall x \psi_1(x) \rightsquigarrow \bigwedge_{P \subseteq Prop} \forall x T_P(x) \Rightarrow \psi_1(x)$$

et, par dualité,

$$\exists x \psi_1(x) \rightsquigarrow \bigvee_{P \subseteq Prop} \exists x T_P(x) \wedge \psi_1(x)$$

□

**Notation.** Dans la suite, on note  $\forall x : P \psi$  (resp.  $\exists x : P \psi$ ) la formule  $\forall x T_P(x) \Rightarrow \psi$  (resp. la formule  $\exists x T_P(x) \wedge \psi$ ).

**1.1.1.3 Définition.** Une formule  $\varphi$  est dite *P-homogène* ou plus simplement *homogène* lorsque toutes les quantifications apparaissant dans  $\varphi$  sont typées de type  $P$ .

**1.1.1.4 Proposition.** *Toute formule  $\varphi$  est équivalente à une combinaison booléenne de formules homogènes.*

*Preuve.* Par 1.1.1.2 on peut supposer que toutes les quantifications apparaissant dans  $\varphi$  sont typées. On constate alors que pour toutes variables  $x$  et  $y$  de types distincts la sous-formule  $x = y$  est équivalente à  $\perp$  (faux) et la sous-formule  $x \neq y$  est équivalente à  $\top$  (vrai). On peut donc supposer que le prédicat d'égalité ne porte que sur des occurrences de variables de même type. Une induction sur la structure de  $\psi$  et les règles usuelles de la logique propositionnelle nous permettent de conclure en usant du fait que, pour toute variable  $x$ , toute partie  $P \subseteq Prop$ , toute formule  $\psi_1$  et  $\psi_2$ , la variable  $x$  n'étant pas libre dans  $\psi_2$ , les règles de réécriture suivantes (et leurs duales dans le cas des quantifications existentielles) préservent l'équivalence sémantique :

1.  $\forall x : P (\psi_1 \wedge \psi_2) \rightsquigarrow (\forall x : P \psi_1) \wedge \psi_2$ ,
2.  $\forall x : P (\psi_1 \vee \psi_2) \rightsquigarrow (\forall x : P \psi_1) \vee \psi_2$ .

□

**Remarque.** Dans la proposition précédente, si on part d'une formule  $\varphi$  dont toutes les quantifications sont typées dans un ensemble  $T_{P_1}, T_{P_2}, \dots, T_{P_n}$  de types, alors notre

construction produit une combinaison booléenne de formules  $P$ -homogènes pour  $P$  variant dans l'ensemble  $\{P_1, \dots, P_n\}$ .

Il nous reste alors, pour que cette étude de la logique du premier ordre sur structures plates soit complète, à examiner le pouvoir expressif des formules homogènes.

En notant  $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$  :

**1.1.1.5 Proposition.** *Pour toute formule  $P$ -homogène  $\varphi$  il existe une partie  $N_\varphi \subseteq \mathbb{N}_\infty$  finie ou de complémentaire fini, telle que, pour toute structure  $M$ , la structure  $M$  est un modèle de  $\varphi$  ssi le cardinal de  $\{e \in E_M \mid \rho_M(e) = P\}$  appartient à  $N_\varphi$ .*

*Preuve.* Remarquons tout d'abord qu'étant données deux structures  $M_1$  et  $M_2$ , si  $M_1 \models \varphi$  et si le nombre de points de  $M_2$  de type  $P$  est égal au nombre de points de  $M_1$  de type  $P$  alors  $M_2 \models \varphi$ . Autrement dit, on peut toujours supposer que tous les points des structures plates considérés dans la suite de cette preuve sont de type  $P$ .

Soit alors  $N_\varphi$  l'ensemble des cardinaux des ensembles  $\rho_M^{-1}(P)$  pour  $M$  variant sur la classe des modèles de  $\varphi$ . Il est clair que cet ensemble caractérise la sémantique de la formule  $\varphi$  au sens où pour toute structure  $M$ ,  $M \models \varphi$  ssi le cardinal de  $\{e \in E_M \mid \rho_M(e) = P\}$  appartient à  $N_\varphi$ .

Il nous reste à prouver que  $N_\varphi$  est fini ou bien de complémentaire fini. Dans le cas où  $N_\varphi$  est fini, il n'y a rien à faire. Supposons donc que  $N_\varphi$  est infini. Il existe nécessairement  $k \in N_\varphi$  supérieur au nombre de quantifications apparaissant dans  $\varphi$  et, par définition de  $N_\varphi$  il existe aussi une structure  $M_1$  ayant exactement  $k$  points de type  $P$  avec  $M_1 \models \varphi$ .

Soit alors  $k' \geq k$ , montrons que  $k' \in N_\varphi$ . Pour ce faire, soit  $M_2$  la structure plate obtenue à partir de  $M_1$  en lui ajoutant  $k' - k$  points de type  $P$ . La théorie des jeux de Ehrenfeucht et Fraïssé nous garantit alors que  $M_1$  et  $M_2$  sont indiscernables par une formule ayant au plus  $k$  quantificateurs. On a donc, en particulier,  $M_2 \models \varphi$  ce dont on déduit  $k' \in N_\varphi$ .  $\square$

**Remarque.** Réciproquement, on vérifie facilement que pour toute partie  $N \subseteq \mathbb{N}_\infty$  finie et ne contenant pas  $\infty$  ou bien cofinie et contenant  $\infty$ , il existe bien une formule  $P$ -homogène  $\varphi$  telle que  $N = N_\varphi$ .

La proposition 1.1.1.4 nous assurant que toute formule  $\varphi$  est équivalente à une disjonction de conjonctions de formules homogènes, on déduit de la proposition précédente, en notant  $N_+ = \{0^+, 1^+, \dots\}$  une copie de l'ensemble des entiers naturels :

**1.1.1.6 Corollaire.** *Pour toute formule  $\varphi$  de la logique du premier ordre, il existe un ensemble fini de fonctions*

$$\{f_1, \dots, f_n\} \subseteq \mathcal{P}(\text{Prop}) \rightarrow \mathbb{N} \cup \mathbb{N}_+$$

*tel que, pour toute structure plate  $M = \langle E, \rho \rangle$ ,  $M \models \varphi$  ssi il existe  $i \in \{1, \dots, n\}$  tel que, pour toute partie  $P \subseteq \text{Prop}$ , si  $f_i(P) = n \in \mathbb{N}$  alors  $|\rho^{-1}(P)| = n$ , et si  $f_i(P) = n_+ \in \mathbb{N}_\infty$  alors  $|\rho^{-1}(P)| \geq n$ .*

On dit parfois que la sémantique de toute formule de la logique du premier ordre sur les structures plates peut être décrite à l'aide de compteurs bornés.

En termes logiques, ce corollaire s'énonce de la façon suivante :

**1.1.1.7 Corollaire.** *Toute formule close  $\varphi$  est équivalente à une disjonction (finie) de formules de la forme :*

$$\exists x_1 : P_{i_1}, \dots, x_n : P_{i_n} (\text{diff}\{x_1, \dots, x_n\} \wedge \forall z \text{diff}\{x_1, x_2, \dots, x_n, z\} \Rightarrow \psi_1(z))$$

où :

1.  $P_1, P_2, \dots, P_k$  désignent  $k$  types avec  $\{i_1, \dots, i_n\} = \{1, \dots, k\}$ ,
2.  $\text{diff}\{x_1, \dots, x_n\}$  désigne la formule  $\bigwedge_{i \neq j} x_i \neq x_j$ ,
3.  $\psi_1(z)$  désigne une formule de la forme  $T_{P_{j_1}}(z) \vee \dots \vee T_{P_{j_m}}(z)$  pour  $\{j_1, \dots, j_m\}$  une partie de  $\{1, \dots, k\}$ .

On dit alors que  $\varphi$  est sous forme normale disjonctive.

**Remarque.** Rechercher et construire, s'il en existe, un modèle d'une formule disjonctive est un problème de complexité linéaire en la taille de la formule.

## 1.1.2 Logique monadique du second ordre

Dans cette partie on montre que la logique monadique du second ordre sur les structures plates est équivalente, en terme de pouvoir d'expression, à la logique du premier ordre.

**1.1.2.1 Théorème.** *Pour toute formule  $\varphi$  de la logique monadique du second ordre sur le vocabulaire des structures plates il existe une formule du premier ordre  $\psi$  équivalente à  $\varphi$ .*

*Preuve.* Ce résultat relativement classique apparaît dans [10]. On en donne ici une nouvelle démonstration.

Remarquons tout d'abord que toute variable ensembliste qui apparaît librement dans une formule jouera le même rôle qu'une constante propositionnelle. Il nous suffit donc de prouver le théorème précédent dans le cas d'une seule quantification du second ordre, la remarque précédente et une induction sur le nombre de quantifications nous permettant alors de conclure. De plus, par dualité on peut se contenter de prouver ce résultat dans le cas d'une quantification existentielle.

Soit donc  $\varphi_1(X)$  une formule du premier ordre sur l'ensemble de symboles propositionnels  $Prop \cup \{X\}$ . On veut montrer que la formule du second ordre  $\exists X \varphi_1(X)$  est équivalente à une formule du premier ordre. Pour ce faire, notons tout d'abord que le résultat 1.1.1.7 s'applique. La forme normale disjonctive obtenue pour  $\varphi_1(X)$  et la distributivité de la quantification ensembliste existentielle sur la disjonction nous ramènent donc à l'étude du cas d'une formule  $\varphi_1$  de la forme, en usant des notations de 1.1.1.7 :

$$\varphi_1(X) = \exists x_1 : P_{i_1}, \dots, x_n : P_{i_n} (diff\{x_1, \dots, x_n\} \wedge \forall z diff\{x_1, x_2, \dots, x_n, z\} \Rightarrow \psi_1(X, z))$$

où, dans ce cas, pour tout indice  $i$ ,  $P_i \subseteq Prop \cup \{X\}$  avec  $\psi_1(z) = T_{P_{j_1}}(z) \vee \dots \vee T_{P_{j_m}}(z)$ . En notant  $\varphi'$  la formule obtenue à partir de  $\varphi_1$  en remplaçant chaque type  $P_i$  par  $P_i \cap Prop$ , on vérifie que les formules  $\exists X \varphi_1(X)$  et  $\varphi'$  sont équivalentes.  $\square$

## 1.2 Le cas sans égalité

Dans cette section, on s'intéresse à la logique du premier ordre sans égalité sur les structures plates .

On note  $Bool(Prop)$  la clôture booléenne de l'ensemble  $Prop$  des symboles ensemblistes. La fonction d'étiquetage des états d'une structure plate est naturellement étendue à ces combinaisons booléennes, i.e. pour tout état  $e$  d'une structure  $M$ ,  $p_1 \wedge p_2 \in \rho_M(e)$  ssi  $\{p_1, p_2\} \subseteq \rho_M(e)$ ,  $p_1 \vee p_2 \in \rho_M(e)$  ssi  $\{p_1, p_2\} \cap \rho_M(e) \neq \emptyset$ , et  $\neg p_1 \in \rho_M(e)$  ssi  $p_1 \notin \rho_M(e)$ . La notion de type de point définie précédemment reste la même dans cette partie.

### 1.2.1 Formules modales

**1.2.1.1 Définition.** On définit l'ensemble  $Mod$  des *formules modales plates* comme l'ensemble des formules  $\alpha$  de la forme :

$$\alpha ::= \top \mid \perp \mid \alpha_1 \wedge \alpha_2 \mid \alpha_1 \vee \alpha_2 \mid \diamond p \mid \square p$$

pour :

1. tout prédicat  $p \in Bool(Prop)$ ,

2. toutes formules  $\alpha_1$  et  $\alpha_2$  de  $Mod$ .

Pour toute formule  $\alpha \in Mod$ , on appelle modèle de  $\alpha$  toute structure plate

$$M = \langle E_M, \rho_M \rangle$$

telle que la relation  $M \models \alpha$  est satisfaite selon les règles usuelles pour les conjonctions et disjonctions, la sémantique des modalités étant décrite par :

1.  $M \models \Diamond p$  lorsque il existe un point  $e \in E_M$  tel que  $p \in \rho_M(e)$ ,
2.  $M \models \Box p$  lorsqu'en tout point  $e \in E_M$  on a  $p \in \rho_M(e)$ .

**Remarque.** Notons ici que pour toute formule  $\alpha \in mod$  il existe une formule  $\varphi_\alpha$  de la logique du premier ordre sans égalité telle que, pour toute structure plate  $M$ ,

$$M \models \alpha \text{ ssi } M \models \varphi_\alpha$$

La réciproque de cette remarque est prouvée proposition 1.2.2.3.

**1.2.1.2 Définition.** Pour toute structure plate  $M$ , on définit le *type modal* de  $M$ , noté  $T_{mod}(M)$ , comme l'ensemble

$$T_{mod}(M) = \{\rho_M(e) \mid e \in M\}$$

Deux structures plates  $M$  et  $N$  sont dites *équivalentes par bisimulation* lorsqu'elles ont même type modal, i.e.  $T_{mod}(M) = T_{mod}(N)$ ; de façon équivalente, lorsqu'il existe une relation  $R \subseteq E_M \times E_N$ , appelée *relation de bisimulation*, telle que :

1. pour tout  $x_1 \in E_{M_1}$  il existe  $x_2 \in E_{M_2}$  tel que  $\rho_{M_1}(x_1) = \rho_{M_2}(x_2)$ ,
2. pour tout  $x_2 \in E_{M_2}$  il existe  $x_1 \in E_{M_1}$  tel que  $\rho_{M_1}(x_1) = \rho_{M_2}(x_2)$ .

**1.2.1.3 Proposition.** *Deux structures plates  $M$  et  $N$  sont équivalentes par bisimulation ssi pour toute formule  $\alpha \in mod$ ,  $M \models \alpha$  ssi  $N \models \alpha$ .*

*Preuve.* On se place ici dans le cas où *Prop* est finie. Si ce n'est pas le cas, il faut autoriser dans *mod* des disjonctions et conjonctions de même cardinalités que *Prop*.  $\square$

## 1.2.2 Caractérisation modale des formules sans égalité

**1.2.2.1 Définition.** Pour toute formule  $\varphi$  de la logique du premier ordre on définit aussi le type modal  $T_{mod}(\varphi)$  comme l'ensemble

$$T_{mod}(\varphi) = \{\pi_\varphi(T_{mod}(M)) \mid M \models \varphi\}$$

où  $Prop_\varphi$  désigne l'ensemble des symboles  $p \in Prop$  apparaissant dans  $\psi$  et  $\pi_\varphi$  désigne la projection de  $\mathcal{P}(\mathcal{P}(Prop))$  dans  $\mathcal{P}(\mathcal{P}(Prop_\varphi))$ , i.e. pour tout ensemble  $A \subseteq \mathcal{P}(Prop)$ ,

$$\pi_\varphi(A) = \{P \cap Prop_\varphi \mid P \in A\}$$

**1.2.2.2 Proposition.** *Pour toute formule  $\varphi$  sans égalité, pour toute structure plate  $M$ ,  $M \models \varphi$  ssi  $\pi_\varphi(T_{mod}(M)) \in T_{mod}(\varphi)$ .*

*Preuve.* Pour simplifier les notations, on suppose tout d'abord que  $Prop = Prop_\varphi$ . Soient alors  $M_1$  et  $M_2$  deux structures plates telle que  $T_{mod}(M_1) = T_{mod}(M_2)$ , i.e. l'ensemble des types des points de  $M_1$  est égal à l'ensemble des types des points de  $M_2$ . On démontre tout d'abord par induction sur la longueur de  $\mathbf{x}$  et la structure de  $\varphi(\mathbf{x})$  que pour tout vecteur de variables deux à deux distinctes  $\mathbf{x} = (x_1, \dots, x_n)$ , pour toute formule du premier ordre  $\varphi(\mathbf{x})$  sans égalité et sans quantifications, dont les variables libres sont parmi  $\{x_1, \dots, x_n\}$ , pour tout vecteur  $\mathbf{a} \in E_{M_1}^n$  et  $\mathbf{b} \in E_{M_2}^n$  si  $\rho_{M_1}(\mathbf{a}) = \rho_{M_2}(\mathbf{b})$  alors  $M_1 \models \varphi[\mathbf{a}]$  ssi  $M_2 \models \varphi[\mathbf{b}]$ .

Les jeux de Ehrenfeucht et Fraïssé tels qu'on les présente dans l'annexe A.2 nous permettent alors de conclure.  $\square$

Il vient donc :

**1.2.2.3 Proposition.** *Toute formule  $\varphi$  de la logique du premier ordre sur les structures plates ne contenant pas le prédicats d'égalité est équivalente à la formule modale  $\alpha_\varphi$  définie par :*

$$\alpha_\varphi = \bigvee_{T_m \in T_{mod}(\varphi)} \left( \bigwedge_{P \in T_m} \diamond T_P \wedge \square \bigvee_{P \in T_m} T_P \right)$$

*Une telle formule est appelée disjonctive.*

*Preuve.* On vérifie facilement que le type modal de  $T_{mod}(\alpha_\varphi)$  est bien  $T_{mod}(\varphi)$ . La proposition 1.2.2.2 nous permet alors de conclure.  $\square$

**Remarque.** Là encore, la construction explicite, s'il en existe, d'un modèle d'une formule modale disjonctive est un problème de complexité linéaire en la taille de cette formule.

## 1.3 Propriétés d'ensembles et ensembles de propriétés

Dans cette section, on généralise les travaux précédents en s'intéressant à la description des propriétés d'ensembles de structures.

Plus précisément, étant donné un vocabulaire  $L$ , étant donné une classe de  $L$ -structures  $\mathcal{C}$  définissable en logique du premier ordre par une formule  $\varphi_{\mathcal{C}}$ , on souhaite décrire les propriétés des ensembles de  $L$ -structures de  $\mathcal{C}$ .



Pour ce faire, on montre tout d'abord que  $\mathcal{P}(\mathcal{C})$ , la classe des ensembles de  $L$ -structures de  $\mathcal{C}$ , est en correspondance bijective avec un ensemble définissable au premier ordre de  $PL$ -structures pour  $PL$  un vocabulaire approprié.

Ce résultat nous permet ensuite de considérer les ensembles de structures de  $\mathcal{C}$  “comme” des structures plates, nous permettant ainsi, via notre présentation logique des domaines rappelée dans l'annexe B, de montrer comment les trois constructions usuelles de “power-domain” peuvent être généralisées.

Ce travail généralise et précise les travaux d'Abramsky [1, 2], de Robinson [41] et Winskel [49].

### 1.3.1 Un vocabulaire pour les ensembles de structures

Dans cette partie, on montre comment construire, pour tout vocabulaire fini  $L$ , un vocabulaire  $PL$  permettant de “coder” les ensembles de  $L$ -structures comme des  $PL$ -structures. Ce codage est réalisé de telle sorte que tout ensemble de structures peut être manipulé “comme” une structure plate dont les points sont les structures considérées. Une telle construction nous a été suggérée par Bruno Courcelle.

Soit donc  $L$  un vocabulaire. On va définir le vocabulaire  $PL$  et une formule de logique du premier ordre  $\mathcal{P}(L)\text{-struct}_L^{-1}$  sur le vocabulaire  $PL$  tels que, en notant  $\text{mod}(\mathcal{P}(L)\text{-struct})$  la classe des modèles de la formule  $\mathcal{P}(L)\text{-struct}$  et  $\mathcal{C}_L$  la classe des  $L$ -structures, il existe une correspondance bijective

$$h : \text{mod}(\mathcal{P}(L)\text{-struct}) \rightarrow \mathcal{P}(\mathcal{C}_L)$$

nous permettant d'interpréter  $\mathcal{P}(\mathcal{C}_L)$  comme la classe des structures plates dont les univers sont composés de  $L$ -structures, les “prédicats” sur ces  $L$ -structures pouvant être n'importe quelle formule de logique monadique du second ordre sur le vocabulaire  $L$ .

Soit en effet  $PL$  le vocabulaire composé de tous les symboles de  $L$  augmenté des prédicats  $\text{point}(x)$  et  $\text{élément-de}(x, y)$ , et dont tous les symboles de constantes de  $L$  sont interprétés dans  $PL$  comme des prédicats unaires.

Intuitivement, pour toute  $PL$ -structure  $M$ , les éléments  $x \in E_{PM}$  satisfaisant le prédicat  $\text{point}(x)$  seront interprétés comme des éléments de l'univers d'une  $L$ -structure se trouvant “dans”  $M$ , de même chaque élément  $x \in E_{PM}$  ne satisfaisant pas le prédicat  $\text{point}(x)$  désignera l'une de ces  $L$ -structures. Avec cette intuition, le prédicat  $\text{élément-de}(x, y)$  permettra d'associer à chaque “ $L$ -structure”  $y$  l'ensemble des points  $x$  de son “univers”.

---

<sup>1</sup>avec une conjonction de cardinalité  $|L|$  dans le cas où  $L$  est infini

Bien entendu, toute  $PL$ -structure ne se prête pas à une telle interprétation. Il faut imposer les conditions suivantes, à l'aide de formules sur le vocabulaire  $PL$  :

1. la formule  $\psi_1$  spécifie, que, “dans” une  $PL$ -structure, une “ $L$ -structure” ne peut faire partie de l'univers d'une autre  $L$ -structure :

$$\psi_1 = \forall y (\exists x \text{ élément-de}(x, y)) \Rightarrow \neg \text{point}(y)$$

2. la formule  $\psi_2$  spécifie que tout “point” appartient à une et une seule  $L$ -structure :

$$\psi_2 = \forall x \text{ point}(x) \Rightarrow \exists ! y \text{ élément-de}(x, y)$$

3. pour tout arité  $i > 0$  de symbole dans  $L$ , en notant  $\{R_{i,1}, \dots, R_{i,j}, \dots\}$  les symboles relationnels dans  $L$  d'arité  $i$  la formule  $\psi_{3,i,j}$  spécifie que l'interprétation du symbole relationnel  $R_{i,j}$  n'est définie qu'à l'“intérieur” des  $L$ -structures :

$$\psi_{3,i,j} = \forall x_1, \dots, x_i R_{i,j}(x_1, \dots, x_i) \Rightarrow \exists y \bigwedge_{k \in \{1, \dots, i\}} \text{élément-de}(x_k, y)$$

4. en notant  $c_1, \dots, c_k, \dots$ , les constantes de  $L$ , la formule  $\psi_{4,k}$  spécifie que chaque  $L$ -structures admet une et une seule interprétation de la constante  $c_k$  (qui est un prédicat unaire dans le vocabulaire  $PL$ ) :

$$\psi_{4,k} = \forall y \neg \text{point}(y) \Rightarrow \exists ! x \text{ élément-de}(x, y) \wedge c_i(x)$$

On définit alors

$$\mathcal{P}(L)\text{-struct} = \psi_1 \wedge \psi_2 \wedge \bigwedge_{i,j} \psi_{3,i,j} \wedge \bigwedge_k \psi_{4,k}$$

On se convainc facilement que toute  $PL$ -structure  $M$  satisfaisant la formule  $\mathcal{P}(L)\text{-struct}$  “code” un ensemble de  $L$ -structures caractérisé par l'ensemble des points  $x \in E_M$  tels que  $\neg \text{point}(x)$ . Réciproquement, tout ensemble de  $L$ -structures peut être codé par une  $PL$ -structure satisfaisant cette même conjonction.

**Notation.** Dans la suite, pour toute  $PL$ -structure  $M \in \text{mod}(\mathcal{P}(L)\text{-struct})$ , pour tout  $y \in E_M$  ne satisfaisant pas le prédicat  $\text{point}(y)$ , on note  $L\text{-struct}(y)$  la  $L$ -structure dénotée par  $y$  dans  $M$ .

**1.3.1.1 Proposition.** *Pour toute MS-formule close  $\varphi$  sur le vocabulaire  $L$  il existe une formule  $L\text{-sat-}\varphi(y)$  sur le vocabulaire  $PL$  telle que, pour toute  $PL$ -structure  $M$  modèle de  $\mathcal{P}(L)\text{-struct}$ , pour tout point  $e \in E_M$ ,*

$$M \models \neg \text{point}[e] \wedge L\text{-sat-}\varphi[e] \text{ ssi } L\text{-struct}(e) \models \varphi$$

*Preuve.* Soit  $\varphi$  une *MS*-formule close sur le vocabulaire  $L$ , soient  $c_1, \dots, c_n$  les  $n$ -constants apparaissant dans  $\varphi$ , et soient  $y, x_1, x_2, \dots, x_n, n+1$  variables n'ayant aucune occurrence dans  $\varphi$ . On définit la formule

$$\varphi'(y, x_1, x_2, \dots, x_n)$$

obtenue à partir de  $\varphi$  via les règles de réécriture :

1. pour tout  $i \in \{1, \dots, n\}$   $c_i \rightsquigarrow x_i$ ,
2. sous réserve que *élément-de*( $x, y$ ) n'est pas une sous-formule de  $\varphi_1$  :

$$\exists x \varphi_1 \rightsquigarrow \exists x \text{élément-de}(x, y) \wedge \varphi_1$$

et

$$\forall x \varphi_1 \rightsquigarrow \forall x \text{élément-de}(x, y) \Rightarrow \varphi_1$$

3. sous réserve que *élément-de*( $x, y$ ) n'est pas une sous-formules de  $\varphi_1$  :

$$\exists X \varphi_1 \rightsquigarrow \exists X (\forall x \in X \text{élément-de}(x, y)) \wedge \varphi_1$$

et

$$\forall X \varphi_1 \rightsquigarrow \forall X (\forall x \in X \text{élément-de}(x, y)) \Rightarrow \varphi_1$$

On se convainc sans difficulté que la formule :

$$L\text{-sat-}\varphi(y) = \exists x_1, \dots, x_n \varphi'(y, x_1, \dots, x_n) \bigwedge_{i \in \{1, \dots, n\}} \text{élément-de}(x_i, y) \wedge c_i(x_i)$$

a bien les propriétés souhaitées. □

**Remarque.** Si  $\varphi$  est une formule du premier ordre (resp. du premier ordre sans égalité) alors  $L\text{-sat-}\varphi(x)$  est définissable au premier ordre (resp. au premier ordre sans égalité).

Dans la suite, on manipulera sans scrupules les ensembles de  $L$ -structures comme des structures plates .

### 1.3.2 Trois sémantiques pour les ensembles de propriétés

Dans cette partie, on décrit les propriétés des ensembles de structures à l'aide d'ensembles de propriétés dont on donne trois sémantiques possibles.

On se donne un vocabulaire  $L$ , la classe  $\mathcal{C}$  des  $L$ -structures, et  $\Phi$  un ensemble de formules.

Pour tout ensemble de structures  $S \in \mathcal{P}(\mathcal{C})$ , tout ensemble fini de formules  $A \in \mathcal{P}_{fin}(\Phi)$ , on étend la relation de satisfaction  $\models$  par :

1. **Hoare** :  $S \models A$  lorsque pour toute formule  $\varphi \in A$  il existe  $M \in S$  telle que  $M \models \varphi$ ,
2. **Smyth** : pour toute structure  $M \in S$  il existe  $\varphi \in A$  telle que  $M \models \varphi$ ,
3. **Plotkin** : les deux conditions précédentes sont satisfaites.

**Remarque.** Dans le cas de Hoare, lorsqu'au moins une des propriétés de  $A$  garantit un comportement acceptable, le non-déterminisme envisagé est *angélique*. Chaque propriété de  $A$  est satisfaite par une structure de  $S$ , et qu'importe s'il reste des structures dans  $S$  décrivant un comportement catastrophique.

Au contraire, dans le cas de Smyth, le non-déterminisme envisagé est *démoniaque*. Toute structure de  $S$  doit satisfaire au moins une formule de  $A$ . Si toutes les formules de  $A$  garantissent des comportements acceptables, on est certain de la "qualité" de l'alternative décrite par  $S$ .

Avec les notations modales, les sémantiques de Hoare, Smyth et Plotkin d'un ensemble fini de formules  $A \subseteq \Phi$  s'expriment respectivement comme :

1. pour la sémantique de Hoare comme la formule  $\bigwedge\{\diamond\varphi \mid \varphi \in A\}$ ; en particulier, l'union  $A \cup B$  de deux ensembles de formules  $A$  et  $B$  est équivalente à la "conjonction"  $A \wedge B$  au sens où, pour tout ensemble  $S$  de  $\mathcal{C}$ -structures,

$$S \models_{hoare} A \cup B \text{ ssi } S \models_{hoare} A \text{ et } S \models_{hoare} B$$

2. pour la sémantique de Smyth comme la formule  $\square \bigvee A$ ,
3. pour la sémantique de Plotkin comme la formule  $\bigwedge\{\diamond\varphi \mid \varphi \in A\} \wedge \square \bigvee A$ .

**Remarque.** On appelle dans ce qui suit *pré-ordre sémantique* l'implication inverse, i.e.  $\varphi$  précède sémantiquement  $\psi$  lorsque tout modèle de  $\psi$  est aussi un modèle de  $\varphi$ . Une définition précise est donnée en annexe B.2.1.2.

**1.3.2.1 Proposition.** *Les préordres sémantiques associés à ces trois sémantiques sont donnés, pour toutes parties finies  $A_1$  et  $A_2$  de  $\Phi$  par :*

1.  $A_1 \preceq_{hoare} A_2$  ssi pour toute formule  $\varphi_1 \in A_1$  il existe une formule  $\varphi_2 \in A_2$  telle que  $\varphi_1 \preceq_{\mathcal{C}} \varphi_2$ ,
2.  $A_1 \preceq_{smyth} A_2$  ssi  $\bigvee A_1 \preceq_{\mathcal{C}} \bigvee A_2$ ,

3.  $A_1 \preceq_{plotkin} A_2$  ssi  $A_1 \preceq_{smyth} A_2$  et  $A_1 \preceq_{hoare} A_2$ .

*Preuve.* Seul le cas de Hoare nécessite une petite démonstration. Par l'absurde, supposons qu'il existe une formule  $\varphi \in A_1$  tel que, pour toute formule  $\psi \in A_2$  on ait  $\varphi \not\leq_C \psi$ . Autrement dit, pour toute formule  $\psi \in A_2$  il existe une structure  $M_\psi \in \mathcal{C}$  telle que  $M_\psi \models \psi$  alors que  $M_\psi \not\models \varphi$ . On considère alors l'ensemble de structures  $\{M_\psi\}_{\psi \in A_2} \in \mathcal{P}(\mathcal{C})$ . C'est un modèle (au sens de la sémantique de Hoare) de  $A_2$  mais par construction, ce n'est pas un modèle de  $A_1$  (toujours pour la même sémantique), ce qui contredit l'hypothèse  $A_1 \preceq_{hoare} A_2$ .  $\square$

**Remarque.** Seule la sémantique de Hoare conduit au préordre sur les parties finies de  $\Phi$  qui est généralement utilisé dans la construction des “power-domains” [39, 42].

### 1.3.3 Connexions avec les constructions classiques

Dans cette partie, on utilise le vocabulaire de la présentation logique des domaines présentée annexe B. Elle est indépendante de la suite de ce mémoire.

On suppose maintenant que  $\Phi$  est  $\mathcal{C}$ -conditionnellement close par conjonction B.2.1.7. En particulier, en appliquant le théorème B.2.1.6, il vient que l'ensemble des classes de théories consistantes et équivalente sur  $\mathcal{C}$  ordonnées par implication sémantique inverse forme un domaine de Scott-Ershov.

On note  $\mathcal{P}(\mathcal{C})$  (resp.  $\mathcal{P}^*(\mathcal{C})$ ) la classe des ensembles (resp. des ensembles non vides) de  $L$ -structures. On note aussi  $\mathcal{P}_{fin}(\Phi)$  (resp.  $\mathcal{P}_{fin}^*(\Phi)$ ) l'ensemble des parties finies (resp. des parties finies non vides) de  $\Phi$ .

Le pré-ordre  $\preceq_{hoare}$  défini ci-dessus conduisant au pré-ordre usuellement associé à la construction du “power-domain” de Hoare, il vient :

**1.3.3.1 Théorème.** *Le “power-domain” de Hoare  $\mathcal{P}_H(T(\Phi)/\mathcal{C})$  est isomorphe au domaine  $T(\mathcal{P}_{fin}(\Phi))/\mathcal{P}(\mathcal{C})$  pour la sémantique de Hoare.*

*Preuve.* On constate tout d'abord que  $\mathcal{P}_{fin}(\Phi)/\mathcal{P}(\mathcal{C})$  ordonné par  $\preceq_{hoare}$  est conditionnellement clos par conjonction sur  $\mathcal{P}(\mathcal{C})$ . En effet, le minimum est  $\emptyset$ , tout ensemble fini de formules est satisfaisable dans  $\mathcal{C}$  au sens de la sémantique de Hoare et, pour tout ensemble fini  $A_1$  et  $A_2 \subseteq \Phi$ , on a :

$$A_1 \wedge A_2 \simeq_{hoare} A_1 \cup A_2$$

La remarque précédente, le corollaire B.1.2.5, le théorème B.2.1.6 et la proposition B.2.1.8 nous permettent alors de conclure.  $\square$

**Remarque.** Les propriétés des “power-domains” nous assurent que la fonction singleton

$\diamond : T(\Phi)/\mathcal{C} \rightarrow T(\mathcal{P}_{fin}(\Phi))/\mathcal{P}(\mathcal{C})$  est, pour la sémantique de Hoare, continue. Or, pour toute théorie  $T \subseteq \Phi$  dirigée et  $\mathcal{C}$ -consistante on a certes :

$$\bigwedge \{\diamond\varphi \mid \varphi \in T\} \preceq_{hoare} \diamond \bigwedge T$$

mais l'équivalence n'est pas satisfaite en toute généralité. En fait, ce paradoxe apparent s'explique par le fait que  $\diamond \bigwedge T$  n'est pas une théorie de  $T(\mathcal{P}_{fin}(\Phi))/\mathcal{P}(\mathcal{C})$ . Pour définir l'image dans  $T(\mathcal{P}_{fin}(\Phi))/\mathcal{P}(\mathcal{C})$  par la fonction “singleton”  $\diamond$  de toute théorie  $T \in T(\Phi)/\mathcal{C}$  il est donc nécessaire de procéder par approximation, i.e. prendre le supremum des éléments compacts de  $T(\mathcal{P}_{fin}(\Phi))/\mathcal{P}(\mathcal{C})$  inférieurs à  $\diamond \bigwedge T$ . Il apparaît alors que ce supremum est bien la classe d'équivalence (modulo  $\simeq_{hoare}$ ) de la théorie

$$\{\diamond\varphi \mid \varphi \in T\}$$

En un sens, la continuité est ici “forcée” !

Pour reconstruire les autres “power-domains”, il nous faut imposer une condition supplémentaire: la propriété de couverture simple B.2.2.3, afin de retrouver les pré-ordres utilisés dans les constructions usuelles.

Il vient en effet :

**1.3.3.2 Théorème.** *Lorsque  $\Phi$  couvre simplement  $\mathcal{C}$  le “power-domain”  $\mathcal{P}_S(T(\Phi)/\mathcal{C})$  de Smyth est isomorphe au domaine  $T(\mathcal{P}_{fin}(\Phi))/\mathcal{P}(\mathcal{C})$  pour la sémantique de Smyth.*

*Preuve.* La condition de couverture simple nous assure que l'ordre sémantique  $\preceq_{smyth}$  coïncide bien avec l'ordre  $\subseteq_S$  dit de Smyth donné par

$$A_1 \subseteq_S A_2 \text{ lorsque } \forall \varphi \in A_2 \exists \psi \in A_1 \psi \preceq_{\mathcal{C}} \varphi$$

On montre alors que  $\mathcal{P}_{fin}(\Phi)/\mathcal{P}(\mathcal{C})$  est, vis à vis de la sémantique de Smyth, conditionnellement clos par conjonction. En effet, on prend pour minimum l'ensemble  $\{\varphi_{\mathcal{C}}\}$ , toute partie de  $\Phi$  est satisfaisable au sens de cette sémantique, et pour toutes parties finies  $A_1$  et  $A_2$  de  $\Phi$  on a, lorsque  $A_1 \wedge A_2$  est satisfaisable dans  $\mathcal{P}(\mathcal{C})$ :

$$A_1 \wedge A_2 \simeq_{smyth} \{\varphi \wedge \psi \in \Phi \mid \varphi \in A_1 \wedge \psi \in A_2\}$$

en notant que la condition  $\varphi \wedge \psi \in \mathcal{C}$  est à comprendre modulo équivalence sémantique, et implique bien entendu que  $\varphi \wedge \psi$  est satisfaisable dans  $\mathcal{C}$ . La proposition B.2.1.8 nous permet de conclure.  $\square$

Pour la sémantique de Plotkin, l'argument précédent ne peut être utilisé. Rien n'assure que  $\mathcal{P}_{fin}^*(\Phi)$  est conditionnellement close par conjonction. On a en effet, pour toutes

parties finies  $A_1$  et  $A_2$  de  $\Phi$  :

$$A_1 \wedge A_2 \simeq_{Plotkin} \{\varphi \wedge \bigvee A_2 \mid \varphi \in A_1\} \cup \{\bigvee A_1 \wedge \psi \mid \psi \in A_2\}$$

rien n'assurant que  $\Phi$ , même lorsqu'elle couvre simplement  $\mathcal{C}$ , est close par disjonction. On obtient cependant le “power-domain” de Plotkin comme le domaine des théories dirigées de  $\mathcal{P}_{fin}^*(\Phi)$ . Plus précisément :

**1.3.3.3 Théorème.** *Lorsque  $\Phi$  couvre simplement  $\mathcal{C}$  le “power-domain”  $\mathcal{P}_P(T(\Phi)/\mathcal{C})$  de Plotkin est isomorphe au domaine  $TD(\mathcal{P}_{fin}^*(\Phi))/\mathcal{P}(\mathcal{C})$  pour la sémantique de Plotkin.*

*Preuve.* Immédiat selon la remarque de la page 115. □





## Chapitre 2

# Systèmes de transitions, bisimulation et logique

Dans ce chapitre, on définit la notion de systèmes de transitions qui apparaît dans de nombreux cas comme un modèle adéquat de système informatique. Pour simplifier les définitions et notations, on supposera, contrairement à la tradition, que les systèmes de transitions sont toujours munis d'un état distingué appelé *état initial* ou bien *racine*.

La notion, très intuitive, de comportement de système informatique, conduit cependant à s'intéresser plus aux arbres induits par les systèmes de transitions, i.e. les arbres obtenus par "développement" ou "expansion" des systèmes de transitions à partir de leur racine. On explicite alors les résultats et outils fondamentaux pour l'étude "comportementale" de ces modèles de machines.

### 2.1 Définitions et propriétés élémentaires des systèmes de transitions

Dans cette partie, nous donnons les définitions des systèmes et arbres de transitions. Nous rappelons en particulier la définition de l'équivalence de bisimulation sur les systèmes de transitions et sa justification logique.

Dans tout ce qui suit,  $Act = \{a, b, c, \dots\}$  désigne un ensemble au plus dénombrable d'étiquettes appelées *actions*, et  $Prop = \{p, q, \dots\}$  un ensemble au plus dénombrable de symboles ensemblistes.

### 2.1.1 Systèmes de transitions

**2.1.1.1 Définition.** Un *système de transitions* est un quadruplet

$$S = \langle E_S, r_S, R_S, \rho_S \rangle$$

avec :

- un ensemble d'*états* ou *sommets*  $E_S$ ,
- un *état initial* ou *sommet initial*  $r_S \in E_S$ ,
- une fonction de transition  $R_S : Act \rightarrow \mathcal{P}(E_S \times E_S)$  telle que, pour toute action  $a \in Act$ , la relation  $R_S(a) \subseteq E_S \times E_S$  est appelée *relation de transition*,
- une fonction d'étiquetage des états  $\rho_S : E_S \rightarrow \mathcal{P}(Prop)$  associant à chaque état  $e \in E_S$  l'ensemble  $\rho_S(e)$  des propriétés valides dans cet état.

Un tel système est dit à *branchement fini* lorsque, pour tout état  $e \in E_S$ , toute action  $a \in Act$ , l'ensemble  $\{f \in E_S \mid (e, f) \in R_S(a)\}$  est fini.

**Remarque.** D'un point de vue logique, un système de transitions apparaît effectivement comme une structure sur le vocabulaire formé de la constante  $r$ , des symboles de prédicats unaires  $p \in Prop$  et, pour toute action  $a \in Act$ , d'un symbole de relation binaire qui sera noté, si besoin est,  $R_a$ .

**Notation.** Etant donné deux états  $e$  et  $f$  d'un système  $S$ , une action  $a \in Act$ , on note aussi  $e \xrightarrow{a}_S f$  la propriété  $(e, f) \in R_S(a)$ , i.e. dans le système  $S$ , la paire  $(e, f)$  est une transition étiquetée par l'action  $a$ . On dira alors que  $f$  est un *a-successeur* ou *a-fils* de l'état  $e$ . L'ensemble des *a-successeurs* de  $e$  est noté  $Succ_a(e)$ . On note aussi :

$$Succ(e) = \bigcup_{a \in Act} Succ_a(e)$$

l'ensemble de tous les *successeurs* ou *fils* de l'état  $e$ . Pour tout état  $e \in E_S$ , on note aussi :

$$S[e/r_S] = \langle E_S, e, R_S, \rho_S \rangle$$

le système obtenu à partir de  $S$  en changeant la racine  $r_S$  en l'état  $e$ .

**Remarque.** Par abus de notation, étant donnés deux systèmes  $S_1$  et  $S_2$  on note aussi, pour toute action  $a \in Act$ ,

$$S_1 \xrightarrow{a} S_2$$

le fait qu'il existe un état  $e \in E_{S_1}$  tel que  $r_{S_1} \xrightarrow{a}_{S_1} e$  les systèmes de transitions  $S_2$  et  $S_1[e/r_{S_1}]$  étant isomorphes. Cette notation induit une relation de transition sur la classe des systèmes de transitions définissant ainsi un "système de transition" universel dans lequel tout système de transitions peut être plongé. Ainsi, toutes les définitions qui suivent pourraient directement porter sur une telle "structure", au risque, cependant, de soulever quelques paradoxes. Les états de ce système de transitions universel forment une classe propre au sens de la théorie des ensembles. Dans la suite, on préfère donc, le plus souvent possible, faire porter les définitions sur des systèmes de transitions donnés, ces définitions se généralisant chaque fois, via des abus de notations similaires, au système universel évoqué ici.

**2.1.1.2 Définition (Chemins dans un système).** Etant donné un système de transitions  $S$  et deux états  $e$  et  $f \in E_S$ , un *chemin fini* de  $e$  à  $f$  est un mot  $e_0.a_1.e_1.a_2.\dots.a_n.e_n \in E_S.(Act.E_S)^*$  tel que  $e = e_0$ ,  $f = e_n$ , et pour tout  $i \in \{1, \dots, n\}$ ,  $e_{i-1} \xrightarrow{a_i}_S e_i$ . La *longueur* d'un tel chemin est l'entier  $n$ . On dit alors qu'un état  $f \in E_S$  est *accessible* à partir d'un état  $e \in E_S$  lorsqu'il existe un chemin de  $e$  à  $f$ .

## 2.1.2 Développement d'un système et arbres de transitions

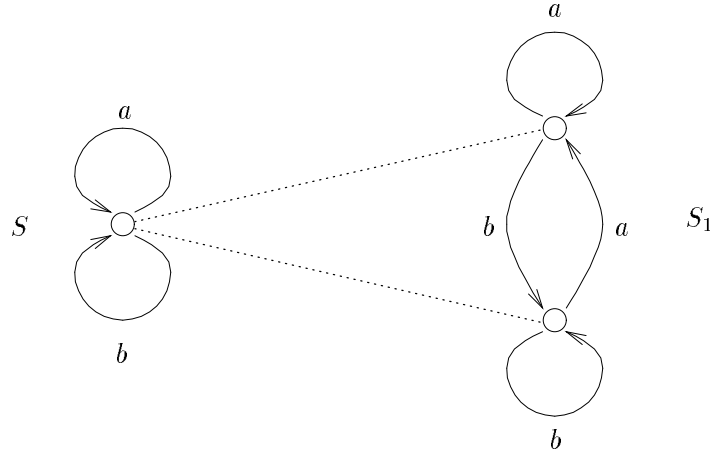
Dans [13] Courcelle formalise une première distinction entre modèle syntaxique de machine et modèle syntaxique de comportements. Dans son approche, les comportements sont définis comme des développements de systèmes de transitions.

**2.1.2.1 Définition (développement d'un système).** Etant donnés deux systèmes de transitions  $S_1 = \langle E_1, r_1, R_1, \rho_1 \rangle$  et  $S_2 = \langle E_2, r_2, R_2, \rho_2 \rangle$ , on dit que  $S_1$  est un *développement* de  $S_2$  lorsqu'il existe une fonction *partielle*  $h : E_1 \rightarrow E_2$  telle que :

1.  $h$  est définie sur  $r_1$  avec  $h(r_1) = r_2$ ,
2. pour tout état  $e \in E_1$  sur laquelle  $h$  est définie :
  - (a)  $\rho_1(e) = \rho_2(h(e))$ ,
  - (b) pour toute action  $a \in Act$ , la restriction de  $h$  à  $\text{Succ}_a(e)$  est une **bijection** de  $\text{Succ}_a(e)$  dans  $\text{Succ}_a(h(e))$ ,

**Remarque.** La fonction  $h$  ci-dessus induit un isomorphisme entre les chemins de  $S_1$  et de  $S_2$  issus des racines  $r_{S_1}$  et  $r_{S_2}$ .

La figure 2 illustre cette notion de développement.

FIG. 2 -  $S_1$  est un développement de  $S$ 

**Remarque.** La composition de deux morphismes au sens ci-dessus restant un morphisme, la notion de développement est transitive.

Il vient :

**2.1.2.2 Théorème.** *Pour tout système de transitions  $S$  il existe un développement  $\widehat{S}$  du système  $S$  tel que pour tout système  $S_1$ , si  $S_1$  est un développement de  $S$  alors  $\widehat{S}$  est un développement de  $S_1$ .*

*Preuve.* On construit  $\widehat{S}$  comme l'arbre des chemins issus de la racine de  $S$ . Plus précisément, on pose :

$$\widehat{S} = \langle E_{\widehat{S}}, r_{\widehat{S}}, R_{\widehat{S}}, \rho_{\widehat{S}} \rangle$$

où :

- l'ensemble des états  $E_{\widehat{S}}$  est l'ensemble des chemins de  $S$  issus de l'état initial  $r_S$ ,
- l'état initial est  $r_{\widehat{S}} = r_S$ ,
- pour toute action  $a \in Act$ , toute paire de chemins  $u$  et  $v$ , avec  $u$  de la forme  $u_1.e$  on pose  $(u, v) \in R_{\widehat{S}}(a)$  lorsque  $v$  est de la forme  $u_1.e.a.f$  avec  $(e, f) \in R_S(a)$ .
- et, pour tout  $u$  de la forme  $u_1.e$ ,  $\rho_{\widehat{S}}(u) = \rho_S(e)$ .

Il est immédiat de constater que  $\widehat{S}$  ainsi défini satisfait bien les conditions du théorème.  $\square$

**Remarque.** Un tel développement est, à isomorphisme près, unique. On dit alors que  $\widehat{S}$  est le *développement universel* de  $S$ .

**2.1.2.3 Définition (Arbre de transitions).** On appelle *arbre de transitions* tout système  $S$  isomorphe à son développement universel.

### 2.1.3 Expansion d'un système, simulation et bisimulation

Le comportement d'un système  $S$  peut être défini comme son développement universel  $\widehat{S}$ . Cependant, comme nous l'avons déjà suggéré dans le premier chapitre, on doit pouvoir dupliquer les successeurs de tout état d'un système de transitions sans changer la sémantique du comportement qu'il code. Cette intuition conduit à la notion d'expansion d'un système.

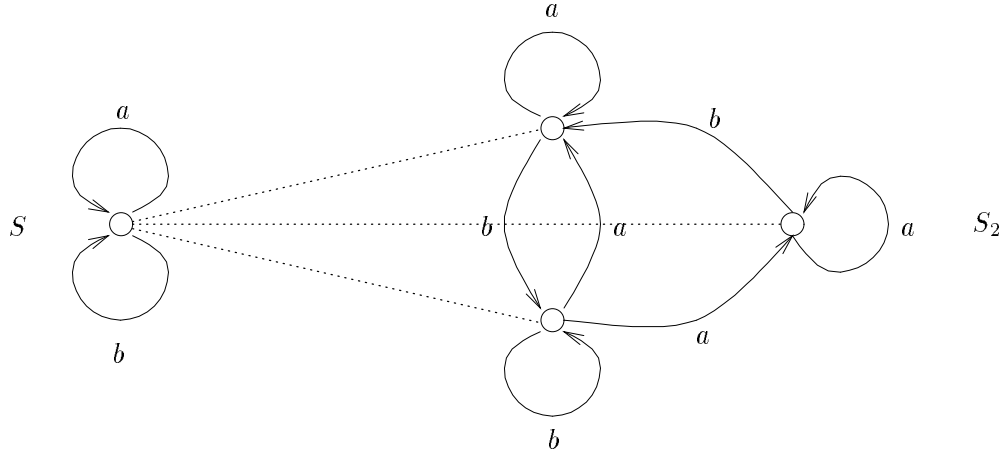
**2.1.3.1 Définition (Expansion d'un système).** Etant donnés deux systèmes de transitions  $S_1 = \langle E_1, r_1, R_1, \rho_1 \rangle$  et  $S_2 = \langle E_2, r_2, R_2, \rho_2 \rangle$ , on dit que  $S_1$  est une *expansion* de  $S_2$  lorsqu'il existe une fonction *partielle*  $h : E_1 \rightarrow E_2$  telle que :

1.  $h$  est définie en  $r_1$  avec  $h(r_1) = r_2$ ,
2. pour tout état  $e \in E_1$  sur lequel la fonction  $h$  est définie :
  - (a)  $\rho_2(h(e)) = \rho_1(e)$ ,
  - (b) pour toute action  $a \in Act$ , la restriction de  $h$  à  $\text{Succ}_a(e)$  est une **surjection** de  $\text{Succ}_a(e)$  dans  $\text{Succ}_a(h(e))$ .

**Remarque.** La fonction  $h$  ci-dessus induit un morphisme surjectif entre les chemins de  $S_1$  et de  $S_2$  issus des racines  $r_{S_1}$  et  $r_{S_2}$ . Cette propriété n'est pas suffisante cependant pour caractériser les fonctions définies ci-dessus.

La figure 3 illustre cette notion d'expansion.

**Remarque.** Les fonctions caractérisant ci-dessus la notion d'expansion sont appelées "abstraction homomorphisms" par Castellani [9]. Dans le cadre de la logique des processus, Arnold et Dicky [6, 4] en donnent une généralisation sous le nom d'"homomorphismes saturants".

FIG. 3 -  $S_2$  est une expansion de  $S$ 

Notons aussi que si un système  $S_1$  est un développement d'un système  $S_2$ , alors  $S_1$  est aussi une expansion de  $S_2$ .

**2.1.3.2 Définition (Chemin  $\alpha$ -indexé).** Pour tout cardinal  $\alpha$ , étant donné  $N_\alpha$  un ensemble de cardinal  $\alpha$ , on appelle chemin  $\alpha$ -indexé de  $S$  un mot  $w \in E.(Act.N_\alpha.E)^*$  tel que sa projection sur  $(E + Act)^*$  est un chemin de  $S$ .

Il vient alors :

**2.1.3.3 Théorème.** Pour tout cardinal infini  $\alpha$ , pour tout système de transitions  $S$  de cardinalité au plus  $\alpha$ , il existe une expansion  $\tilde{S}$  de  $S$  telle que, pour tout système  $S_1$  de cardinalité au plus  $\alpha$ , si  $S_1$  est une expansion de  $S$  alors le système  $\tilde{S}$  est une expansion de  $S_1$ .

*Preuve.* On construit  $\tilde{S}$  comme l'arbre des chemins  $\alpha$ -indexés de  $S$ . Plus précisément, on pose :

$$\tilde{S} = \langle E_{\tilde{S}}, r_{\tilde{S}}, R_{\tilde{S}}, \rho_{\tilde{S}} \rangle$$

où :

- l'ensemble des états  $E_{\tilde{S}}$  est l'ensemble des chemins  $\alpha$ -indexés de  $S$  issus de l'état initial  $r_S$ ,
- l'état initial est  $r_{\tilde{S}} = r_S$ ,

- pour toute action  $a \in Act$ , toute paire de chemins  $u$  et  $v$ , avec  $u$  de la forme  $u_1.e$  on pose  $(u, v) \in R_{\tilde{S}}(a)$  lorsque  $v$  est de la forme  $u_1.e.a.n.f$  avec  $(e, f) \in R_S(a)$ .
- et, pour tout  $u$  de la forme  $u_1.e$ ,  $\rho_{\tilde{S}}(u) = \rho_S(e)$ .

Etant alors donné  $S_1$  une expansion de  $S$ , il est facile de voir que, en termes de développements,  $\widehat{S}_1$  est une expansion de  $\widehat{S}$ . De surcroît, étant donné  $h : E_{\widehat{S}_1} \rightarrow E_{\widehat{S}}$  une fonction démontrant cette expansion, pour tout état  $e \in E_{\widehat{S}}$  le cardinal de  $h^{-1}(e)$  est au plus  $\alpha$ . Il est alors facile de construire une fonction surjective  $h' : E_{\tilde{S}} \rightarrow E_{\widehat{S}_1}$  telle que, pour tout chemin  $\alpha$ -indexé  $w \in E_{\tilde{S}}$ ,  $h \circ h'(w) \in \widehat{S}$  est la projection de  $w$  sur  $(E_S + Act)^*$ . Cela nous prouve alors que  $\tilde{S}$  est bien une expansion de  $\widehat{S}_1$  et donc, par composition, une expansion de  $S_1$ .  $\square$

**Remarque.** Une telle expansion est unique à isomorphisme près. On dit alors que le système  $\tilde{S}$  est l'*expansion  $\alpha$ -universelle* de  $S$ . Du point de vue de l'informatique on s'intéressera essentiellement au cas dénombrable. Les expansions  $\omega$ -universelles sont aussi appelées dans la suite  *$\omega$ -expansions*.

**2.1.3.4 Définition (Milner).** Etant donné  $S_1$  et  $S_2$  deux systèmes de transitions, une relation de *simulation* entre  $S_1$  et  $S_2$  est une relation  $R \subseteq E_{S_1} \times E_{S_2}$  satisfaisant :

1.  $(r_1, r_2) \in R$ ,
2. pour toute paire  $(e_1, e_2) \in R$ ,  $\rho_{S_1}(e_1) = \rho_{S_2}(e_2)$  et, pour toute action  $a \in Act$ , s'il existe un état  $f_1 \in E_1$  tel que  $e_1 \xrightarrow{a}_{S_1} f_1$  alors il existe un état  $f_2 \in E_2$  tel que  $e_2 \xrightarrow{a}_{S_2} f_2$  avec  $(e_2, f_2) \in R$ .

Lorsqu'il existe une telle relation, on dit que  $S_2$  simule  $S_1$ . On vérifie aisément que, sur la classe des systèmes de transitions, la relation de simulation est un préordre. L'équivalence induite par ce préordre est appelée *équivalence de simulation*.

**Remarque.** Les notions de développements et d'expansions préservent la simulation au sens où, pour tous systèmes  $S_1$  et  $S_2$ , si  $S_1$  simule  $S_2$  alors tout développement (resp. toute expansion) de  $S_1$  simule tout développement (resp. toute expansion) de  $S_2$ . En particulier, la notion de simulation étant invariante par développement, on peut tout aussi bien limiter notre étude au cas des arbres de transitions.

**2.1.3.5 Définition (Park).** On dit qu'une relation de simulation  $R$  entre  $S_1$  et  $S_2$  est une relation de *bisimulation* [38] lorsque  $R^{-1}$  est aussi une relation de simulation entre  $S_2$

et  $S_1$ . Lorsqu'il existe une telle relation, on dit que les systèmes  $S_1$  et  $S_2$  sont *bisimilaires*. On vérifie aisément que, sur la classe des systèmes de transitions, cette relation est une relation d'équivalence appelée *équivalence de bisimulation*. Par extension, deux états  $e$  et  $f$  d'un même système  $S$  sont dit en bisimulation lorsqu'il existe une relation de bisimulation  $R$  sur  $E_S$  avec  $(e, f) \in R$ .

**Remarque.** Bien entendu, deux systèmes équivalents par bisimulation sont équivalents par simulation. La réciproque est fautive comme le montre la figure 4 où les systèmes  $S_1$  et  $S_2$  sont équivalents par simulation sans être en bisimulation.

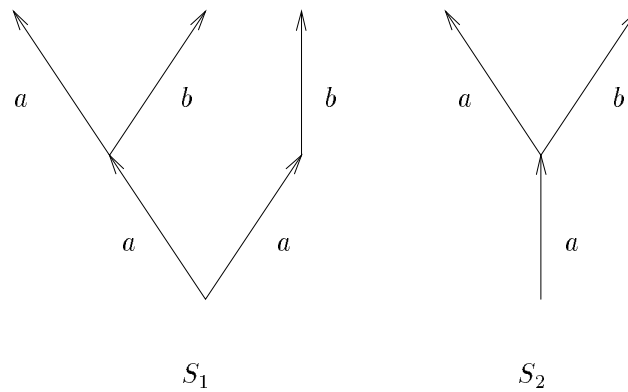


FIG. 4 - Deux systèmes équivalents par simulation sans être bisimilaires.

En fait, la notion d'expansion caractérise l'équivalence de bisimulation au sens suivant :

**2.1.3.6 Proposition.** *Deux systèmes sont en bisimulation ssi ils admettent une expansion commune.*

*Preuve.* Classique. □

**Remarque.** Dans les travaux de Castellani, Arnold et Dicky [3, 6, 9] il est prouvé que deux systèmes sont en bisimulation ssi ils sont expansions d'un même système. Bien entendu, un tel système peut être choisi "minimal" en considérant les quotients (des composantes connexes de des racines) de ces systèmes par équivalence de bisimulation. Notre approche ici, avec la notion d' $\alpha$ -expansion qui joue un rôle fondamentale dans le dernier chapitre, s'attache à définir des représentants "maximaux" de classes de systèmes équivalents par bisimulation.



Plus précisément :

**2.1.3.7 Corollaire.** *Pour tout cardinal infini  $\alpha$ , deux systèmes de cardinal au plus  $\alpha$  sont en bisimulation ssi il admettent la même  $\alpha$ -expansion.*

## 2.2 Caractérisation logique de la bisimulation

Dans cette partie, nous montrons que l'équivalence de bisimulation est la plus fine des équivalences comportementales possibles au sens où, sur les arbres de transitions à branchements finis, elle peut être caractérisée par la logique du premier ordre sans égalité. On montre aussi comment l'équivalence de bisimulation caractérise la logique modale paramétrée dont on rappelle la définition.

### 2.2.1 Logique du premier ordre sans égalité

On rappelle que *LPOSEG* désigne l'ensemble des formules de la logique du premier ordre sans prédicat d'égalité.

**2.2.1.1 Proposition.** *Pour tous arbres de transitions  $S_1$  et  $S_2$  à branchement fini, les deux propositions suivantes sont équivalentes :*

1. *les arbres  $S_1$  et  $S_2$  sont bisimilaires,*
2. *pour toute formule close  $\varphi \in \text{LPOSEG}$  on a  $S_1 \models \varphi$  ssi  $S_2 \models \varphi$ .*

*Preuve.* Il est bien connu [22], que si les systèmes  $S_1$  et  $S_2$  ne sont pas en bisimulation, alors il existe une formule de logique du premier ordre sans égalité qui les distingue. Réciproquement, il nous faut montrer qu'il n'existe pas de formule de la logique du premier ordre sans égalité qui puisse distinguer deux arbres de transitions équivalents par bisimulation.

Etant donné  $\varphi(\mathbf{x}) \in \text{LPOSEG}$  une formule ayant au plus  $n$  variables libres parmi  $\mathbf{x} = x_1, x_2, \dots, x_n$ , on rappelle qu'un modèle de  $\varphi(\mathbf{x})$  est un système de transition  $S$  associé à un vecteur d'états  $\mathbf{e} \in E_S^n$ . Il apparaît alors que pour tout  $n \in \mathbb{N}$ , deux modèles  $\langle S, \mathbf{e} \rangle$  et  $\langle S', \mathbf{e}' \rangle$  pour  $\mathbf{e} \in E_S^n$  et  $\mathbf{e}' \in E_{S'}^n$ , satisfont les mêmes formules sans quantificateurs ayant au plus  $n$  variables libres ssi la relation  $\{(r_{S_1}, r_{S_2})\} \cup \{(e_i, e'_i)\}_{i \in \{1, \dots, n\}}$  est une relation de bisimulation entre les sous-systèmes de  $S_1$  et  $S_2$  induits par les  $n + 1$  états  $r_{S_1}, e_1, \dots, e_n$  (resp. les  $n + 1$  états  $r_{S_2}, e'_1, \dots, e'_n$ ). Une induction sur le nombre de quantificateurs et les jeux de Ehrenfeucht et Fraïssé tels qu'on les présente annexe A.2 nous permettent

alors de conclure. □

**Remarque.** La restriction aux arbres est essentielle comme l'illustre la figure 5 où le système  $S$  est un modèle de  $R_a(r, r)$  alors que son développement  $\widehat{S}$  ne satisfait pas  $R_a(r, r)$ . Ce phénomène s'explique intuitivement par le fait que la formule  $R_a(r, r)$  contient implicitement le prédicat d'égalité, les formules  $R_a(r, r)$  et  $\exists x R_a(r, x) \wedge r = x$  étant équivalentes.

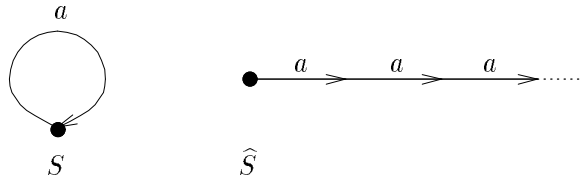


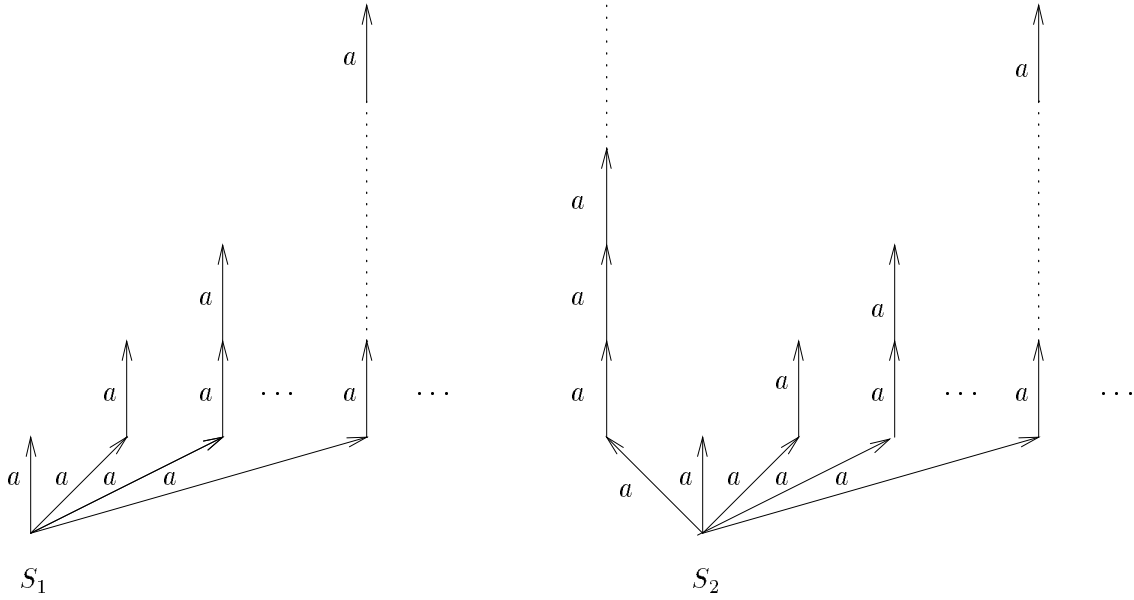
FIG. 5 -  $S \models R_a(r, r)$  alors que  $\widehat{S} \not\models R_a(r, r)$ .

La restriction aux arbres à branchement fini est tout aussi essentielle. En notant, pour tout  $n \in \mathbb{N}$ ,  $L_n(a)$  l'arbre de transitions étiquetées par la seule action  $a$  qui est formé d'une seule branche de  $n + 1$  états, en notant aussi  $L_\infty(a)$  l'arbre de transitions étiquetées par l'action  $a$  qui est formé d'une seule branche infinie, on définit, comme l'illustre la figure 6, les systèmes  $S_1$  et  $S_2$  par  $S_1 \xrightarrow{b} S$  (resp.  $S_2 \xrightarrow{b} S$ ) ssi  $b = a$  et  $S \in \{L_n(a) \mid n \in \mathbb{N}\}$  (resp.  $S \in \{L_n(a) \mid n \in \mathbb{N}\} \cup \{L_\infty(a)\}$ ). Il vient :

**2.2.1.2 Proposition.** *Les arbres  $S_1$  et  $S_2$  ne sont pas bisimilaires, alors que pour toute formule  $\varphi \in LPO$  (à fortiori avec LPOSEG) on a équivalence de  $S_1 \models \varphi$  et  $S_2 \models \varphi$ .*

*Preuve.* On vérifie tout d'abord que lorsque deux systèmes de transitions sont équivalents par bisimulation les ensembles des mots finis et infinis portés par les chemins issus de leur racine sont identiques. Il apparaît donc que les systèmes  $S_1$  et  $S_2$  ne sont pas en bisimulation.

Par ailleurs, on montre facilement à l'aide des jeux de Ehrenfeucht et Fraïssé qu'il n'existe pas de formule de logique du premier ordre qui sépare l'arbre  $S_1$  de l'arbre  $S_2$ . En effet, ces jeux s'arrêtent nécessairement après un nombre fini de coups, un nombre fini de sommets sur les arbres  $S_1$  et  $S_2$  ayant été sélectionnés. De fait, l'existence d'une branche infinie n'est pas exprimable en logique du premier ordre. □

FIG. 6 - Deux systèmes non bisimilaires et non séparable par  $LPO$ .

**Remarque.** Pour tout cardinal  $\alpha$ , il faut en fait, pour caractériser l'équivalence de bisimulation sur les arbres de transitions de cardinalité  $\alpha$ , étendre la logique  $LPO$  à l'aide de conjonctions et disjonctions de même cardinalité [2].

### 2.2.2 Logique modale paramétrée

Dans cette partie, on rappelle la définition de la logique modale paramétrée, une extension, pour traiter le cas des étiquettes de transitions, de la logique modale  $K$ .

**2.2.2.1 Définition.** On définit l'ensemble des *formules modales paramétrées*  $LMod$  (dont l'introduction en informatique est due à Hennessy et Milner [23]) comme l'ensemble des formules  $\alpha$  de la forme

$$\alpha ::= \top \mid \perp \mid p \mid \alpha_1 \wedge \alpha_2 \mid \alpha_1 \vee \alpha_2 \mid \langle a \rangle \alpha \mid [a] \alpha \mid \neg \alpha$$

pour :

1. toute action  $a \in Act$ ,
2. toute constante  $p \in Prop$ ,

3. toutes formules  $\alpha$ ,  $\alpha_1$  et  $\alpha_2$  de  $LMod$ .

**2.2.2.2 Définition.** Etant donnée une formule  $\alpha$  de logique modale paramétrée et un système  $S$ , on définit l'ensemble  $\alpha^S$  des états de  $S$  *satisfaisant la formule  $\alpha$*  par l'induction suivante :

1.  $\top^S = E_S$ ,  $\perp^S = \emptyset$ ,  $p^S = \{e \in E_S \mid p \in \rho_S(e)\}$ ,
2.  $(\neg\alpha)^S = E_S - \alpha^S$ ,
3.  $(\alpha \wedge \beta)^S = \alpha^S \cap \beta^S$ ,
4.  $(\alpha \vee \beta)^S = \alpha^S \cup \beta^S$ ,
5.  $(\langle a \rangle \alpha)^S = \{e \in E_S \mid \exists f \in \alpha^S, (e, f) \in R_S(a)\}$ ,
6.  $([a] \alpha)^S = \{e \in E_S \mid \forall f \in E_S, (e, f) \in R_S(a) \Rightarrow f \in \alpha^S\}$ .

**2.2.2.3 Proposition.** *Pour toute formule  $\alpha \in LMod$ , il existe une formule  $\varphi_\alpha \in LPOSEG$  telle que, pour tout système  $S$ ,  $r_S \in \alpha^S$  ssi  $S \models \varphi_\alpha$ .*

*Preuve.* On définit  $\varphi_\alpha$  par induction sur la structure de  $\alpha$  de la façon suivante :

1.  $\varphi_\top = \top$ ,  $\varphi_\perp = \perp$ ,
2.  $\varphi_{\neg\alpha} = \neg\varphi_\alpha$ ,
3.  $\varphi_{\alpha \wedge \beta} = \varphi_\alpha \wedge \varphi_\beta$  et  $\varphi_{\alpha \vee \beta} = \varphi_\alpha \vee \varphi_\beta$ ,
4. et, enfin,  $\varphi_{\langle a \rangle \alpha} = \exists z R_a(r, z) \wedge \varphi_\alpha[z/r]$  et  $\varphi_{[a] \alpha} = \forall z \neg R_a(r, z) \vee \varphi_\alpha[z/r]$  pour  $z$  une variable du premier ordre n'apparaissant pas dans  $\varphi_\alpha$ , en notant  $\varphi_\alpha[z/r]$  la formule obtenue à partir de  $\varphi_\alpha$  en remplaçant toute occurrence de la constante  $r$  par la variable  $z$ .

L'équivalence sémantique des formules  $\alpha$  et  $\varphi_\alpha$  se prouve alors facilement par induction.

□

**Remarque.** La réciproque est fautive comme l'illustre, par exemple, la formule  $\exists x \exists y R_a(x, y)$ . Intuitivement, en logique modale paramétrée, les propriétés des états sont décrites “de proches en proches”, à partir de la racine des systèmes de transitions.

**Notation.** Par abus, on notera  $S \models \alpha$  la propriété  $r_S \in \alpha^S$ .

**Remarque.** On note que les opérateurs de modalités  $\langle a \rangle$  et  $[a]$  sont duaux l'un de l'autre. Plus précisément, pour toute formule  $\alpha \in LMod$ , les formules  $\neg \langle a \rangle \neg \alpha$  et  $[a] \alpha$  sont

équivalentes sur la classe des systèmes de transitions.

**2.2.2.4 Définition.** [Van Benthem] On dit qu'une formule close  $\varphi \in LPO$  *préserve la bisimulation* lorsqu'il n'existe pas deux systèmes  $S_1$  et  $S_2$  bisimilaires tels que  $S_1 \models \varphi$  alors que  $S_2 \models \neg\varphi$  (de façon équivalente lorsque, pour tout système  $S$ , toute expansion  $S_1$  de  $S$ ,  $S \models \varphi$  ssi  $S_1 \models \varphi$ ).

Il vient :

**2.2.2.5 Théorème (Van Benthem).** *Une formule close  $\varphi \in LPO$  préserve la bisimulation ssi elle est sémantiquement équivalente à une formule de la logique modale paramétrée.*

*Preuve.* Cf. l'ouvrage de Van Benthem [7].  $\square$

## 2.3 Forme normale disjonctive pour $LMod$

Dans cette partie, on applique l'étude faite de la logique du premier ordre sans égalité sur les structures plates au cas de la logique modale paramétrée sur les systèmes de transitions. Cette approche nous conduit à la notion de forme normale disjonctive pour toute formule de la logique modale paramétrée et à un nouveau combinateur modal qui permet de mieux comprendre, dans cette fin de chapitre, le pouvoir d'expression de la logique modale paramétrée, et dans toute la suite de cette thèse, le pouvoir d'expression du  $\mu$ -calcul modal.

Une version préliminaire de ce qui suit fut déjà présentée par l'auteur [25].

### 2.3.1 Un connecteur pour deux modalités

En s'inspirant de la proposition 1.2.2.3 on définit le combinateur  $a \rightarrow A$  de la façon suivante :

**2.3.1.1 Définition.** Etant donné un ensemble *fini*  $A$  de formules de  $LMod$ , étant donné une action  $a \in Act$ , on définit la formule  $a \rightarrow A$  par :

$$a \rightarrow A \stackrel{\text{def}}{=} \bigwedge_{\alpha \in A} \langle a \rangle \alpha \wedge [a] \bigvee_{\alpha \in A} \alpha$$

On vérifie que pour tout système de transitions  $S$ , les propositions suivantes sont équivalentes :

1.  $S \models a \rightarrow A$ ,

2. les deux conditions suivantes sont satisfaites :

- (a) pour tout  $S'$  tel que  $S \xrightarrow{a} S'$  il existe  $\alpha \in A$  telle que  $S' \models \alpha$ ,
- (b) pour tout  $\alpha \in A$  il existe  $S'$ , tel que  $S \xrightarrow{a} S'$  avec  $S' \models \alpha$ .

**2.3.1.2 Proposition.** *De façon analogue au cas des structures discrètes, les modalités paramétrées peuvent être codées de la façon suivante :*

$$\langle a \rangle \alpha \stackrel{\text{def}}{=} a \rightarrow \{\alpha, \top\}$$

et

$$[a] \alpha \stackrel{\text{def}}{=} (a \rightarrow \{\alpha\}) \vee (a \rightarrow \emptyset)$$

*Preuve.* Immédiate. □

Dans la suite, on utilisera ce connecteur à la place des connecteurs modaux.

## 2.3.2 Formules modales disjonctives

**2.3.2.1 Définition.** L'ensemble des *formules modales disjonctives* (de la logique modale paramétrée) est inductivement défini comme :

1. les constantes  $\perp$  et  $\top$ ,
2. toutes disjonctions de formules disjonctives,
3. toutes conjonctions de la forme  $\bigwedge_{i \in \mathcal{I}} (a_i \rightarrow A_i) \bigwedge_{j \in \mathcal{J}} l_j$ , où :
  - (a) les  $a_i$  sont des actions de *Act* deux à deux distinctes,
  - (b) les  $A_i$  sont des ensembles finis de formules disjonctives,
  - (c) les  $l_j$  sont des *littéraux*, c'est à dire des constantes propositionnelles de *Prop* ou des négations de constantes propositionnelles,
  - (d)  $\mathcal{I}$  et  $\mathcal{J}$  sont des ensembles finis d'indices.

Il vient :

**2.3.2.2 Proposition.** *Pour toute formule  $\alpha$  de la logique modale paramétrée il existe une formule disjonctive  $\hat{\alpha}$  qui est équivalente à  $\alpha$ .*

*Preuve.* Par induction sur la structure de  $\alpha$ . On suppose que toutes les modalités ont été "encodées" à l'aide du constructeur  $a \rightarrow A$  selon la proposition 2.3.1.2. Modulo les règles

classiques de logique propositionnelle, on fait alors “descendre” les conjonctions à l’aide de la règle de réécriture :

$$(a \rightarrow A) \wedge (a \rightarrow B) \rightsquigarrow a \rightarrow \{\alpha \wedge \bigvee B \mid \alpha \in A\} \cup \{\bigvee A \wedge \beta \mid \beta \in B\}$$

pour toute action  $\alpha$ , tous ensembles finis de formules  $A$  et  $B$ ; une règle dont on vérifie sans peine qu’elle préserve la sémantique des formules.  $\square$

La pertinence de la notion de formule disjonctive, et l’usage restreint de la conjonction qu’elle induit, apparaît lorsqu’on s’intéresse au problème de la satisfaisabilité. Etant donné une formule, on souhaite décider si elle admet un modèle. En général, dans le cas de la logique modale paramétrée, contenant déjà la logique propositionnelle, ce problème est au moins  $NP$ -difficile.

**2.3.2.3 Théorème.** *La satisfaisabilité de toute formule disjonctive  $\alpha \in LMod$  est décidable en temps linéaire en la taille de la formule  $\alpha$ .*

*Preuve.* Pour illustrer les propriétés des formules disjonctives on donne la construction explicite d’un modèle de  $\alpha$  lorsque c’est possible. Cette construction procède par induction sur la structure de  $\alpha$ .

1. Si  $\alpha$  est une constante  $\top$  ou  $\perp$ , on conclut aisément.
2. Si  $\alpha$  est de la forme  $\beta_1 \vee \beta_2$ , la construction d’un modèle pour  $\alpha$  se fait soit en construisant un modèle pour  $\beta_1$ , soit un modèle pour  $\beta_2$ ,
3. Si  $\alpha$  est de la forme  $\bigwedge_{i \in \mathcal{I}} (a_i \rightarrow A_i) \bigwedge_{j \in \mathcal{J}} l_j$ , de deux choses l’une :
  - (a) l’une des formules apparaissant dans l’un des  $A_i$  n’est pas satisfaisable, ou bien l’ensemble des littéraux  $\{l_j\}_{j \in \mathcal{J}}$  est inconsistant, i.e. il existe une proposition  $p \in Prop$  telle que  $p$  et  $\neg p$  apparaissent tous deux dans cet ensemble, dans ce cas, la formule  $\alpha$  n’est pas satisfaisable,
  - (b) ou bien, dans le cas contraire, pour toute formule  $\beta$  appartenant à l’un des ensembles  $A_i$  on dispose d’un modèle  $S_\beta$  de la formule  $\beta$ . La construction d’un modèle  $S$  pour  $\alpha$  se fait alors de la façon suivante : pour toute action  $a_i$ , pour toute formule  $\beta \in A_i$ , on construit un et un seul  $a_i$ -successeur de  $S_\beta$  isomorphe à  $S_\beta$ , i.e.  $S \xrightarrow{a_i} S_\beta$  et on pose

$$\rho_S(rS) = \{p \in Prop \mid \exists j \in \mathcal{J}, p = l_j\}$$

On vérifie que  $S$  est bien un modèle de  $\alpha$ .

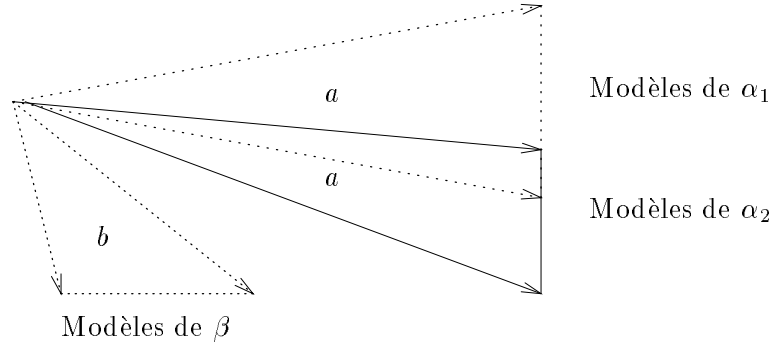


FIG. 7 - Modèles de  $a \rightarrow \{\alpha_1, \alpha_2\} \wedge b \rightarrow \{\beta\}$  avec  $Act = \{a, b\}$ .

□

**Remarque.** Ce résultat provient du fait, comme l'illustre la figure 7, que la sémantique des formules disjonctives peut se définir sans intersection. En effet, les seules conjonctions autorisées apparaissent dans les sous-formules de la forme :

$$\bigwedge_{a \in Act_1} a \rightarrow A_a$$

pour  $Act_1$  une partie finie de  $Act$ . Or les modèles d'une telle formule sont exactement ceux admettant :

1. pour toute action  $a \in Act_1$ , un ensemble de  $a$ -successeurs  $\mathcal{S}$  admettant un recouvrement  $\{\mathcal{S}_\alpha | \alpha \in A_a\}$  de telle sorte que pour chaque  $\alpha \in A_a$ , l'ensemble  $\mathcal{S}_\alpha$ , non vide, ne contient que des modèles de  $\alpha$ ,
2. pour toute action  $a \notin Act_1$ , un ensemble arbitraire de  $a$ -successeurs.

Ainsi, les conjonctions apparaissant implicitement dans le connecteur  $a \rightarrow A$  ont une sémantique qui s'exprime à l'aide d'une union. De même, la conjonction sur les actions a une sémantique qui s'exprime à l'aide d'une union, a posteriori disjointe.



## Chapitre 3

# Un calcul de points fixes

Dans ce chapitre, on présente le  $\mu$ -calcul modal : une extension par point fixe de la logique modale paramétrée généralement attribuée à Kozen [30].

Intuitivement, on désire associer à toute formule du  $\mu$ -calcul une sorte d'automate lisant les états de tout système de transitions de telle sorte que l'état d'un système de transitions est accepté par cet automate ssi le système induit par cet état est un modèle de la formule.

Par exemple, pour savoir si la formule  $\alpha \vee \beta$  est satisfaite par un état donné, il suffit de vérifier que la formule  $\alpha$  ou la formule  $\beta$  est satisfaite par cet état; c'est du non-déterminisme au sens de la théorie des automates. Pour savoir si la formule de point fixe  $\mu X.\alpha(X)$  est satisfaite dans un certain état on teste la formule  $\alpha[\mu X.\alpha(X)/X]$ . Pour savoir si la formule  $\alpha \wedge \beta$  est satisfaite dans un état, on vérifie en parallèle que chaque formule  $\alpha$  et  $\beta$  est satisfaite par cet état; c'est un branchement universel à la manière des automates alternants [34] sur les arbres binaires.

Cette intuition nous conduit tout d'abord à la caractérisation de la sémantique des formules du  $\mu$ -calcul à l'aide de la notion de *tableaux* de formules : sortes d'automates totalement développés qui nous ont été présentés par Walukiewicz. Pour ce faire, on présente en détail la notion de calcul de points fixes par approximations successives qui, avec la notion de signature [43], se révèle être un outil puissant d'étude de la sémantique des calculs de points fixes. Cette étude montre en particulier que le critère infinitaire de parité pour les automates sur les objets infinis, appelé aussi condition de chaîne [31], coïncide très précisément avec la notion d'alternance de points fixes.

On définit alors la notion d'*automates modaux alternants* dont on prouve qu'elle capture, d'un point de vue sémantique, le pouvoir d'expression du  $\mu$ -calcul. Ces automates modaux permettent une description compacte de tout système d'équations modales de

plus petits et plus grands points fixes, évitant ainsi la nécessaire “forme arborescente” des formules du  $\mu$ -calcul.

### 3.1 Le $\mu$ -calcul modal

Dans cette section, on définit le  $\mu$ -calcul de Kozen [30] : une extension par plus petits et plus grands points fixes de la logique modale paramétrée. Contrairement à l’approche usuelle, on évite cependant d’utiliser les modalités  $\langle a \rangle$  et  $[a]$  en utilisant plutôt le combinateur  $a \rightarrow A$  défini en 2.3.1.1.

Dans ce qui suit,  $Const = \{p, q, \dots\}$  désigne un ensemble de constantes propositionnelles, et  $Var = \{X, Y, Z, \dots\}$  désigne un ensemble de variables d’ensembles, en supposant que l’ensemble  $Prop$  des symboles propositionnels est égal à la réunion de  $Const$  et  $Var$ .

On appelle alors *littéral* tout symbole  $p$  ou  $X \in Prop$  ou sa négation  $\neg p$  ou  $\neg X$ .

#### 3.1.1 Un calcul de points fixes

**3.1.1.1 Définition.** Le  $\mu$ -calcul modal est défini comme le plus petit ensemble des formules  $\alpha$  de la forme :

$$\alpha = l \mid \beta_1 \wedge \beta_2 \mid \beta_1 \vee \beta_2 \mid a \rightarrow A \mid \mu X. \beta(X) \mid \nu X. \beta(X)$$

pour :

- tout littéral  $l$ ,
- toutes formules  $\beta, \beta_1, \beta_2$  et tout ensemble fini  $A$  de formules du  $\mu$ -calcul,
- toute action  $a \in Act$ ,
- toute variable  $X \in Var$ .

L’occurrence d’une variable  $X$  dans une formule  $\alpha$  est dite *liée* lorsqu’elle apparaît dans une sous-formule de la forme  $\sigma X. \beta$  pour  $\sigma$  valant  $\mu$  ou  $\nu$ . Dans le cas contraire, une telle occurrence de variable est dite *libre*. Une formule n’admettant aucune occurrence de variable libre est dite *close*.

**3.1.1.2 Définition.** Dans une formule  $\alpha$ , une occurrence de variable est dite *positive* lorsqu’elle n’apparaît pas sous la forme  $\neg X$ . Une formule  $\alpha$  est alors dite positive lorsque, pour toute sous-formule de  $\alpha$  de la forme  $\sigma X. \beta$ , toutes les occurrences libres de  $X$  apparaissant dans  $\beta$  sont positives.

Dans la suite, on notera  $\mathbf{L}_\mu$  l'ensemble des formules positives du  $\mu$ -calcul.

**3.1.1.3 Définition.** Pour toute formule positive  $\alpha$  du  $\mu$ -calcul, tout système de transitions  $S$ , on définit l'ensemble  $\alpha^S$  des états du système  $S$  *satisfaisant* la formule  $\alpha$  à l'aide de l'ensemble des règles d'inférences de la logique modale paramétrée, définition 2.2.2.2, augmenté, pour le traitement des opérateurs de point fixe, des règles suivantes :

1.  $(\mu X.\alpha(X))^S = \bigcap \{E' \subseteq E_S \mid \alpha^{S[E'/X]} \subseteq E'\}$ ,
2.  $(\nu X.\alpha(X))^S = \bigcup \{E' \subseteq E_S \mid E' \subseteq \alpha^{S[E'/X]}\}$ ,

où  $S[E'/X]$  désigne le système de transitions obtenu à partir de  $S$  en changeant la fonction  $\rho_S$  d'étiquetage des états en la fonction  $\rho_{S[E'/X]}$  définie pour tout  $e \in E_S$  par :

$$\rho_{S[E'/X]}(e) = \begin{cases} \rho_S(e) - \{X\} & \text{si } e \notin E', \\ \rho_S(e) \cup \{X\} & \text{si } e \in E'. \end{cases}$$

**Remarque.** La restriction à des occurrences positives dans  $\beta(X)$  nous assure que la fonction  $E' \mapsto (\beta(X))^{S[E'/X]}$  est croissante sur l'ensemble des parties de  $E_S$ . Le théorème de point fixe dit “de Knaster-Tarski” s'applique. On vérifie alors que  $(\mu X.\alpha(X))^S$  et  $(\nu X.\alpha(X))^S$  sont respectivement le plus petit et le plus grand point fixe de cette fonction. En particulier, la formule  $\nu X.X$  est satisfaite par tout état, la formule  $\mu X.X$  n'est satisfaite par aucun.

**3.1.1.4 Proposition.** *Pour toute formule positive  $\alpha$ , il existe une formule  $\varphi_\alpha \in \text{LMSO}$  telle que, pour tout système  $S$ ,  $r_S \in (\alpha)^S$  ssi  $S \models \varphi_\alpha$ .*

*Preuve.* Par induction sur la structure de  $\alpha$  selon les règles suivantes :

1.  $\varphi_X = X(r)$  et  $\varphi_p = p(r)$ ,
2.  $\varphi_{\neg\alpha} = \neg\varphi_\alpha$ ,
3.  $\varphi_{\alpha_1 \wedge \alpha_2} = \varphi_{\alpha_1} \wedge \varphi_{\alpha_2}$ ,
4.  $\varphi_{\alpha_1 \vee \alpha_2} = \varphi_{\alpha_1} \vee \varphi_{\alpha_2}$ ,
5.  $\varphi_{a \rightarrow A} = \bigwedge_{\alpha \in A} \exists x R_a(r, x) \varphi_\alpha[x/r] \wedge \forall x \neg R_a(r, x) \vee \bigvee_{\alpha \in A} \varphi_\alpha[x/r]$
6.  $\varphi_{\nu X.\alpha(X)} = \exists X ((\forall y (X(y) \Rightarrow \varphi_{\alpha(X)}[y/r])) \wedge X(r))$ ,
7.  $\varphi_{\mu X.\alpha(X)} = \forall X ((\forall y \varphi_{\alpha(X)}[y/r] \Rightarrow X(y)) \Rightarrow X(r))$ ,

On vérifie facilement que ces règles ne font que traduire la sémantique des  $\mu$ -formules positives.  $\square$

**Notation.** Par abus, on notera  $S \models \alpha$  le fait que  $r_S \in \alpha^S$ . La notion de préordre sémantique s'étend aux formules positives du  $\mu$ -calcul modal. On note alors  $\alpha \preceq \beta$  le fait que pour tout système de transitions  $S$ , si  $S \models \beta$  alors  $S \models \alpha$ .

**Remarque.** On peut toujours définir la négation de toute  $\mu$ -formule  $\alpha$  en posant, comme dans le cas de la logique modale paramétrée,  $(\neg\alpha)^S = E_S - \alpha^S$ . C'est en fait sans pertes de généralité que nous nous sommes restreint aux négations ne portant que sur des occurrences libres de variables libres ou des constantes. Les règles de réécriture suivantes, qui préservent la sémantique des formules, nous permettent en effet de nous y ramener :

1.  $\neg\neg\alpha \rightsquigarrow \alpha$ ,
2.  $\neg\mu X.\alpha \rightsquigarrow \nu X.\neg\alpha[\neg X/X]$ ,
3.  $\neg\nu X.\alpha \rightsquigarrow \mu X.\neg\alpha[\neg X/X]$ ,
4.  $\neg(\alpha_1 \wedge \alpha_2) \rightsquigarrow \neg\alpha_1 \vee \neg\alpha_2$ ,
5.  $\neg(\alpha_1 \vee \alpha_2) \rightsquigarrow \neg\alpha_1 \wedge \neg\alpha_2$ ,
6.  $\neg a \rightarrow A \rightsquigarrow a \rightarrow \{\bigwedge_{\alpha \in A} \neg\alpha, \top\} \vee \bigvee_{\alpha \in A} (a \rightarrow \{\neg\alpha\} \vee a \rightarrow \emptyset)$  où  $\top$  désigne ici la formule  $\nu X.X$ .

Notons aussi qu'une variable libre a le même comportement qu'une constante propositionnelle. Dans la suite, on pourra donc, sans perte de généralité, restreindre notre étude aux formules closes.

En terme de pouvoir de séparation, on vérifie, par exemple en codant les points-fixes à l'aide de conjonctions et disjonctions infinies, que la sémantique des formules est close par bisimulation, i.e. pour toute formule  $\alpha$ , tous systèmes  $S_1$  et  $S_2$  bisimilaires, on a équivalence de  $S_1 \models \alpha$  et  $S_2 \models \alpha$ .

La proposition suivante montre cependant que le pouvoir de séparation du  $\mu$ -calcul modal est supérieur à celui de la logique du premier ordre sans égalité :

**3.1.1.5 Proposition.** *Les systèmes  $S_1$  et  $S_2$  de la proposition 2.2.1.2 sont séparables par la formule  $\alpha = \nu X.a \rightarrow \{X, \top\}$ , i.e.  $S_1 \not\models \alpha$  alors que  $S_2 \models \alpha$ . Ces deux systèmes étaient non séparables par LPOSEG.*

*Preuve.* La formule  $\nu X.a \rightarrow X, \top$  caractérise en effet les systèmes de transitions ayant, à partir de leur racine, une branche infinie (ou bien un cycle) de transitions étiquetées par l'action “a”.  $\square$

### 3.1.2 Formules bien nommées

Dans les preuves portant sur le  $\mu$ -calcul, il est souvent nécessaire de distinguer les occurrences de variables liées par des opérateurs de point fixe différents. La définition suivante et la proposition qui l’accompagne facilitent cette distinction.

**3.1.2.1 Définition.** Une formule  $\alpha$  du  $\mu$ -calcul est dite *bien nommée*, ou bien à *variables séparées*, lorsque, pour toute variable  $X$  :

1. il existe au plus un sous-terme de  $\alpha$  de la forme  $\sigma X.\beta$ ,
2. s’il existe un sous-terme de  $\alpha$  de la forme  $\sigma X.\beta$  alors aucune occurrence de la variable  $X$  n’est libre dans  $\alpha$ .

Dans une formule bien nommée  $\alpha$ , tout sous-terme de la forme  $\sigma X.\beta$  est appelé la *définition de la variable  $X$*  dans  $\alpha$  et on le note  $\mathcal{D}_\alpha(X)$ . La fonction (partielle)  $\mathcal{D}_\alpha$  associant à toute variable liée sa définition dans  $\alpha$  est appelée *fonction de liaison* de  $\alpha$ .

Dans la suite, on appelle  $\nu$ -variable (resp.  $\mu$ -variable) de la formule  $\alpha$  toute variable  $X$  liée dans  $\alpha$  avec  $\mathcal{D}_\alpha(X)$  de la forme  $\nu X.\beta$  (resp.  $\mu X.\beta$ ).

**3.1.2.2 Proposition.** *Pour toute formule  $\alpha$  du  $\mu$ -calcul il existe une formule  $\beta$  bien nommée équivalente à  $\alpha$ .*

*Preuve.* Par induction sur la structure de  $\alpha$  en utilisant le fait que, pour toute formule  $\beta$ , si toute occurrence de  $X$  dans  $\beta$  est libre et si  $Y$  n’apparaît pas dans  $\beta$  les formules  $\sigma X.\beta$  et  $\sigma Y.\beta[Y/X]$  ( $\sigma$  désignant  $\mu$  ou  $\nu$ ) sont équivalentes. Cette opération est appelée un renommage consistant de la variable  $X$  en la variable  $Y$ .  $\square$

Dans la suite, on se place systématiquement dans le cas de formules bien nommées. En particulier, l’opération de substitution pour construire la formule  $\alpha[\beta/X]$  est implicitement accompagnée d’un renommage consistant des variables liées de  $\alpha$  ou  $\beta$  de telle sorte que la formule ainsi obtenue soit bien nommée.

**3.1.2.3 Définition (Ordre de dépendance).** Pour toute formule (bien nommée)  $\alpha$ , on définit l’*ordre de dépendance* sur l’ensemble des variables liées comme la plus petite

relation d'ordre  $\leq_\alpha$  telle que,

- pour toutes variables  $X$  et  $Y$  liées dans  $\alpha$ , si la variable  $X$  admet une occurrence libre dans  $\mathcal{D}_\alpha(Y)$  alors  $X \leq_\alpha Y$ .

Lorsque  $X \leq_\alpha Y$  on dira que la variable  $Y$  dépend de la variable  $X$  dans  $\alpha$ .

**Remarque.** L'ordre de dépendance est plus faible que l'ordre considéré généralement sur les variables liées apparaissant dans une formule. En particulier, les variables  $X$  et  $Y$  dans la formule  $\mu X.\langle a \rangle(X \vee \nu Y.[a]Y)$  sont indépendantes.

De cet ordre de dépendance sur les variables liées d'une formule, on extrait la notion d'indice d'une variable.

**3.1.2.4 Définition.** Pour toute formule  $\alpha$ , toute variable liée  $X$  apparaissant dans  $\alpha$ , on définit inductivement le  $\mu$ -indice  $\pi_\alpha^\mu(X)$  (resp. le  $\nu$ -indice  $\pi_\alpha^\nu(X)$ ) de la variable  $X$  dans  $\alpha$  par les règles suivantes.

1. Lorsque  $X$  est une  $\mu$ -variable, s'il n'existe pas de  $\nu$ -variable  $Y$  dans  $\alpha$  dont  $X$  dépend, i.e. telle que  $Y \leq_\alpha X$ , alors  $\pi_\alpha^\mu(X) = 0$  et  $\pi_\alpha^\nu(X) = 1$ , sinon  $\pi_\alpha^\mu(X)$  (resp.  $\pi_\alpha^\nu(X)$ ) est égale au maximum des  $\mu$ -indices (resp.  $\nu$ -indices) des  $\nu$ -variables dont  $X$  dépend augmenté de un.

Par exemple, dans la formule  $\alpha = \nu X.\mu Y.a \rightarrow \{X \vee Y\}$ ,  $\pi_\alpha^\mu(X) = 1$  et  $\pi_\alpha^\mu(Y) = 2$ .

2. Lorsque  $X$  est une  $\nu$ -variable, s'il n'existe pas de  $\mu$ -variable  $Y$  dans  $\alpha$  dont  $X$  dépend, i.e. telle que  $Y \leq_\alpha X$ , alors  $\pi_\alpha^\mu(X) = 1$  et  $\pi_\alpha^\nu(X) = 0$ , sinon  $\pi_\alpha^\mu(X)$  (resp.  $\pi_\alpha^\nu(X)$ ) est égale au maximum des  $\mu$ -indices (resp.  $\nu$ -indices) des  $\mu$ -variables dont  $X$  dépend augmenté de un.

Avec le même exemple  $\pi_\alpha^\nu(X) = 0$  et  $\pi_\alpha^\nu(Y) = 1$ .

Autrement dit, le  $\mu$ -indice (resp. le  $\nu$ -indice) d'une variable liée dans une formule  $\alpha$  est égal au nombre d'alternances de  $\mu$  et de  $\nu$ -variables dont  $X$  dépend, en considérant que toutes les variables dépendent d'une  $\mu$ -variable (resp. une  $\nu$ -variable) fictive.

**Remarque.** L'indice  $\pi_\alpha^\mu(X)$  (resp. l'indice  $\pi_\alpha^\nu(X)$ ) est pair ssi  $X$  est une  $\mu$ -variable (resp. une  $\nu$ -variable).

### 3.1.3 Calcul de point fixe par approximations

La notion de calcul par approximation et la notion de signature ordinale qu'elle induit, présentés succinctement dans [43], nous donnent un outil puissant pour étudier la

sémantique du  $\mu$ -calcul.

**3.1.3.1 Définition.** Pour tout ordinal  $\tau$  on définit les constructeurs de point fixe  $\nu^\tau X.\alpha$  et  $\mu^\tau X.\alpha$  dont la sémantique est inductivement définie, en usant des notations de la définition 3.1.1.3, par :

- $(\mu^0 X.\alpha)^S = \emptyset$  et  $(\nu^0 X.\alpha)^S = E_S$ ,
- $(\sigma^{\tau+1} X.\alpha)^S = \alpha^{S[(\sigma^\tau X.\alpha(X))^S/X]}$  pour  $\sigma$  valant  $\mu$  ou  $\nu$ , tout ordinal  $\tau$ ,
- $(\mu^\tau X.\alpha)^S = \bigcup_{\tau' < \tau} (\mu^{\tau'} X.\alpha)^S$  pour tout ordinal limite  $\tau$ ,
- et  $(\nu^\tau X.\alpha)^S = \bigcap_{\tau' < \tau} (\nu^{\tau'} X.\alpha)^S$  pour tout ordinal limite  $\tau$ .

Il vient :

**3.1.3.2 Lemme (Knaster-Tarski).** *Pour toute formule  $\alpha$  dans laquelle toutes les occurrences libres de la variable  $X$  sont positives, les égalités  $(\mu X.\alpha)^S = \bigcup_\tau (\mu^\tau X.\alpha)^S$  et  $(\nu X.\alpha)^S = \bigcap_\tau (\nu^\tau X.\alpha)^S$  sont satisfaites.*

Dans la suite, on étend la notion de fonction de liaison en autorisant des approximations de points fixes.

**3.1.3.3 Définition (Fonction de liaison généralisée).** Pour toute formule positive  $\gamma$ ,  $\mathcal{D}_\gamma$  étant la fonction de liaison associée à  $\gamma$ , on appelle *fonction de liaison généralisée* toute fonction  $\mathcal{D}'$  telle que, pour toute variable  $X$  liée dans  $\gamma$ , si  $\mathcal{D}'(X)$  est de la forme  $\sigma^\tau X.\alpha$  alors  $\mathcal{D}_\gamma(X) = \sigma X.\alpha$ , sinon  $\mathcal{D}'(X) = \mathcal{D}(X)$ .

**3.1.3.4 Définition (Expansion).** Pour toute formule positive  $\gamma$ , toute sous-formule  $\alpha$  de  $\gamma$  on définit l'*expansion de la formule  $\alpha$*  dans la formule  $\gamma$  via la fonction de liaison  $\mathcal{D}_\gamma$  de  $\gamma$  comme la formule :

$$\langle \alpha \rangle_\gamma = (\cdots (\alpha[\mathcal{D}_\gamma(X_n)/X_n]) \cdots) [\mathcal{D}_\gamma(X_1)/X_1]$$

où  $X_1, \dots, X_n$  sont les variables liées de  $\gamma$  apparaissant librement dans  $\alpha$  ordonnées de telle sorte que si  $X_i \leq_\gamma X_j$  alors  $i \leq j$ , les substitutions étant effectuées l'une après l'autre, dans l'ordre des indices décroissants.

**Remarque.** La notion d'expansion d'une variable permet d'associer à toute sous-formule  $\alpha$  de  $\gamma$  une formule (close lorsque  $\gamma$  est close) équivalente au sens où on peut, sans changer la sémantique de  $\gamma$ , remplacer la sous-formule  $\alpha$  dans  $\gamma$  par son expansion.

**Exemple :** Avec  $\alpha = \nu X.\mu Y.(a \rightarrow \{X\} \vee b \rightarrow \{Y\})$ , qui désigne la classe des systèmes dont les développements sont des arbres étiquetés sur  $Act = \{a, b\}$  et dont toutes les branches sont infinies et contiennent une infinité de  $a$ , on a  $X \preceq_\alpha Y$ , et donc

$$\begin{aligned} \langle Y \rangle_\alpha &= Y[\mu Y.(a \rightarrow \{X\} \vee b \rightarrow \{Y\})/Y][\alpha/X] \\ &= \mu Y.(a \rightarrow \{\alpha\} \vee b \rightarrow \{Y\}) \end{aligned}$$

Notons qu'appliquer les substitutions dans l'autre sens conduirait à la formule

$$Y[\alpha/X][\mu Y.(a \rightarrow \{X\} \vee b \rightarrow \{Y\})/Y] = \mu Y.(a \rightarrow \{X\} \vee b \rightarrow \{Y\})$$

qui n'est pas une formule close.

**Notation.** Par extension, pour toute fonction de liaison généralisée  $\mathcal{D}'$  de  $\gamma$ , pour toute sous-formule  $\alpha$  de  $\gamma$ , on note  $\langle \alpha \rangle_{\mathcal{D}'}$  l'expansion de la formule  $\alpha$  à l'aide de la fonction  $\mathcal{D}'$  au lieu de  $\mathcal{D}_\gamma$ .

### 3.1.4 Signatures

Dans la suite, on se fixe une formule  $\gamma$ , positive, et on suppose que les  $\mu$ -variables  $\{X_1, \dots, X_m\}$  et les  $\nu$ -variables  $\{Y_1, \dots, Y_m\}$  sont indicées de telle sorte que, pour tout indice  $i$  et  $j$ , si  $X_i \leq_\gamma X_j$  (resp.  $Y_i \leq_\gamma Y_j$ ) alors  $i \leq j$ , i.e. l'indexation des  $\mu$ -variable (resp. des  $\nu$ -variables) induit une extension linéaire de l'ordre dépendance sur les  $\mu$ -variables (resp. les  $\nu$ -variables).

**3.1.4.1 Définition.** Etant donnée  $s = (\tau_1, \tau_2, \dots, \tau_p)$  une séquence de  $p$  ordinaux avec  $p \leq m$  (resp.  $p \leq n$ ) on note  $\mathcal{D}_s^\mu$  (resp.  $\mathcal{D}_s^\nu$ ) la *fonction de liaison généralisée* définie par :

1. pour toute  $\mu$ -variable  $X_i$  avec  $\mathcal{D}(X_i) = \mu X_i.\alpha$ , si  $i \leq p$  alors  $\mathcal{D}_s^\mu(X_i) = \mu^{\tau_i} X_i.\alpha$  (resp.  $\mathcal{D}_s^\nu(X_i) = \mu X_i.\alpha$ ), sinon  $\mathcal{D}_s^\sigma(X_i) = \mathcal{D}(X_i)$ ,
2. pour toute  $\nu$ -variable  $Y_j$  avec  $\mathcal{D}(Y_j) = \nu Y_j.\alpha$ , si  $j \leq p$  alors  $\mathcal{D}_s^\mu(Y_j) = \nu Y_j.\alpha$  (resp.  $\mathcal{D}_s^\nu(Y_j) = \nu^{\tau_j} Y_j.\alpha$ ), sinon  $\mathcal{D}_s^\sigma(Y_j) = \mathcal{D}(Y_j)$ ,

pour  $\sigma$  dénotant  $\mu$  ou  $\nu$ .

L'ordre lexicographique sur les séquences d'ordinaux étant total et bien fondé, on définit :

**3.1.4.2 Définition (Signatures).** Pour tout système de transitions  $S$ , pour toute sous-formule  $\alpha$  de  $\gamma$  :

1. si  $\langle \alpha \rangle_{\mathcal{D}}$  est satisfaite dans un état  $e \in E_S$ , la  $\mu$ -signature  $Sig_\mu(\alpha, e)$  de la formule  $\alpha$  dans l'état  $e$  est la plus petite, selon l'ordre lexicographique, séquence d'ordinaux  $s$



de longueur  $m$  telle que :

$$S[e/r] \models \langle \alpha \rangle_{\mathcal{D}_s^\mu}$$

2. si  $\langle \alpha \rangle_{\mathcal{D}}$  n'est pas satisfaite dans un état  $e \in E_S$ , la  $\nu$ -signature  $Sig_\nu(\alpha, e)$  de la formule  $\alpha$  dans l'état  $e$  est la plus petite, selon l'ordre lexicographique, séquence d'ordinaux  $s$  de longueur  $n$  telle que :

$$S[e/r] \not\models \langle \alpha \rangle_{\mathcal{D}_s^\nu}$$

où  $S[e/r]$  désigne le système de transitions obtenu à partir de  $S$  en prenant l'état  $e$  comme racine à la place de l'état  $r_S$ .

La proposition suivante illustre les propriétés “compositionnelles” des signatures qui se révèlent fondamentales pour donner une sémantique “calculatoire” aux formules du  $\mu$ -calcul.

**3.1.4.3 Lemme.** *Pour tout système de transitions  $S$ , tout état  $s \in E_S$  tel que  $S[s/r] \models \langle \delta \rangle_{\mathcal{D}}$ , les propriétés suivantes sont satisfaites.*

1. Si  $\delta = \alpha \wedge \beta$  alors  $Sig_\mu(\alpha \wedge \beta, s) \geq \max_{lex}(Sig_\mu(\alpha, s), Sig_\mu(\beta, s))$ ,
2. Si  $\delta = \alpha \vee \beta$  alors  $Sig_\mu(\alpha \vee \beta, s) = Sig_\mu(\alpha, s)$  ou bien  $Sig_\mu(\alpha \vee \beta, s) = Sig_\mu(\beta, s)$ ,
3. Si  $\delta = l_1 \wedge \dots \wedge l_n \wedge \bigwedge_{a \in Act_1} a \rightarrow A_a$  pour  $Act_1 \subseteq Act$ , alors, pour toute action  $a \in Act_1$  :
  - (a) pour toute formule  $\alpha \in A_a$  il existe un état  $t$  tel que  $(s, t) \in R_S(a)$  avec  $Sig_\mu(\alpha, t) \leq Sig_\mu(\delta, s)$ ,
  - (b) pour tout état  $t$  tel que  $(s, t) \in R_S(a)$  il existe  $\alpha \in A_a$  telle que  $Sig_\mu(\alpha, t) \leq Sig_\mu(\delta, s)$ ,
4. Si  $\delta = \nu Y.\alpha(Y)$  alors  $Sig_\mu(\nu Y.\alpha(Y), s) = Sig_\mu(\alpha(Y), s)$ .
5. Si  $\delta = \mu X.\alpha(X)$  avec  $X = X_i$  alors les préfixes de longueur  $i-1$  de  $Sig_\mu(\mu X.\alpha(X), s)$  et  $Sig_\mu(\alpha(X), s)$  sont égaux.
6. Si  $\delta = Y$  avec  $\mathcal{D}(Y) = \nu Y.\alpha(Y)$  alors  $Sig_\mu(Y, s) = Sig_\mu(\alpha(Y), s)$ .
7. Si  $\delta = X$  avec  $\mathcal{D}(X) = \mu X.\alpha(X)$  et  $X = X_i$  alors  $Sig_\mu(\alpha(X), s) < Sig_\mu(X, s)$  et la différence est à la  $i$ -ième position.

où  $\mathcal{D}$  désigne la fonction de liaison associée à la formule  $\gamma$ , des propriétés similaires étant obtenues pour les  $\nu$ -signatures par dualité.

*Preuve.* Les cas 1, 2 sont immédiats. Le cas 3 s'obtient par généralisation des cas 1 et 2 à des conjonctions et disjonctions infinies.

Les cas 4 et 6 proviennent du fait que pour toute séquence d'ordinaux  $s$  de longueur  $m$ , les formules  $\langle\langle Y \rangle\rangle_{\mathcal{D}_s^\mu}$ ,  $\langle\langle \alpha(Y) \rangle\rangle_{\mathcal{D}_s^\mu}$  et  $\langle\langle \nu Y.\alpha(Y) \rangle\rangle_{\mathcal{D}_s^\mu}$  sont trivialement équivalentes. On a en effet  $\nu Y.\alpha(Y) \simeq \alpha[\nu Y.\alpha(Y)/Y]$ , et  $\simeq$  est préservé par les substitutions.

Cas 5 : Pour toutes séquences d'ordinaux  $s$  de longueur  $i - 1$ , les formules  $\langle\langle \alpha(X_i) \rangle\rangle_{\mathcal{D}_s^\mu}$  et  $\langle\langle \mu X_i.\alpha(X_i) \rangle\rangle_{\mathcal{D}_s^\mu}$  sont équivalentes. De plus, aucune variable  $X_j$  avec  $j > i$  ne peut apparaître libre dans  $\mathcal{D}(X_k)$  pour  $k \leq i$ . Il apparaît donc que  $Sig_\mu(\alpha(X_i), s)$  est de la forme  $s.\tau.s_1$  pour  $s$  une séquence de longueur  $i - 1$ ,  $\tau$  un ordinal et  $s_1$  une séquence d'ordinaux de longueur  $m - i$  ssi  $s_1 = 0.\dots.0$  et  $Sig_\mu(\mu X_i.\alpha(X_i), s) = s.0.0.\dots.0$ .

Cas 7 : Pour toute séquence d'ordinaux  $s$  de longueur  $i - 1$ , pour tout ordinal  $\tau$ , pour toute séquence d'ordinaux  $s_1$  et  $s_2$  de longueur  $m - i$  les formules  $\langle\langle \alpha(X_i) \rangle\rangle_{\mathcal{D}_{s.\tau.s_1}^\mu}$  et  $\langle\langle X_i \rangle\rangle_{\mathcal{D}_{s.(\tau+1).s_2}^\mu}$  sont équivalentes car, là encore, aucune variable  $X_j$  avec  $j > i$  ne peut apparaître libre dans  $\mathcal{D}(X_k)$  pour  $k \leq i$ , et les formules  $\alpha[\mu^\tau X_i.\alpha(X_i)/X_i]$  et  $\mu^{\tau+1} X_i.\alpha(X_i)$  sont équivalentes. On en déduit donc que  $Sig_\mu(\alpha(X_i), s)$  est de la forme  $s.\tau.s_1$  ssi  $Sig_\mu(X_i, s)$  est de la forme  $s.(\tau + 1).s_2$  avec  $s_1 = s_2 = 0.\dots.0$ .  $\square$

## 3.2 Une sémantique calculatoire pour le $\mu$ -calcul

Dans cette partie, nous donnons une sémantique calculatoire aux formules du  $\mu$ -calcul. Après avoir défini la notion de tableau associé à une formule du  $\mu$ -calcul nous montrons qu'un système de transitions  $S$  satisfait une formule  $\alpha$  si et seulement si pour tout tableau  $\mathcal{T}$  pour  $\alpha$  il existe un marquage adéquat de  $\mathcal{T}$  par  $S$ .

### 3.2.1 Tableaux

La notion de formule gardée permet de simplifier sensiblement les énoncés et preuves qui vont suivre.

**3.2.1.1 Définition.** Un point fixe de la forme  $\sigma X.\alpha$  est dit *gardé* lorsque toute occurrence de la variable  $X$  apparaissant dans  $\alpha$  se trouve dans une sous-formule de la forme  $a \rightarrow A$ . Une formule  $\alpha$  est dite *gardée* lorsque tout point fixe apparaissant dans  $\alpha$  est gardé.

Il vient :

**3.2.1.2 Proposition (Kozen 1983).** *Pour toute formule  $\alpha$  du  $\mu$ -calcul, il existe une formule gardée  $\beta$  équivalente à  $\alpha$ .*

*Preuve.* [30]. □

**3.2.1.3 Définition.** Etant donnée  $\gamma$  une formule bien nommée et gardée,  $\mathcal{D}_\gamma$  étant la fonction de liaison des variables liées de  $\gamma$ , on définit le système de réduction  $\mathcal{S}^\gamma$  associé à la formule  $\gamma$  de la façon suivante :

$$\begin{array}{l}
 (et) \quad \frac{\{\alpha, \beta, C\}}{\{\alpha \wedge \beta, C\}} \quad (ou) \quad \frac{\{\alpha, C\} \quad \{\beta, C\}}{\{\alpha \vee \beta, C\}} \\
 (\mu) \quad \frac{\{\alpha(X), C\}}{\{\mu X. \alpha(X), C\}} \quad (\nu) \quad \frac{\{\alpha(X), C\}}{\{\nu X. \alpha(X), C\}} \\
 (reg) \quad \frac{\{\alpha(X), C\}}{\{X, C\}} \quad \text{pour toute variable } X \text{ liée dans } \gamma \\
 \quad \quad \quad \text{avec } \mathcal{D}_\gamma(X) = \sigma X. \alpha(X) \\
 (mod) \quad \frac{\{\alpha\} \cup \{\forall B | a \rightarrow B \in \{C\}, B \neq A\}}{\{C\}} \quad \text{pour tous } a \rightarrow A \in \{C\}, \alpha \in A
 \end{array}$$

où  $\forall \emptyset$  est interprété par  $\perp$ ,  $\alpha$  et  $\beta$  étant n'importe quelles formules du  $\mu$ -calcul,  $a$  n'importe quelle action,  $A$  et  $B$  n'importe quels ensembles finis de formules et  $C$  une séquence de formules, en notant par exemple  $\{\alpha, C\}$  l'ensemble contenant la formule  $\alpha$  et toutes les formules apparaissant dans  $C$ .

Etant donné un ensemble fini de formules  $C$  que nous voulons dériver, nous cherchons à instancier une règle de telle sorte que ses conclusions (apparaissant en dessous de la règle) coïncident avec l'ensemble de formules  $C$ . Les nouveaux ensembles de formules à réduire sont alors les hypothèses (apparaissant au dessus) de la règle appliquée.

**Remarque.** La règle *(mod)* admet autant de prémisses qu'il y a de formules dans les ensembles  $A$  tels que  $a \rightarrow A \in C$ . Par exemple

$$\frac{\{\alpha_1, \alpha_3\} \quad \{\alpha_2, \alpha_3\} \quad \{\alpha_1 \vee \alpha_2, \alpha_3\} \quad \{\beta_1\} \quad \{\beta_2\}}{\{a \rightarrow \{\alpha_1, \alpha_2\}, a \rightarrow \{\alpha_3\}, b \rightarrow \{\beta_1, \beta_2\}\}}$$

est une instance de la règle *(mod)*.

**3.2.1.4 Définition.** Un *tableau* pour  $\gamma$  est une paire  $\langle T, L \rangle$ , où  $T$  est un arbre (éventuellement infini) et  $L$  est une fonction d'étiquetage tels que :

1. la racine de  $T$  est étiquetée par  $\{\gamma\}$ ,
2. les fils de tout nœud  $n$  sont créés et étiquetés selon les règles du système  $\mathcal{S}^\gamma$ , i.e. pour tout nœud  $n$ , l'ensemble de ses fils étant  $\{m_1, m_2, \dots, m_k\}$ , la règle

$$\frac{L(m_1)L(m_1) \cdots L(m_k)}{L(n)}$$

est une instance de l'une des règles du système  $\mathcal{S}'$ , la règle  $(mod)$  étant *uniquement appliquée lorsqu'aucune autre règle n'est applicable*.

**Remarque.** Considérons à nouveau notre exemple d'instanciation de la règle  $(mod)$  ci-dessus. Si un nœud  $n$  est étiqueté par la conclusion de cette règle alors il admet cinq fils étiquetés par les hypothèses correspondantes. Un nœud ainsi obtenu par réduction de l'action  $a$  sera appelé *a-nœud*. Dans notre exemple, le nœud  $n$  admet trois *a*-fils et deux *b*-fils.

Pour une formule donnée, il existe en général plusieurs tableaux. La notion de nœud modal et de nœud de choix nous permet cependant d'explicitier un peu plus les similitudes entre deux tableaux pour une même formule.

**3.2.1.5 Définition.** Pour tout tableau  $\mathcal{T} = \langle T, L \rangle$ , un nœud  $n \in T$  sera appelé :

1. *nœud modal* lorsque  $n$  est une feuille ou bien un nœud à partir duquel la règle  $(mod)$  est appliquée,
2. *nœud de choix* lorsque  $n$  est la racine de  $T$  ou bien le fils d'un nœud modal.

On dira aussi qu'un nœud  $m \in T$  est *proche* d'un nœud  $n \in T$  lorsqu'il existe un chemin dans  $T$  de  $n$  à  $m$  sans application de la règle  $(mod)$ .

Il vient :

**3.2.1.6 Proposition.** Si  $\mathcal{T}_1 = \langle T_1, L_1 \rangle$  et  $\mathcal{T}_2 = \langle T_2, L_2 \rangle$  sont deux tableaux pour  $\gamma$  alors il existe une bijection  $\mathcal{E}$  entre les nœuds modaux et les nœuds de choix de  $\mathcal{T}_1$  et  $\mathcal{T}_2$  telle que :

1. en notant  $n_0$  la racine  $\mathcal{T}_1$ ,  $\mathcal{E}(n_0)$  est la racine de  $\mathcal{T}_2$ ,
2. pour tout nœud de choix  $n \in T_1$ , pour tout nœud modal  $m \in T_1$ ,  $m$  est proche de  $n$  ssi  $\mathcal{E}(m)$  est proche de  $\mathcal{E}(n)$ ,
3. pour tout nœud modal  $n \in T_1$ , pour toute action  $a \in Act$ , pour tout nœud de choix  $m \in T_1$ ,  $m$  est un *a*-fils de  $n$  ssi  $\mathcal{E}(m)$  est un *a*-fils de  $\mathcal{E}(n)$ ,
4. pour tout nœud modal, tout nœud de choix  $n \in T_1$ ,  $L_1(n) = L_2(\mathcal{E}(n))$ .

*Preuve.* Il suffit de constater que tout nœud de choix est caractérisé par les nœuds modaux qui en sont proches. □

**Remarque.** Les nœuds modaux ont la particularité d'être caractérisables par leur étiquette. Plus précisément, lorsque  $\mathcal{T} = \langle T, L \rangle$  est un tableau pour une formule  $\gamma$ , un nœud

$n \in T$  est un nœud modal ssi l'ensemble de formules  $L(n)$  est composé de formules de la forme  $p$  ou  $\neg p$  pour  $p \in Prop$  ou bien  $a \rightarrow A$  pour une action  $a \in Act$  et un ensemble fini de formules  $A$ . De plus, toutes les formules apparaissant dans  $L(n)$  sont des *sous-formules* de  $\gamma$ .

### 3.2.2 Sémantique des tableaux

Dans cette partie, on interprète les tableaux comme des automates infinis sur les systèmes de transitions qui capture la sémantique des formules à partir desquelles ils sont construits.

**3.2.2.1 Définition (Trace).** Etant donné un chemin  $\mathcal{P}$  sur un tableau  $\mathcal{T} = \langle T, L \rangle$ , une *trace* sur  $\mathcal{P}$  est une fonction  $F$  associant à tout nœud  $n$  sur un segment initial de  $\mathcal{P}$  (éventuellement  $\mathcal{P}$  tout entier) une formule  $\alpha$  de  $L(n)$  satisfaisant les conditions supplémentaires suivantes. Pour tout nœud  $m$  où  $F(m)$  est définie, pour  $n \in \mathcal{P}$  le fils de  $m$  dans  $\mathcal{P}$  :

1. si la formule  $F(m)$  n'est pas réduite par la règle associant  $m$  à  $n$  alors  $F(n) = F(m)$ ,
2. si la formule  $F(m)$  est réduite en  $m$  alors  $F(n)$  est l'un des résultats de cette réduction.

La notion de “résultat de réduction” est évidente sauf pour la règle (*mod*). Dans le cas où  $m$  est un nœud modal et  $L(n) = \{\alpha\} \cup \{\bigvee B \mid a \rightarrow B \in C, B \neq A\}$  avec  $a \rightarrow A \in L(m)$  et  $\alpha \in A$ , la trace  $F$  devra vérifier :

1.  $F(n) = \alpha$  si  $F(m) = a \rightarrow A$ ,
2.  $F(n) = \bigvee B$  si  $F(m) = a \rightarrow B$  pour une certaine formule  $a \rightarrow B \in C$  telle que  $B \neq A$ ,
3. dans les autres cas,  $F(m)$  est un littéral,  $F(n)$  n'est pas définie et la trace termine au nœud  $m$ .

**Remarque.** Les formules étant gardé, tout chemin infini passe infiniment souvent par des nœuds modaux. Toute trace infinie est donc complètement caractérisée par son image sur les nœuds modaux et les nœuds de choix, sa valeur sur les noeuds se trouvant entre un noeud modal  $m$  et un noeud de choix  $n$  qui lui est proche étant totalement déterminé par ces noeuds.

**3.2.2.2 Définition ( $\mu$ -trace).** On dit qu'il y a une *régénération* de la variable  $X$  sur une trace  $F$  associée à un chemin  $\mathcal{P}$  lorsque, pour un certain nœud  $m$  et son fils  $n$  sur le

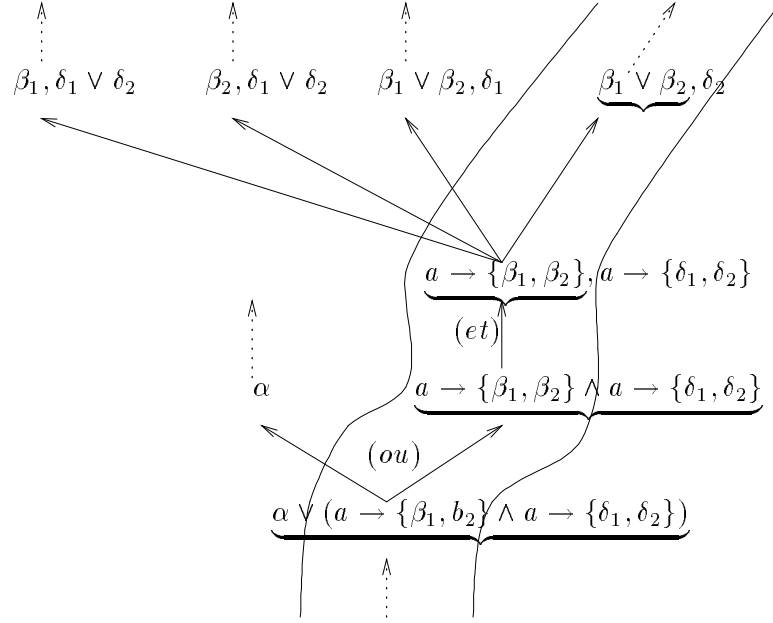


FIG. 8 - Un exemple de chemin et trace.

chemin  $\mathcal{P}$  on a  $F(m) = X$  et  $F(n) = \alpha(X)$  avec  $\mathcal{D}_\gamma(X) = \sigma X.\alpha(X)$ , i.e. la règle (*reg*) est appliquée au nœud  $m$  sur la variable  $X$ .

Une telle trace sera appelée  $\mu$ -trace lorsque c'est une trace infinie (définie sur tout le chemin  $\mathcal{P}$ ) sur laquelle la plus petite variable (au sens de l'ordre de dépendance) régénérée infiniment souvent est une  $\mu$ -variable.

De façon similaire, une trace sera appelée  $\nu$ -trace lorsque c'est une trace infinie sur laquelle la plus petite variable régénérée infiniment souvent est une  $\nu$ -variable.

**Remarque.** Toute trace infinie est soit une  $\mu$ -trace soit une  $\nu$ -trace. En effet, toutes les règles excepté la régénération diminuent la taille des formules apparaissant dans les labels de tableaux. De plus, bien que l'ordre de dépendance soit un ordre partiel, deux variables apparaissant infiniment souvent et pouvant être régénérées successivement sur une même trace sont nécessairement comparables selon l'ordre de dépendance. Ceci garantit l'existence de la plus petite variable régénérée infiniment souvent.

La sémantique proprement dite est décrite via la notion de marquage.

**3.2.2.3 Définition (Marquage).** Pour un tableau  $\mathcal{T} = \langle T, L \rangle$  nous définissons un *marquage* du tableau  $\mathcal{T}$  par un système de transitions  $S$  comme une relation  $M \subseteq E_S \times T$

satisfaisant les conditions suivantes :

1.  $(r_S, n_0) \in M$ , où  $n_0$  est la racine de  $T$ .
2. Si une paire  $(s, m)$  appartient à  $M$  et une règle autre que  $(mod)$  est appliquée en  $m$  alors pour au moins un fils  $n$  de  $m$  la paire  $(s, n)$  appartient à  $M$ .
3. Si une paire  $(s, m)$  appartient à  $M$  et la règle  $(mod)$  est appliquée au nœud  $m$  alors, pour toute action  $a$  telle qu'il existe une formule de la forme  $a \rightarrow A$  dans  $L(n)$ :
  - (a) pour tout  $a$ -fils  $n$  de  $m$  il existe un état  $f \in E_S$  tel que  $e \xrightarrow{a}_S f$  avec  $(f, n) \in M$ .
  - (b) pour tout état  $f$  tel que  $e \xrightarrow{a}_S f$  il existe un  $a$ -fils  $n$  de  $m$  avec  $(f, n) \in M$ .

**3.2.2.4 Définition (Marquage consistant).** En utilisant les notations de la définition 3.2.2.3, un marquage  $M$  d'un tableau  $\mathcal{T}$  par un système de transitions  $S$  est dit *consistant* lorsqu'il satisfait les conditions suivantes :

**consistance locale** pour tout nœud modal et pour tout état  $e$ , si  $(e, m) \in M$  alors pour tout littéral  $l$  apparaissant dans  $L(m)$ , si  $l = \neg p$  alors  $p \notin \rho_S(e)$ , et si  $l = p$  alors  $p \in \rho_S(e)$ ,

**consistance globale** pour tout chemin  $\mathcal{P} = n_0, n_1, \dots$  de  $T$  marqué, i.e. pour tout  $i = 0, 1, \dots$  il existe un état  $e_i$  tel que  $(e_i, n_i) \in M$ ,  $\mathcal{P}$  ne contient pas de  $\mu$ -trace.

Il vient :

**3.2.2.5 Théorème.** *Pour toute formule positive et gardée  $\gamma$ , tout système de transitions  $S$ ,  $S \models \gamma$  si et seulement si pour tout tableau  $\mathcal{T}$  de  $\gamma$  il existe un marquage consistant de  $\mathcal{T}$  relativement à  $S$ .*

*Preuve.* Le reste de cette partie est dévolu à la preuve de ce théorème.

Soit  $\gamma$  une telle formule,  $\mathcal{D}$  la fonction de liaison associée,  $S$  un système de transitions. Notre preuve repose essentiellement sur le calcul par approximation transfinie des points fixes.

(*Seulement si*) Supposons  $\gamma$  satisfaite dans un état  $e_0$  d'un système de transitions  $S$ . Soit  $\mathcal{T}$  un tableau pour  $\gamma$ . Nous allons construire un marquage consistant  $M$  de  $\mathcal{T}$  par  $S$ .

- Pour  $n_0$  la racine de  $\mathcal{T}$ , on pose  $(e_0, n_0) \in M$ .
- Si une règle unaire (hormis un cas particulier de  $(mod)$ ) est appliquée en un nœud  $n$  alors, comme souhaité, pour l'unique nœud  $m$  fils de  $n$ , pour tout état  $e$  tel que  $(e, n) \in M$ , on pose  $(e, m) \in M$ .

- Si la règle (*ou*) est appliquée dans un nœud  $n$ ,

$$\frac{\{\alpha, C\} \quad \{\beta, C\}}{\{\alpha \vee \beta, C\}}$$

alors, pour toute paire  $(e, n) \in M$ , étant donné  $m_1$  le fils de  $n$  étiqueté par  $\{\alpha, C\}$  et  $m_2$  l'autre fils, si  $\text{Sig}_\mu(\alpha, e) \leq \text{Sig}_\mu(\beta, e)$  alors on pose  $(e, m_1) \in M$  sinon on pose  $(e, m_2) \in M$ .

- Si la règle (*mod*) est appliquée dans un nœud  $n$  avec  $(e, n) \in M$ , alors pour toute règle  $a \rightarrow A \in L(n)$ :

1. pour toute formule  $\alpha \in A$  il existe un état  $f$  avec  $(e, f) \in R(a)$  tel que  $\text{Sig}_\mu(\alpha, f) \leq \text{Sig}_\mu(a \rightarrow A, e)$ , on pose alors  $(f, m_\alpha) \in M$  pour  $m_\alpha$  le  $a$ -fils de  $n$  tel que  $L(m_\alpha) = \{\alpha\} \cup \{\vee B \mid a \rightarrow B \in L(n) \text{ avec } A \neq B\}$ .
2. pour tout état  $f$  tel que  $(e, f) \in R(a)$  on sait que  $\text{Sig}_\mu(\vee A, f) \leq \text{Sig}_\mu(a \rightarrow A, e)$  ainsi, en appliquant le lemme 3.1.4.3, pour au moins un  $\alpha \in A$ ,  $\text{Sig}_\mu(\alpha, f) = \text{Sig}_\mu(\vee A, f) \leq \text{Sig}_\mu(a \rightarrow A, e)$ , on pose alors  $(f, m_\alpha) \in M$  avec  $m_\alpha$  défini comme dans le cas 1 ci-dessus,

**Remarque.** Notre construction nous assure donc que tout état  $f$  tel que  $(e, f) \in R_S(a)$  est marqué par au moins un  $a$ -fils du nœud  $n$ , et, de façon similaire, tout  $a$ -fils du nœud  $n$  est marqué par un état  $a$ -successeur de l'état  $e$ .

Dans les deux cas, pour un tel état  $f$ , pour toute formule  $a \rightarrow B \in L(n)$  avec  $B \neq A$ , on a bien  $\text{Sig}_\mu(\vee B, f) \leq \text{Sig}_\mu(a \rightarrow B, e)$ . Par conséquent, pour toute trace  $\mathbf{Tr}$  sur un chemin  $\mathcal{P}$  contenant  $n$  et  $m_\alpha$  respectivement marqués par  $e$  et  $f$  comme ci-dessus, on a bien  $\text{Sig}_\mu(\mathbf{Tr}(\mu_\alpha), f) \leq \text{Sig}_\mu(\mathbf{Tr}(n), f)$ .

On vérifie alors, par induction, que pour toute paire  $(s, n) \in M$  construite selon les règles précédentes, pour toute formule  $\alpha \in L(n)$ ,  $S[s/r] \models \langle \alpha \rangle_{\mathcal{D}}$ . En particulier, un tel marquage est localement consistant.

Pour vérifier la consistance globale d'un tel marquage, étant donné

$$\{(s, m) \in M \mid s \in \mathcal{S} \wedge m \in \mathcal{P}\}$$

un chemin quelconque de  $M$ ,  $\mathcal{S}$  et  $\mathcal{P}$  étant ses projections sur  $S$  et  $T$ , nous allons montrer qu'il ne peut y avoir de  $\mu$ -trace sur  $\mathcal{P}$ .

Supposons en effet que  $\mathbf{Tr}$  est une  $\mu$ -trace sur  $\mathcal{P}$  et soit  $X_i$  la plus petite variable régénérée infiniment souvent sur cette trace. La variable  $X_i$  est une  $\mu$ -variable et, à partir d'un



certain rang dans  $\mathcal{P}$  seules des variables dépendant de  $X_i$  sont régénérées. Le lemme 3.1.4.3 et la construction de notre marquage nous assurent que, à partir de ce rang, pour toutes paires  $(e, n)$  et  $(f, m)$  apparaissant dans ce chemin de  $M$  telles que  $m$  et le fils de  $n$ , on a :

$$\Pi_i(\text{Sig}_\mu(\mathbf{Tr}(m), f)) \leq \Pi_i(\text{Sig}_\mu(\mathbf{Tr}(n), e))$$

où, pour toute signature  $s$ ,  $\Pi_i(s)$  désigne les  $i$  premiers ordinaux de  $s$ , cette inégalité étant stricte chaque fois que le nœud  $m$  s'obtient à partir du nœud  $n$  en appliquant la règle de régénération de la variable  $X_i$ . La variable  $X_i$  étant régénérée infiniment souvent, on peut donc extraire de ce chemin une suite infinie strictement décroissante de séquences d'ordinaux de longueur  $i$ . L'ensemble des ordinaux étant bien fondé, cela conduit à une absurdité.

(Si) Soit  $\mathcal{T}$  un tableau pour  $\gamma$ . Par l'absurde, supposons qu'il existe un marquage consistant de  $\mathcal{T}$  par  $S$  tout en supposant  $S \not\models \gamma$ . Nous allons alors montrer, par induction, qu'il existe une  $\mu$ -trace  $\mathbf{Tr}$  sur un chemin  $\mathcal{P}$  de  $\mathcal{T}$  dont tous les nœuds sont marqués par des états de  $S$ .

Avec  $n_0$  la racine de  $\mathcal{T}$ , on pose  $\mathbf{Tr}(n_0) = \gamma$ , on a bien  $(r_S, n_0) \in M$ . Supposons alors que nous ayons construit  $\mathcal{P}$  et  $\mathbf{Tr}$  jusqu'à un certain nœud  $n$  avec  $\alpha = \mathbf{Tr}(n) \in L(n)$  et  $(e, n) \in M$  pour un certain état  $e \in E_S$  tel que  $S[e/r_S] \not\models \alpha$ . On procède alors selon la règle de réduction appliquée au nœud  $n$ .

- Dans le cas d'une règle unaire, en notant  $m$  l'unique fils de  $n$  : si elle est appliquée à la formule  $\alpha$  on pose  $\mathbf{Tr}(m) = \beta$  où  $\beta$  est le résultat de cette réduction de  $\alpha$  ; si elle n'est pas appliquée à  $\alpha$  on pose  $\mathbf{Tr}(m) = \alpha$ . Dans le cas où la règle (*et*) est appliquée sur  $\alpha = \alpha_1 \wedge \alpha_2$  on pose  $\mathbf{Tr}(m) = \alpha_1$  lorsque  $\text{Sig}_\nu(\alpha_1, s)$  est plus petite que  $\text{Sig}_\nu(\alpha_2, s)$ , sinon on pose  $\mathbf{Tr}(m) = \alpha_2$ .
- Si la règle (*ou*) est appliquée en  $n$  on choisit l'un des fils  $m$  de  $n$  tel que  $(e, m) \in M$  en posant  $\mathbf{Tr}(m) = \alpha_1 \in L(m)$  ou  $\mathbf{Tr}(m) = \alpha_2 \in L(m)$  dans le cas où la réduction porte sur la formule  $\alpha = \alpha_1 \vee \alpha_2$  selon le réduct porté par  $m$ , ou bien  $\mathbf{Tr}(m) = \alpha$  dans le cas où  $\alpha$  n'est pas réduite. Dans tous les cas,  $S[e/r_S] \not\models \alpha$  et donc l'état  $e$  ne satisfait pas le nouvel élément de cette trace.
- Si la règle (*mod*) est appliquée en  $n$  alors, par définition d'un marquage consistant,  $\alpha$  ne peut être un littéral, c'est donc une formule de la forme  $a \rightarrow A$ . Dans ce cas :
  1. Soit il existe une formule  $\beta \in A$  telle que pour tout  $a$ -successeur  $t$  de  $s$  on a  $S[t/r_S] \not\models \beta$  et  $\text{Sig}_\nu(\alpha, t) \leq \text{Sig}_\nu(a \rightarrow A, s)$ . Dans ce cas, on choisit un fils  $m$  de

$n$  étiqueté par  $\{\beta\} \cup \{\bigvee \mathbf{E} : a \rightarrow \mathbf{E} \in L(n), \mathbf{E} \neq A\}$ . On prend aussi un état  $t$ ,  $a$ -successeur de  $s$ , tel que  $(t, m) \in M$ . On pose  $\mathbf{Tr}(m) = \beta$ .

2. Soit il existe un état  $t$ , tel que  $(s, t) \in R_S(a)$  avec  $S[t/r_S] \not\equiv \bigvee A$  et  $\text{Sig}_\nu(\bigvee A, t) \leq \text{Sig}_\nu(a \rightarrow A, s)$ . Dans ce cas, on prend un fils  $m$  de  $n$  tel que  $(t, m) \in M$ . On pose alors  $\mathbf{Tr}(m) = \bigvee A$  ou bien  $\mathbf{Tr}(m) = \beta \in A$  selon que  $\bigvee A \in L(m)$  ou  $\beta \in L(m)$ . De toute façon, on a  $\text{Sig}_\nu(\beta, t) \leq \text{Sig}_\nu(\bigvee A, t)$ .

En utilisant des arguments similaires au cas *Seulement Si*, on montre alors facilement que cette trace ne peut être une  $\nu$ -trace, c'est donc une  $\mu$ -trace ce qui contredit la consistance globale du marquage.  $\square$

### 3.3 Automates modaux alternants

Dans cette section, on caractérise le pouvoir expressif du  $\mu$ -calcul modal à l'aide d'une notion appropriée d'automates appelés automates modaux alternants. Intuitivement, une telle caractérisation illustre le fait que les systèmes d'équations de plus grands et plus petits points fixes ne sont ni plus ni moins que des automates avec critère de parité [32].

La notion d'automate modal alternant nous permettra dans le dernier chapitre de cette thèse de manipuler plus simplement les langages de systèmes de transitions définissables en  $\mu$ -calcul modal.

#### 3.3.1 Définition des automates modaux alternants

**3.3.1.1 Définition (Automates modaux alternants).** Un *automate modal alternant* est un quadruplet

$$A = \langle Q, q_0, Act, \Sigma_p, \delta, \Omega \rangle$$

avec :

1. un ensemble fini d'états  $Q$ ,
2. un état initial  $q_0$ ,
3. un ensemble d'étiquettes  $\Sigma_p = \mathcal{P}(Prop_A)$ ,  $Prop_A$  une partie finie de  $Prop$ ,
4. une fonction de transition  $\delta : Q \times \Sigma_p \rightarrow \mathcal{P}(Act \rightarrow \mathcal{P}(\mathcal{P}(Q)))$ ,
5. une fonction de parité  $\Omega : Q \rightarrow \mathbb{N}$ .

**3.3.1.2 Définition (Calcul).** Etant donné un système de transitions  $S$ , un automate modal alternant  $A$ , un *calcul modal alternant* ou plus simplement calcul de  $A$  sur  $S$  est un arbre  $T$  fini ou infini dont les nœuds sont étiquetés sur  $Q \times E_S$  et qui vérifie les propriétés suivantes :

1. la racine de  $T$  est étiquetée par  $(q_0, r_S)$ ,
2. pour tout nœud  $n$  de  $T$  étiqueté par une paire  $(q, e)$  il existe une transition  $f \in \delta(q, \rho_S(e) \cap Prop_A)$  telle que, pour toute action  $a \in Act$  :
  - (a) pour tout sommet  $e_1 \in E_S$  tel que  $(e, e_1) \in R_S(a)$  il existe un ensemble d'états  $Q_1 \in f(a)$  tel que, pour tout état  $q_1 \in Q_1$ , le nœud  $n$  admet un fils étiqueté  $(q_1, e_1)$ ,
  - (b) pour tout ensemble d'états  $Q_1 \in f(a)$  il existe un sommet  $e_1 \in E_S$  tel que  $(e, e_1) \in R_S(a)$  et, pour tout état  $q_1 \in Q_1$ , le nœud  $n$  admet un fils étiqueté  $(q_1, e_1)$ .

Un calcul  $T$  est dit *acceptant* lorsque, pour tout chemin infini de  $T$  le minimum des indices  $\Omega(q)$ , pour  $q$  variant dans l'ensemble des états apparaissant infiniment souvent sur ce chemin, est pair.

**Remarque.** En fait, le calcul de l'automate défini ci-dessus porte implicitement sur le développement universel du système considéré. Il est parfaitement possible de définir une autre notion de calcul portant sur le système  $S$  lui-même, i.e. un marquage adéquat des états de  $S$ . Cette alternative n'est cependant pas explicitée dans le présent mémoire.

On dit qu'un système de transitions  $S$  est *reconnu* par un automate modal alternant  $A$  lorsqu'il existe un calcul acceptant de l'automate  $A$  sur le système  $S$ . On note alors  $L(A)$  le langage reconnu par  $A$ , c.a.d. la classe des systèmes de transitions reconnus par l'automate  $A$ . Par extension, pour tout état  $q \in Q$ , on notera  $L_A(q)$  le langage reconnu par l'automate obtenu à partir de  $A$  en prenant l'état  $q$  comme état initial à la place de l'état  $q_0$ .

### 3.3.2 Traces et développement d'un automate

**3.3.2.1 Définition (Traces d'automate).** On appelle *trace* de  $A$  tout mot fini  $u$  de la forme

$$u = (P_1, f_1, a_1, Q_1, q_1) \cdots (P_n, f_n, a_n, Q_n, q_n)$$

avec, pour tout  $i \in \{1, \dots, n\}$ ,

1. un ensemble de propositions  $P_i \in \Sigma_p$ ,

2. une “transition modale alternante”  $f_i \in \delta(q_{i-1}, P_i)$  (avec  $q_0$  l’état initial de  $A$ ),
3. une action  $a_i \in Act$ ,
4. un ensemble d’états  $Q_i \in f_i(a_i)$ ,
5. et un état  $q_i \in Q_i$ .

**Notation.** Dans la suite, pour toute trace  $u$  on pose  $dest(u) = q_0$  si  $u$  est le mot vide et  $dest(u) = q$  si  $u$  est de la forme  $v.(P, f, a, Q', q)$ .

**3.3.2.2 Définition (Développement d’un automate modal alternant).** On appelle développement d’un automate modal alternant  $A$ , l’automate  $\widehat{A}$ , en forme d’arbre et éventuellement infini, défini par :

1. un ensemble d’états  $\widehat{Q}$  égal à l’ensemble des traces finies de  $A$ ,
2. un état initial  $\widehat{q}_0 = \epsilon$ ,
3. une fonction de transition  $\widehat{\delta} : \widehat{Q} \times \Sigma_p \rightarrow \mathcal{P}(Act \rightarrow \mathcal{P}(\mathcal{P}(\widehat{Q})))$  définie, pour toute trace  $u$ , tout ensemble de propositions  $P \in \Sigma_p$  par  $F \in \widehat{\delta}(u, P)$  ssi il existe une fonction  $f \in \delta(dest(u), P)$  telle que, pour toute action  $a \in Act$ ,

$$F(a) = \{\{u.(P, f, a, Q', q') \mid q' \in Q'\} \mid Q' \in f(a)\}$$

4. une fonction de parité  $\widehat{\Omega}$  définie pour toute trace  $u$  par  $\widehat{W}(u) = \Omega(dest(u))$ .

**Remarque.** Le développement d’un automate  $A$  est effectivement un “arbre”<sup>1</sup> dont les “transitions” sont cependant de formes plus complexes que dans le cas des arbres de transitions. En général l’ensemble des états de  $\widehat{A}$  est infini. Cela ne nous empêche pas pour autant d’étendre les notions de calcul et calcul acceptant d’un automate fini à ce cas infini.

**3.3.2.3 Proposition.** *Pour tout automate modal alternant  $A$ , le langage des systèmes de transitions reconnus par  $A$  est égal au langage des systèmes de transitions reconnus par le développement  $\widehat{A}$  de  $A$ .*

*Preuve.* Immédiate. □

**Remarque.** Tout calcul  $T$  de  $\widehat{A}$  sur  $S$  est complètement décrit par l’ensemble de ses étiquettes. Un nœud étiqueté par une paire  $(u_1, e_1)$  est le fils d’un nœud étiqueté  $(u, e)$  ssi  $u_1$  est de la forme  $u.(P, f, a, Q, q)$ , et dans ce cas  $(e, e_1) \in R_S(a)$ .

---

<sup>1</sup>c’est l’analogie des développements universels de systèmes de transitions!

### 3.3.3 Formes normales arborescentes

**3.3.3.1 Définition (Forme normale arborescente).** Un automate modal alternatif

$$A = \langle Q, q_0, \Sigma_p, \delta, \Omega \rangle$$

est dit sous *forme normale arborescente* lorsqu'il existe une relation d'ordre  $\leq_A \subseteq Q \times Q$ , dont la relation de couverture associée est notée  $\ll_A$  (i.e.  $q \ll_A q'$  ssi  $q <_A q'$  et il n'existe pas d'état  $q''$  tel que  $q < q'' < q'$ ), satisfaisant :

1.  $q_0$  est le minimum,
2. pour tous états  $q_1, q_2, q_3 \in Q$ , si  $q_2 \leq_A q_1$  et  $q_3 \leq_A q_1$  alors  $q_2$  et  $q_3$  sont comparables, i.e. la relation de couverture  $\ll_A$  induit une structure d'arbre sur les états de  $A$ ,
3. pour tous états  $q_1, q_2 \in Q$ , s'il existe  $P \in \Sigma_p$ ,  $a \in Act$ ,  $f \in \delta(q_1, P)$ ,  $Q_1 \in f(a)$  tels que  $q_2 \in Q_1$  alors :
  - (a) soit  $q_1 \ll_A q_2$ ,
  - (b) soit  $q_2 \leq_A q_1$  (on dit alors que  $q_2$  est un *état retour* et que  $q_1$  est la *source d'un retour*),
4. pour tout état retour  $q_2 \in Q$ , pour tout état  $q \in Q$  tel que  $q_2 \leq_A q$ ,  $\Omega(q_2) \leq \Omega(q)$ .

La figure 9 illustre la notion d'automate sous forme arborescente dans le cas, plus "lisible", d'un automate de mots avec critère de parité.

**3.3.3.2 Lemme.** *Sans perte de généralité, on peut se restreindre aux automates modaux alternants sous forme normale arborescente.*

*Preuve.* Etant donné un automate modal alternatif

$$A = \langle Q, q_0, \Sigma_p, \delta, \Omega \rangle$$

la construction à partir de  $A$  d'un automate sous forme arborescente se fait par développement de  $A$ , puis par repliage en veillant à satisfaire la condition 4.

Plus précisément, en notant  $\widehat{Q}$  l'ensemble des *traces finies* de  $A$ , on définit la relation de transition  $\delta_c$  en posant, pour toute trace de la forme

$$u = (P_1, f_1, a_1, Q_1, q_1) \cdots (P_n, f_n, a_n, Q_n, q_n)$$

pour tout ensemble de propositions  $P \in \Sigma_p$ ,  $\delta_c(u, P)$  est l'ensemble des fonctions  $f_g$  pour  $g \in \delta(q_n, P)$  avec  $f_g$  défini de la façon suivante : pour toute action  $a \in Act$

$$\{v_1, v_2, \dots, v_m\} \in f_g(a)$$

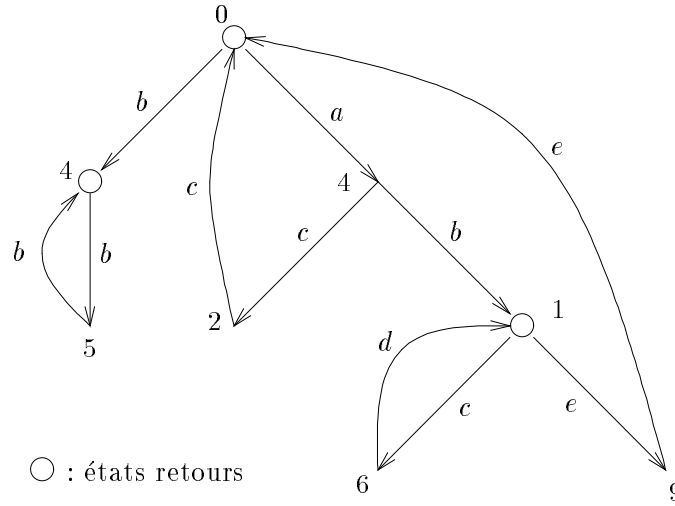


FIG. 9 - Un automate de mots sous forme normale arborescente reconnaissant

$$L = (a(b(cd)^*ee + cc) + b(bb)^\omega)^\omega.$$

ssi il existe  $Q' = \{r_1, \dots, r_m\} \in g(a)$  tel que, pour tout  $k \in \{1, \dots, m\}$ :

1. s'il existe  $i \in \{1, n\}$  tel que  $q_i = r_k$  avec, pour tout état  $q \in \{q_i, q_{i+1}, \dots, q_n\}$ ,  $\Omega(q_i) = \Omega(r_k) \leq \Omega(q)$ , alors  $v_k = (P_1, f_1, a_1, Q_1, q_1) \dots (P_i, f_i, a_i, Q_i, q_i)$
2. sinon  $v_k = u.(P, g, a, Q', r_k)$ .

On vérifie alors que si  $A$  est un automate fini, il n'existe qu'un nombre fini de traces de  $A$  accessibles à partir du mot vide  $\epsilon \in \widehat{Q}$  en n'usant que des transitions définies par  $\delta_c$ .

En prenant comme ensemble d'états l'ensemble de ces traces accessibles via  $\delta_c$ , comme état initial le mot vide  $\epsilon$ , comme fonction de transition la fonction  $\delta_c$  et comme fonction de parité la fonction  $\Omega_c$  définie à partir de  $\Omega$  par  $\Omega_c(\epsilon) = \Omega(q_0)$  et pour tout chemin de la forme  $u.(P, f, a, Q, q)$ ,  $\Omega_c(u.(P, f, a, Q, q)) = \Omega(q)$ , on a bien un automate sous forme arborescente (en prenant comme ordre l'ordre préfixe) reconnaissant le même langage de systèmes de transitions que l'automate modal alternant  $A$ .  $\square$

**Remarque.** Dans un automate sous forme normale arborescente, l'ordre  $\leq_A$  définissant la structure arborescente sous-jacente est déterminé de façon unique. Il caractérise en effet la distance de chaque état à l'état initial.

### 3.3.4 Automates alternants et $\mu$ -formules modales

**3.3.4.1 Théorème.** *Pour tout automate modal alternant  $A$  il existe une formule positive  $\alpha$  du  $\mu$ -calcul modal telle que, pour tout système de transitions  $S$ ,  $S \in L(A)$  ssi  $S \models \alpha$ .*

*Preuve.* En appliquant le lemme 3.3.3.2, on peut toujours supposer que l'automate  $A$  est sous forme normale arborescente. On construit alors la formule  $\alpha$  à partir de l'automate  $A$  par induction sur la structure arborescente de  $A$  des feuilles vers la racine.

On suppose qu'on dispose d'un ensemble de variables  $\{X_q\}_{q \in Q}$ . Pour tout état  $q \in Q$ , on construit par induction la formule  $\alpha_q$  de la façon suivante :

$$\alpha_q = (\sigma_q X_q) \bigvee_{P \in \Sigma_p} \bigvee_{f \in \delta(q, P)} \alpha_P \wedge \bigwedge_{a \in Act} a \rightarrow \left\{ \bigwedge_{q_1 \in Q_1} \beta_{q, q_1} \mid Q_1 \in f(a) \right\}$$

avec :

1.  $\sigma_q = \nu$  lorsque  $\Omega(q)$  est pair,  $\sigma_q = \mu$  lorsque  $\Omega(q)$  est impair,
2.  $\beta_{q, q_1} = X_{q_1}$  si  $q_1 \leq_A q$ , i.e.  $q_1$  est un état retour, et  $q$  la source d'un retour sur  $q_1$ , et  $\beta_{q, q_1} = \alpha_{q_1}$  dans le cas contraire.

**Remarque.** Dans cette formule, on peut tout aussi bien supprimer le constructeur de point fixe lorsque  $q$  n'est pas un état retour sans changer la sémantique de la formule  $\alpha_q$ . La variable  $X_q$  ne peut en effet apparaître librement dans la suite de la formule. Ces constructeurs sont introduits dans ce cas dans le seul but d'uniformiser la construction.

**Notation.** Dans la suite, pour tout état  $q \in Q$ , pour toute partie  $Q_1 \subseteq Q$  on note

$$C(q, Q_1) \stackrel{\text{def}}{=} \bigwedge_{q_1 \in Q_1} \beta_{q, q_1}$$

et on note, pour toute fonction  $f : Act \rightarrow \mathcal{P}(\mathcal{P}(Q))$ , pour toute action  $a \in Act$  :

$$V(q, f, a) \stackrel{\text{def}}{=} a \rightarrow \{C(q, Q_1) \mid Q_1 \in f(a)\}$$

Avec ces notations,

$$\alpha_q = \sigma_q X_q \bigvee_{P \in \Sigma_p} \bigvee_{f \in \delta(q, P)} \bigwedge_{a \in Act} V(q, f(a))$$

On pose alors  $\alpha = \alpha_{q_0}$ . Il nous reste à montrer qu'un système de transitions est reconnu par  $A$  ssi c'est un modèle de  $\alpha$ . La sémantique du  $\mu$ -calcul modal et celle des automates modaux alternants préservant l'équivalence de bisimulation, on peut, sans perte

de généralité se restreindre au cas des arbres de transitions. Soit donc  $\mathcal{T} = \langle T, L \rangle$  un tableau pour  $\alpha$  et  $S$  un arbre de transitions.

Un nœud modal  $m \in T$  est dit *immédiatement inconsistant* lorsque qu'il existe une proposition  $p \in Prop_A$  telle que  $p \in L(m)$  et  $\neg p \in L(m)$ . Par définition des marquages consistants, un nœud modal immédiatement inconsistant ne peut apparaître dans ces marquages, on pourra donc se restreindre à l'ensemble  $T_{mc} \subseteq T$  des nœuds modaux de  $T$  qui ne sont pas immédiatement inconsistants. Dans la suite, pour tout nœud  $m \in T_{mc}$ , on note

$$P_m = L(m) \cap Prop_A$$

En notant  $Tr_A$  l'ensemble des traces finies de l'automate  $A$  on peut montrer par induction sur la structure de  $\mathcal{T}$  qu'il existe une unique fonction

$$F : T_{mc} \rightarrow \mathcal{P}_{fin}(Tr_A \times (Act \rightarrow \mathcal{P}(\mathcal{P}(Q))))$$

telle que :

1. pour tout nœud  $m \in T_{mc}$ ,  $m$  est proche de la racine de  $\mathcal{T}$  ssi il existe  $f \in \delta(q_0, P_m)$  tel que

$$F(m) = \{(\epsilon, f)\}$$

2. pour tout nœud  $m \in T_{mc}$  tel que  $F(m) = \{(u_i, f_i)\}_{i \in \mathcal{I}_m}$  :

- (a) en notant, pour toute trace de l'automate  $A$ ,  $dest(u) = q_0$  si  $u = \epsilon$  et  $dest(u) = q$  si  $u$  est de la forme  $v.(P, f, Q, q)$ ,

$$L(m) = P_m \cup \{\neg p \mid p \notin P_m\} \cup \bigcup_{i \in \mathcal{I}_m} \{V(dest(u_i), f_i(a)) \mid a \in Act\}$$

- (b) pour tout nœud  $n \in T_{mc}$ , pour toute action  $a \in Act$ ,  $n$  est proche d'un  $a$ -fils du nœud  $m$  ssi pour tout  $i \in \mathcal{I}_k$  il existe  $Q_i \in f_i(a)$ , et pour tout  $q \in Q_i$ , une fonction  $f_q \in \delta(q, P_n)$  tels que

$$F(n) = \bigcup_{i \in \mathcal{I}_m} \{(u_i.(P_m, f_i, a, Q_i, q), f_q) \mid q \in Q_i\}$$

Il vient :

**3.3.4.2 Proposition (Correspondance locale).** *Il existe un marquage localement consistant de  $\mathcal{T}$  par  $S$  ssi il existe un calcul du développement de l'automate  $A$  sur  $S$ .*



*Preuve.* Soit  $M$  un marquage localement consistant de  $\mathcal{T}$  par  $S$ . On pose

$$F(M) = \{(u, e) \in Tr_A \times E_S \mid \exists m \in T_{mc}, (e, m) \in M \wedge \exists f, (u, f) \in F(m)\}$$

Une induction simple mais fastidieuse sur la structure de  $\mathcal{T}$  nous permet de vérifier que  $F(M)$  est bien un calcul du développement de l'automate  $A$  sur  $S$ .

Réciproquement, soit  $C \subseteq Tr_A \times E_S$  un calcul du développement de l'automate  $A$  sur  $S$ . On pose

$$F_{mc}^{-1}(C) = \{(e, m) \in E_S \times T_{mc} \mid F(m) = \{(u_i, f_i)\}_{i \in \mathcal{I}_m} \wedge \forall i \in \mathcal{I}_m, (u_i, e) \in C\}$$

En définissant alors  $F^{-1}(C) \subseteq E_S \times T$  comme le plus petit ensemble contenant  $F_{mc}^{-1}(C)$  et tel que, pour toute paire  $(e, m)$  et  $(e_1, m_1) \in F_{mc}^{-1}(C)$ , tout fils  $n$  de  $m$ , pour tout sommet  $n_1$ , si  $m_1$  est proche de  $n_1$  et  $n_1$  est proche de  $n$  alors  $(e_1, n_1) \in F^{-1}(C)$ , on vérifie, là encore par une induction simple mais fastidieuse sur la structure de  $C$ , que  $F^{-1}(C)$  ainsi défini est bien un marquage localement consistant de  $\mathcal{T}$  par  $S$ . En fait, on vérifie que tout marquage localement consistant de  $\mathcal{T}$  par  $S$  dont la restriction au nœud modaux est égale à  $F_{mc}^{-1}(C)$  est nécessairement égal à  $F^{-1}(C)$ .  $\square$

De plus :

**3.3.4.3 Proposition (Correspondance globale).** *En usant des notations de la preuve précédente: si  $M$  est un marquage consistant de  $\mathcal{T}$  par  $S$  alors  $F(M)$  est un calcul acceptant du développement de l'automate  $A$  sur  $S$  et, réciproquement, si  $C$  est un calcul acceptant du développement de l'automate  $A$  sur  $S$  alors  $F^{-1}(C)$  est un marquage consistant de  $\mathcal{T}$  par  $S$ .*

*Preuve.* Soit  $M$  un marquage consistant de  $\mathcal{T}$  par  $S$  et  $(u_0, e_0).(u_1, e_1) \dots$  une branche infinie de  $F(M)$ . Il nous faut montrer que cette branche est acceptante.

Par construction de la fonction  $F$  il existe un chemin  $\mathcal{P}$  de  $\mathcal{T}$  et une trace  $\mathbf{Tr}$  de  $\mathcal{P}$  tels que, en notant  $m_0.m_1 \dots$  la séquence de nœuds modaux apparaissant sur  $\mathcal{P}$ , pour tout  $i \in \mathbb{N}$ ,  $(e_i, m_i) \in M$  et il existe une "transition"  $f_i$  et une action  $a_i \in Act$  telle que :

1.  $\mathbf{Tr}(m_i) = V(\text{dest}(u_i), f_i(a_i))$ ,
2.  $u_{i+1}$  est de la forme  $u_i.(P_{m_i}, f_i, a_i, Q', q')$  pour  $Q' \in \delta(\text{dest}(u_i), P_{m_i})$  et  $q' = \text{dest}(u_{i+1}) \in Q'$ .

Or le marquage  $M$  est consistant donc  $\mathbf{Tr}$  est une  $\nu$ -trace, autrement dit, la plus petite (selon l'ordre de dépendance) variable  $X_q$  régénérée infiniment souvent sur  $\mathbf{Tr}$  est une  $\nu$ -variable, i.e.  $\Omega(q)$  est pair. Mais, comme l'automate  $A$  est sous-forme normale arborescente,

la condition 3b de la définition 3.3.3.1 nous assure que l'état  $q$  est aussi, parmi les états apparaissant infiniment souvent sur la séquence  $dest(u_0).dest(u_1).\dots$ , celui d'indice de parité minimal.

Réciproquement, soit  $C$  un calcul du développement de l'automate  $A$  sur l'arbre  $S$ , soit  $\mathcal{P}$  un chemin infini de  $\mathcal{T}$  apparaissant dans  $F^{-1}(C)$ , i.e. tel que en notant  $m_0.m_1.\dots$  la séquence de nœuds modaux apparaissant sur  $\mathcal{P}$  il existe  $e_0.e_1.\dots \in E_S^\omega$  tel que, pour tout  $i \in \mathbb{N}$ ,  $(e_i, m_i) \in F^{-1}(C)$ , et soit  $\mathbf{Tr}$  une trace sur le chemin  $\mathcal{P}$ .

Là encore, on vérifie qu'il existe une séquence  $u_0.u_1.\dots \in Tr_A^\omega$  telle que, pour tout  $i \in \mathbb{N}$ ,  $(u_i, e_i) \in C$  satisfaisant les conditions 1 et 2 ci-dessus. Un raisonnement analogue, comme  $C$  est un calcul acceptant, nous permet alors d'affirmer que  $\mathbf{Tr}$  est une  $\nu$ -trace car l'indice minimum des états apparaissant infiniment souvent sur  $dest(u_0).dest(u_1).\dots$  est pair.  $\square$

*Preuve 3.3.4.1 suite.* On a donc montré que pour tout marquage consistant  $M$  du tableau  $\mathcal{T}$  par  $S$ , la restriction de  $M$  aux nœuds modaux de  $\mathcal{T}$  nous permet d'extraire un calcul acceptant  $F(M)$  du développement de l'automate  $A$  sur  $S$ , et, réciproquement, tout calcul  $C$  acceptant du développement de l'automate  $A$  sur  $S$  induit un marquage  $F_m^{-1}c(C)$  des nœuds modaux de  $\mathcal{T}$  par  $S$  consistant au sens où le marquage minimal de  $\mathcal{T}$  par  $S$  qu'il induit est consistant. Le théorème 3.2.2.5 nous permet alors de conclure.  $\square$

**3.3.4.4 Théorème.** *Pour toute formule du  $\mu$ -calcul modal  $\alpha$  il existe un automate modal alternant fini  $A$  tel que, pour tout système de transitions  $S$ ,  $S \models \alpha$  ssi  $S \in L(A)$ .*

*Preuve.* Ce théorème sera une conséquence immédiate des résultats énoncés dans le chapitre suivant. On ne donne donc ici qu'une idée de la preuve.

En appliquant le lemme 3.2.1.2, On peut toujours supposer que  $\alpha$  est gardée et bien nommée. A l'aide des règles de réécriture suivantes :

1.  $\sigma X.\alpha_1(X) \rightsquigarrow \alpha_1[\sigma X.\alpha_1(X)/X]$  pour  $\sigma = \mu$  ou  $\sigma = \nu$ ,
2.  $\alpha \rightsquigarrow \sigma X.\alpha$ , lorsque la variable  $X$  n'apparaît pas dans  $\alpha$  pour  $\sigma = \mu$  ou  $\sigma = \nu$ ,
3.  $a \rightarrow A_1 \wedge a \rightarrow A_2 \rightsquigarrow a \rightarrow \{\alpha_1 \wedge \bigvee A_2 \mid \alpha_1 \in A_1\} \cup \{\bigvee A_1 \wedge \alpha_2 \mid \alpha_2 \in A_2\}$ ,
4. et  $\alpha_1 \wedge (\alpha_2 \vee \alpha_3) \rightsquigarrow (\alpha_1 \wedge \alpha_2) \vee (\alpha_1 \wedge \alpha_3)$ ,

qui préservent l'équivalence sémantique, on peut, modulo associativité et commutativité des opérations booléennes, se ramener à une formule  $\beta$  équivalente à  $\alpha$ , bien nommée, de la forme de la formule construite dans la preuve du théorème 3.3.4.4.

De la structure de  $\beta$  on extrait facilement alors un automate modal alternant sous-forme normale arborescente  $A$  que l'on prouve équivalent à  $\beta$  de la même façon que précédemment.  $\square$



## Chapitre 4

# Formes normales disjonctives

Dans le cas des arbres binaires, les travaux de Muller et Schupp montrent que tout automate alternant peut être simulé par un automate non alternant [35]. Dans ce cas, une “alternance” est simplement définie par le fait que l’automate lit, en parallèle et à partir de deux états différents, le même sommet de l’arbre à reconnaître.

En terme de  $\mu$ -calcul, Niwiński [36], en traduisant tout automate de Rabin en formule d’un “ $\mu$ -calcul binaire” sans conjonctions, nous indique que le théorème de simulation dans le cas des automates alternants se traduit par une suppression de la plupart des conjonctions. Seules demeurent dans ce “ $\mu$ -calcul binaire” les conjonctions permettant de spécifier les propriétés des fils droit et gauche d’un même sommet. Ces conjonctions ne génèrent pas pour autant d’alternance au sens précédent.

Dans le cas des systèmes de transitions les choses se compliquent. Il ne semble pas possible de faire disparaître toutes les alternances au sens ci-dessus. Une première intuition suggère que l’absence totale d’alternances permettrait de compter partiellement le nombre de successeurs d’un sommet ce qui, dans une sémantique “close par bisimulation” est exclu.

Par contre, l’usage du combinateur  $a \rightarrow A$  et les résultats obtenus dans le cas de la logique modale paramétrée nous suggèrent qu’il est cependant possible d’interdire au moins toute conjonction de la forme  $a \rightarrow A_1 \wedge a \rightarrow A_2$  sans affecter le pouvoir d’expression du calcul. Cette intuition nous conduit à la notion de forme normale disjonctive pour le  $\mu$ -calcul : une notion découverte indépendamment par Igor Walukiewicz et le présent auteur, et qui a fait l’objet d’une publication jointe [27].

Plus précisément, on définit un sous-ensemble de formules du  $\mu$ -calcul modal, appelées disjonctives, qui a les propriétés remarquables suivantes :

1. toute formule du  $\mu$ -calcul est équivalente à une formule disjonctive, appelée forme normale disjonctive,

2. le problème de la satisfaisabilité est décidable en temps linéaire dans le cas des formules disjonctives (alors qu'il est *EXP*-space complet dans le cas général).

Ce résultat quant à la complexité du problème de la satisfaisabilité s'accompagne de la construction explicite, pour tout formule satisfaisable, d'un modèle fini. On retrouve ainsi, dans un cadre uniforme, les résultats présentés par Streett, Emerson et Jutla [43, 17]. Cette notion de forme normale disjonctive apparaît aussi comme un outil fondamental dans la preuve, par Igor Walukiewicz [47], de complétude de l'axiomatisation du  $\mu$ -calcul modal proposée par Kozen.

Dans ce chapitre, on montre aussi comment les formules disjonctives induisent naturellement une notion d'automate non alternant au sens où, sur les  $\omega$ -expansions de systèmes de transitions, on peut restreindre la notion de calcul à une notion sans alternances au sens ci-dessus, sans affecter la sémantique des automates. En se restreignant aux arbres binaires, cette approche apparaît comme une preuve alternative du théorème de simulation de Muller et Schupp [35] et du lemme de complémentation de Rabin [40].

## 4.1 Formes normales disjonctives

Dans cette section, nous définissons la notion de formule disjonctive pour le  $\mu$ -calcul, donnons leurs propriétés essentielles et montrons que toute formule est équivalente à une formule disjonctive.

### 4.1.1 Définitions et propriétés élémentaires

**4.1.1.1 Définition.** L'ensemble des *formules disjonctives du  $\mu$ -calcul* est le plus petit ensemble satisfaisant les conditions suivantes :

1. si  $A_1, A_2, \dots, A_n$  sont  $n$  ensembles finis de formules disjonctives, si  $a_1, a_2, \dots, a_n$  sont  $n$  actions deux à deux distinctes, si  $l_1, l_2, \dots, l_m$  sont  $m$  littéraux, alors la formule

$$\bigwedge_{i \in [1, n]} (a_i \rightarrow A_i) \bigwedge_{j \in [1, m]} l_j$$

est aussi disjonctive<sup>1</sup>,

2. si  $\alpha$  et  $\beta$  sont disjonctives alors  $\alpha \vee \beta$  l'est aussi,

---

<sup>1</sup>le cas  $n = 0$  constitue la base de l'induction !

3. si  $\alpha$  est disjonctive et si toutes les occurrences de la variable  $X$  dans  $\alpha$  sont positives et n'apparaissent pas dans un contexte de la forme  $X \wedge \beta$  pour une formule  $\beta$  quelconque, alors  $\mu X.\alpha$  et  $\nu X.\alpha$  sont disjonctives <sup>2</sup>.

**Remarque.** Avec un parenthésage approprié des conjonctions, par exemple un parenthésage de la forme

$$(\cdots(((\cdots(a_1 \rightarrow A_1 \wedge a_2 \rightarrow A_2) \wedge \cdots) \wedge a_n \rightarrow A_n) \wedge l_1) \wedge \cdots) \wedge l_m$$

les formules disjonctives admettent un unique tableau. De plus, chaque chemin infini sur un tel tableau admet une et une seule trace.

La pertinence des formules disjonctives est illustrée par la proposition suivante :

**4.1.1.2 Théorème.** *Une formule disjonctive  $\alpha$  est satisfaisable ssi la formule  $\beta$  obtenue à partir de  $\alpha$  en remplaçant toutes les occurrences de  $\mu$ -variables par la constante  $\perp$  et toutes les occurrences de  $\nu$ -variables par la constante  $\top$  est satisfaisable.*

*Preuve.* Soit  $\mathcal{T}_\alpha = \langle T_\alpha, L_\alpha \rangle$  et  $\mathcal{T}_\beta = \langle T_\beta, L_\beta \rangle$  des tableaux pour les formules  $\alpha$  et  $\beta$  respectivement. Observons tout d'abord que  $T_\beta$  est un arbre fini. L'arbre  $T_\alpha$  au contraire peut être vu comme le développement d'un graphe obtenu à partir de  $T_\beta$  en ajoutant une arête de retour à partir de tout nœud étiqueté par  $\{\perp\}$  ou  $\{\top\}$  vers le nœud portant la définition de la variable correspondante, i.e. une  $\mu$ -définition pour les nœuds étiquetés par  $\perp$ , une  $\nu$ -définition pour les nœuds étiquetés par  $\top$ .

Nous supposons donc construite une projection  $h : T_\alpha \rightarrow T_\beta$  qui, à tout nœud de l'arbre  $T_\alpha$  associe le nœud correspondant dans l'arbre  $T_\beta$ . La figure 10 illustre une telle construction.

Il est facile de montrer, par induction sur la structure de  $\alpha$  que si  $\alpha$  est satisfaisable alors  $\beta$  l'est aussi.

Réciproquement, supposons que  $\beta$  est satisfaisable et soit  $M$  un marquage consistant de  $T_\beta$  par un modèle arbitraire de  $\beta$ . On peut supposer que  $M_\beta$  est minimal vis-à-vis de l'inclusion. Soit alors  $T'_\beta$  le sous-arbre de  $T_\beta$  formé de tous les nœuds apparaissant dans  $M_\beta$  et posons  $T'_\alpha = h^{-1}(T'_\beta)$ . Nous construisons alors un modèle  $S_\alpha = \langle E, r, \{R(a)_{a \in Act}\}, \rho \rangle$  pour  $\alpha$  de la façon suivante :

1.  $E$  est l'ensemble des nœuds modaux de  $T'_\alpha$ ,
2.  $r$  est l'unique (sous l'hypothèse de minimalité) nœud modal de  $T'_\alpha$  qui soit proche de la racine de  $T'_\alpha$ .

---

<sup>2</sup>on doit autoriser les sous-formules de la formes  $X \wedge \beta$  lorsque la variable  $X$  est libre !

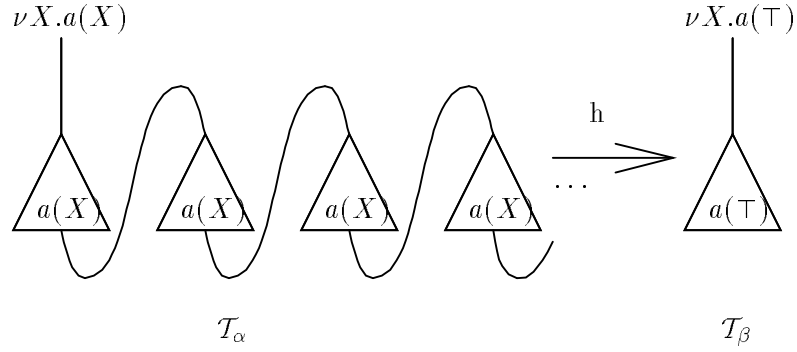


FIG. 10 - Projection de  $T_\alpha$  sur  $T_\beta$  dans le cas  $\alpha = \nu X.a(X)$

3. pour tout état  $e$  et  $f \in E$ , on pose  $(e, f) \in R(a)$  ssi le nœud  $f$  est proche d'un  $a$ -fils du nœud  $e$ ,
4. enfin  $\rho(e) = \{p \mid p \in L_\alpha(e)\}$ ; notons que, en travaillant à partir d'un modèle de  $\beta$ , si  $p \in \rho(e)$ , on ne peut avoir  $\neg p \in \rho(e)$ .

Un marquage consistant de  $T_\alpha$  par  $S_\alpha$  est alors donné par l'ensemble  $M_\alpha$  des paires  $(e, m) \in E \times T_\alpha'$  telles que  $e$  est proche de  $m$ . Comme aucun nœud étiqueté par  $\{\perp\}$  n'appartient au marquage  $M_\beta$ , aucune  $\mu$ -variable n'est régénérée dans les chemins de  $M_\alpha$ . Toutes les traces apparaissant dans  $M_\alpha$  sont donc des  $\nu$ -traces.  $\square$

**4.1.1.3 Proposition.** *Toute formule disjonctive  $\alpha$  satisfaisable admet un modèle fini.*

*Preuve.* Le modèle de  $\alpha$  construit ci-dessus est infini. Cependant, comme  $T_\alpha$  est un arbre régulier, ce modèle peut être replié en un système fini qui sera toujours,  $\alpha$  étant stable par développement, un modèle de  $\alpha$ .  $\square$

Puisque la formule  $\beta$  dans le théorème précédent ne contient pas de point fixe, le théorème 2.3.2.3 s'applique, il vient :

**4.1.1.4 Corollaire.** *La satisfaisabilité d'une formule disjonctive du  $\mu$ -calcul modal est décidable en temps linéaire.*

**Remarque.** Ce résultat peut être comparé au problème de la satisfaisabilité d'une formule du  $\mu$ -calcul quelconque qui est *EXPSPACE*-complet !



### 4.1.2 Une équivalence de tableaux

Avant de prouver, pour toute formule  $\alpha$ , l'existence d'une formule disjonctive équivalente à  $\alpha$ , on introduit une notion d'équivalence sur les tableaux nous permettant d'affirmer que les formules à partir desquels ils sont construits sont équivalentes. Bien entendu, cette condition n'est pas nécessaire, mais elle nous suffira dans la suite pour prouver le théorème principal de ce chapitre.

On montre en fait que la structure de tableau induite par les nœuds de choix et les nœuds modaux auxquels on ajoute la notion de traces, capture la sémantique des tableaux.

**4.1.2.1 Définition.** Deux tableaux  $\mathcal{T}_1$  et  $\mathcal{T}_2$  sont dits *simplement équivalents* ou plus simplement *équivalents* dans la suite lorsqu'il existe une bijection  $\mathcal{E}$  entre les nœuds de choix et les nœuds modaux de  $\mathcal{T}_1$  et  $\mathcal{T}_2$  telle que :

1. en notant  $n_0$  la racine de  $\mathcal{T}_1$ ,  $\mathcal{E}(n_0)$  est la racine de  $\mathcal{T}_2$ ,
2. pour tout nœud de choix  $n \in \mathcal{T}_1$ , pour tout nœud modal  $m \in \mathcal{T}_1$ ,  $m$  est proche de  $n$  ssi  $\mathcal{E}(m)$  est proche de  $\mathcal{E}(n)$ ,
3. pour tout nœud modal  $n \in \mathcal{T}_1$ , pour toute action  $a \in Act$ , pour tout nœud de choix  $m \in \mathcal{T}_1$ ,  $m$  est un  $a$ -fils de  $n$  ssi  $\mathcal{E}(m)$  est un  $a$ -fils de  $\mathcal{E}(n)$ ,
4. pour tout nœud modal  $n$ , les ensembles des littéraux apparaissant dans  $L_1(n)$  et dans  $L_2(\mathcal{E}(n))$  sont égaux.
5. pour tout chemin  $\mathcal{P}$  de  $\mathcal{T}_1$  il existe une  $\mu$ -trace sur  $\mathcal{P}$  ssi il existe une  $\mu$ -trace sur (le chemin induit par) l'image par  $\mathcal{E}$  (des nœuds de choix et des nœuds modaux) de  $\mathcal{P}$ .

**Remarque.** Cette définition d'équivalence n'est pas sans rappeler la définition 3.2.1.6. Dans le cas présent, la contrainte sur l'étiquetage ne porte plus que sur les littéraux étiquetant les nœuds modaux. La sémantique est cependant conservée via la conservation des types de traces.

Il vient :

**4.1.2.2 Théorème.** *Si deux formules gardées ont des tableaux équivalents alors elles sont équivalentes.*

*Preuve.* Soit  $\alpha$  et  $\beta$  deux formules gardées et  $\mathcal{T}_\alpha$  et  $\mathcal{T}_\beta$  deux tableaux équivalents pour  $\alpha$  et  $\beta$ . Soit  $\mathcal{E}$  la bijection donnant l'équivalence entre  $\mathcal{T}_\alpha$  et  $\mathcal{T}_\beta$ . On montre alors, en utilisant le théorème 3.2.2.5 que tout modèle de  $\alpha$  est un modèle de  $\beta$ , ce qui, par symétrie, nous permettra de conclure.

Soit  $S$  un modèle de  $\alpha$ . En appliquant le théorème sus-cité, il existe un marquage consistant  $M_\alpha$  de  $\mathcal{T}_\alpha$  par  $\mathcal{T}_\beta$ . Soit alors  $(e, n) \in M_\alpha$ . Il est immédiat de constater que le marquage des nœuds qui ne sont ni des nœuds de choix, ni des nœuds modaux est induit par le marquage de ces derniers. En effet, pour tout nœud  $n$  de cette sorte, pour tout état  $e \in E_S$ , on a équivalence de :

1.  $(e, n) \in M_\alpha$ ,
2. il existe un nœud de choix  $m_1$  et un nœud modal  $m_2$  tel que  $m_1$  est un parent de  $n$ , lui-même parent de  $m_2$  avec  $(e, m_1) \in M_\alpha$  et  $(e, m_2) \in M_\alpha$  (et il n'existe pas d'autres nœuds de choix ou nœuds modaux entre  $m_1$  et  $n$  et entre  $n$  et  $m_2$ ).

On définit alors un marquage  $M_\beta$  de  $\mathcal{T}_\beta$  par  $S$  comme le plus petit marquage satisfaisant :

1. pour tout nœud modal ou nœud de choix  $m$ , pour tout état  $e \in E_S$ , si  $(e, m) \in M_\alpha$  alors  $(e, \mathcal{E}(m)) \in M_\beta$ .
2. pour tout autre nœud  $n$  de  $\mathcal{T}_\beta$ , si  $n$  apparaît entre un nœud de choix  $m_1$  et un nœud modal  $m_2$  tels que  $(e, m_1)$  et  $(e, m_2) \in M_\beta$ , aucun autre nœud modal ou nœud de choix n'apparaissant entre  $m_1$  et  $n$  ni entre  $n$  et  $m_2$ , alors  $(e, n) \in M_\beta$ .

En appliquant alors la définition de l'équivalence de tableaux, on obtient immédiatement la consistance locale et globale de  $M_\beta$ , d'où le résultat.  $\square$

### 4.1.3 Existence de forme normale disjunctive

**4.1.3.1 Théorème.** *Pour toute formule positive  $\gamma$  du  $\mu$ -calcul il existe une formule disjunctive  $\hat{\gamma}$  équivalente à  $\gamma$ .*

**Remarque.** L'analyse de complexité est, à ce jour, laissé aux lecteurs intéressés. Il est clair cependant que, dans la construction proposée ci-dessous, la formule  $\hat{\gamma}$  est de taille au moins doublement exponentielle en la taille de la formule  $\gamma$ , la construction restant "élémentaire" au sens où le nombre d'exponentielles imbriquées est à l'évidence borné par trois.

**Preuve.** Sans perte de généralité, nous pouvons supposer que la formule  $\gamma$  est close, gardée, construite avec le connecteur  $a \rightarrow A$  à la place des modalités, toutes les négations ne portant que sur des constantes.

Soit  $\mathcal{T}$  un tableau régulier pour  $\gamma$ . Il en existe car la structure locale de tout tableau est définie à l'aide d'un ensemble fini de règles à instancier sur un ensemble fini d'étiquettes possibles. Un ordre arbitraire de priorité sur ces instances de règles nous donne un tableau régulier.

On cherche alors à replier le tableau  $\mathcal{T}$  en un tableau sous *forme arborescente* de telle sorte que ce graphe puisse être lu comme le tableau régulier d'une formule disjonctive  $\hat{\gamma}$  équivalente à  $\gamma$ .

Un repliage simple, reposant uniquement sur la régularité de  $\mathcal{T}$ , nous permet simplement de conserver toute l'information relative à la consistance locale. Pour conserver un critère adéquat de consistance globale la construction se révèle un peu plus complexe.

Plus précisément :

**4.1.3.2 Lemme.** *Pour le tableau  $\mathcal{T} = \langle T, L \rangle$  il existe un arbre avec arêtes retours*

$$\mathcal{T}_l = \langle T_l, L_l \rangle$$

tel que :

1.  $\mathcal{T}_l$  se développe en  $\mathcal{T}$ .
2. tout nœud de  $\mathcal{T}_l$  peut être associé à une couleur rouge ou bleu de telle sorte que pour tout chemin infini de  $\mathcal{T}_l$  il y a une  $\mu$ -trace sur ce chemin ssi le nœud le plus haut dans  $\mathcal{T}_l$  apparaissant infiniment souvent sur ce chemin est rouge.

**Remarque.** Avant de faire la preuve de ce lemme, on notera la similitude d'un arbre avec arêtes retour tel que décrit ci-dessus et la forme normale arborescente d'un automate décrite en 3.3.3.1.

*Preuve.* On construit tout d'abord un automate de mots adéquat :

**4.1.3.3 Proposition.** *Il existe un automate de mots  $\mathcal{A}$ , sur l'alphabet  $\mathcal{P}_{fin}(\mathbf{L}_\mu)$  des ensembles finis de  $\mu$ -formules, déterministe et complet, avec critère de parité qui reconnaît exactement les chemins infinis de  $\mathcal{T}$  qui ne contiennent pas de  $\mu$ -trace.*

*Preuve.* Soit  $\mathcal{A}_1$  l'automate défini en prenant :

1. comme ensemble d'états l'ensemble des couples  $(\alpha, A)$  pour  $A$  n'importe quelle étiquette de nœud du tableau  $\mathcal{T}$  et  $\alpha$  toute formule de  $A$ ,
2. comme ensemble de transitions l'ensemble des triplets de la forme  $((\alpha, A), A, (\beta, B))$  tels qu'il existe  $B_1, \dots, B_n$  tels que

$$\frac{BB_1B_2 \cdots B_n}{A}$$

soit une instance de règle dans la construction de  $\mathcal{T}$ , la formule  $\beta$  étant, dans le cas ou  $\alpha \neq \beta$ , l'une des réductions possibles, par cette instance de règle, de la formule  $\alpha$ ,

3. comme fonction de parité

$$\pi((\alpha, A)) = \begin{cases} \pi_\gamma^\mu(\gamma) & \text{si } \alpha \text{ n'est pas une variable,} \\ \pi_\gamma^\mu(X) & \text{si } \alpha = X \text{ pour une variable } X \text{ apparaissant dans } \gamma. \end{cases}$$

Il apparaît donc que l'automate  $\mathcal{A}_1$ , non déterministe, reconnaît un chemin infini de  $\mathcal{P}$  ssi ce dernier contient au moins une  $\mu$ -trace. On obtient alors l'automate  $\mathcal{A}$  désiré par complémentation et déterminisation, le critère de parité pouvant être conservé [31].  $\square$

*Preuve. Lemme 4.1.3.2 (suite).* Etant donné  $\mathcal{A}$  l'automate construit ci-dessus, on “synchronise” le tableau  $\mathcal{T}$  avec l'automate  $\mathcal{A}$ . Plus précisément on associe à chaque nœud  $n$  de  $\mathcal{T}$  un état  $S(n)$  de  $\mathcal{A}$  modélisant la lecture de  $\mathcal{T}$  par  $\mathcal{A}$ :

1. pour  $n_0$  la racine de  $\mathcal{T}$  on définit  $S(n_0)$  comme l'état initial de  $\mathcal{A}$ ,
2. pour tout nœud  $n$ , tout fils  $m$  de  $n$ , on prend  $S(m)$  égal à l'unique état de  $\mathcal{A}$  tel que  $(S(n), L(n), S(m))$  soit une transition de l'automate  $\mathcal{A}$  ( $\mathcal{A}$  est déterministe).

Avec un tel étiquetage, un chemin infini de  $\mathcal{T}$  contient une  $\mu$ -trace ssi l'indice minimum des états apparaissant sur le calcul associé est impair.

L'arbre avec arêtes retour  $\mathcal{T}_l = \langle T_l, L_l \rangle$  que nous cherchons peut alors être défini de la façon suivante :

- $T_l$  est la plus petite partie de  $T \times S(T)$  telle que :
    1.  $(n_0, s_0) \in T_l$  avec  $n_0$  (resp.  $s_0$ ) la racine de  $\mathcal{T}$  (resp. l'état initial de  $\mathcal{A}$ ),
    2. pour tout  $(n, s) \in T_l$  (avec  $S(n) = s$ ), pour tout fils  $m$  de  $n$  dans  $\mathcal{T}$  :
      - (a) soit il existe un nœud  $(n_1, s_1) \in T_l$  tel que :
        - $L(m) = L(n_1)$ ,
        - $S(m) = S(n_1)$  c'est à dire  $s_1$ ,
        - il existe un chemin (nécessairement unique) du nœud  $n_1$  au nœud  $m$  dans  $\mathcal{T}$  tel que, pour tout nœud  $n_2$  sur ce chemin,  $\pi(S(n_1)) \leq \pi(S(n_2))$ ,
- dans ce cas, on oublie le nœud  $m$  et on crée une arête retour de  $(n, s)$  vers  $(n_1, s_1)$ ,

(b) ou bien la condition précédente n'est pas remplie, on prend alors  $(m, S(m)) \in T_l$ .

- $L_l$  est définie, pour tout  $(n, s) \in T_l$ , par  $L_l(n, s) = L(n)$ .

**Remarque.** Observons tout d'abord que  $T_l$  ainsi construit est fini. En effet, le tableau  $\mathcal{T}$  est à branchements finis et l'automate  $\mathcal{A}$  est fini. Il vient donc que sur tout chemin  $\mathcal{P}$  de  $\mathcal{T}$  la condition 2a est toujours remplie avant une profondeur bornée dans l'arbre  $\mathcal{T}$ .

On vérifie aisément que le développement de  $T_l$  est bien isomorphe à  $\mathcal{T}$ . On définit alors la couleur d'un nœud  $(n, s) \in T_l$  sur lequel pointe une arête de retour comme étant *rouge* si l'indice de l'état  $s$  and  $\mathcal{A}$  est impair, *bleu* sinon, tous les nœuds qui ne sont pas de retour étant coloriés en bleu.

Un tel coloriage remplit bien la seconde condition énoncée, i.e. sur le développement de  $T_l$  (c'est à dire  $\mathcal{T}$ ) un chemin infini contient une  $\mu$ -trace ssi le nœud le plus haut dans  $T_l$  apparaissant infiniment souvent est un nœud *rouge*.  $\square$

*Preuve du théorème 4.1.3.1 (suite).* A partir de cet arbre avec arêtes retours  $T_l$  on construit une formule disjonctive  $\widehat{\gamma}$  qui a un tableau équivalent à  $\mathcal{T}$ . Pour ce faire, on suppose qu'on dispose pour chaque nœud  $(n, s) \in T_l$  d'une variable  $X_{n,s}$ .

On construit alors la formule recherchée en associant à chaque nœud  $(n, s) \in T_l$  tel que  $L(n) \neq \emptyset$  une formule  $(\widehat{n, s})$  inductivement définie, des "feuilles" vers la racine, par :

1. si  $(n, s)$  est une feuille alors  $(n, s)$  n'est pas la cible d'une arête retour (car la formule  $\gamma$  est gardée); en ce cas :
  - (a) soit  $(n, s)$  est la source d'une arête retour vers un nœud  $(n_1, s_1)$ , on pose donc  $(\widehat{n, s}) = X_{n_1, s_1}$
  - (b) soit  $L(n) \neq \emptyset$  et  $(n, s)$  n'est pas la source d'une arête retour, c'est donc que  $n$  est un nœud modal et que  $L(n)$  ne contient que des littéraux constants (de la forme  $p$  ou  $\neg p$ ); étant donné  $l_1, l_2, \dots, l_n$  ces littéraux on pose  $(\widehat{n, s}) = \bigwedge_{i \in \{1, \dots, n\}} l_i$ ,
  - (c) soit  $L(n) = \emptyset$  ( $n$  est le  $a$ -fils d'un nœud  $m$  avec  $a \rightarrow A \in L(m)$  ssi  $A = \emptyset$ ), on pose temporairement  $(\widehat{n, s}) = *$ ,
2. lorsque l'une des règles  $(\mu)$ ,  $(\nu)$  ou  $(and)$  est appliquée alors  $(n, s)$  admet un fils unique  $(n_1, s_1)$  et on pose  $(\widehat{n, s}) = \sigma_{n,s} X_{n,s} (\widehat{n_1, s_1})$  où  $\sigma_{n,s} = \mu$  ou  $\nu$  selon que le nœud dans  $T_l$  est rouge ou bleu,
3. lorsque la règle  $(ou)$  est appliquée  $(n, s)$  admet deux fils  $v_1$  et  $v_2$ , on pose donc  $(\widehat{n, s}) = \sigma_{n,s} X_{n,s} \widehat{v_1} \vee \widehat{v_2}$ , selon la même convention que ci-dessus pour  $\sigma_{n,s}$ ,

4. lorsque la règle (*mod*) est appliquée, i.e.  $n$  est un nœud modal selon la typologie 3.2.1.5, soit  $Act_1$  l'ensemble des actions  $a \in Act$  telle que  $(n, s)$  admet au moins un  $a$ -fils, et pour toute action  $a \in Act_1$ ,  $A_a$  l'ensemble des formules  $\widehat{v}$  des  $a$ -fils  $v$  de  $(n, s)$ , en posant  $A_a = \emptyset$  si  $\widehat{v} = *$  pour un  $a$ -fils  $v$  de  $(n, s)$ ; on pose alors

$$\widehat{(n, s)} = \sigma X_{n,s} \bigwedge B \wedge \bigwedge_{a \in Act'} a \rightarrow A_a$$

où  $B$  est l'ensemble des littéraux constants apparaissant dans  $L(n)$  et  $\sigma_{n,s}$  est défini comme précédemment .

A la suite de cette construction, comme  $\gamma$  est close, pour la racine  $u_0$  de  $\mathcal{T}_l$ , la formule  $\widehat{u}_0$ , disjonctive, est elle-même close. On pose alors  $\widehat{\gamma} = \widehat{u}_0$ .

Soit alors  $\widehat{\mathcal{T}}$  un tableau pour  $\widehat{\gamma}$ . Il est alors facile de construire, en s'aidant de la définition des  $\widehat{u}$  pour tout nœud  $u \in \mathcal{T}_l$ , une fonction  $\mathcal{E} : \widehat{\mathcal{T}}$  dans  $\mathcal{T}$  prouvant, au sens de la définition 4.1.2.1 que les tableaux  $\widehat{\mathcal{T}}$  et  $\mathcal{T}$  sont équivalents ce qui, par application du théorème 4.1.2.2, nous prouve l'équivalence de  $\gamma$  et  $\widehat{\gamma}$ .  $\square$

En particulier, cette réduction de toute formule positive du  $\mu$ -calcul à une formule disjonctive équivalente nous donne :

**4.1.3.4 Corollaire.** *La satisfaisabilité d'une formule positive du  $\mu$ -calcul est décidable. De plus, toute formule positive du  $\mu$ -calcul satisfaisable admet un modèle fini.*

*Preuve.* Conséquence immédiate des théorèmes 4.1.1.2 et 4.1.3.1.  $\square$

## 4.2 Automates modaux

Dans cette section, on définit la notion d'automate modal non alternant qu'on montre équivalente, en termes de pouvoir expressif, au  $\mu$ -calcul modal.

L'idée de construire des automates pouvant "lire" directement des systèmes de transitions est déjà présente dans les travaux de Bernholtz et Grumberg [8] mais la construction proposée traite seulement de la logique temporelle arborescente et nous conduirait à une notion d'automate trop expressive dans le cas général.

Dans cette section  $\Sigma_p$  désigne, comme dans le cas des automates modaux alternants, l'ensemble  $\mathcal{P}(Prop_A)$  pour  $Prop_A$  une partie finie de  $Prop$ .

### 4.2.1 Définitions des automates

**4.2.1.1 Définition.** Un *automate modal* est un  $n$ -uplet

$$A = \langle Q, q_0, \Sigma_p, \delta, \Omega \rangle$$

pour :

1. un ensemble fini d'états  $Q$ ,
2. un état initial  $q_0$ ,
3. un ensemble fini d'étiquette  $\Sigma_p$ ,
4. une fonction de transition  $\delta : Q \times \Sigma_p \rightarrow \mathcal{P}(Act \rightarrow \mathcal{P}(Q))$ ,
5. une fonction de parité,  $\Omega : Q \rightarrow \mathbb{N}$ .

**4.2.1.2 Définition.** Pour un système de transitions  $S$  et un automate modal  $A$  tel que décrit ci-dessus, un *calcul* de  $A$  sur  $S$  est un arbre fini ou infini  $T$  dont les nœuds sont étiquetés sur  $(Q, E_S)$  satisfaisant les conditions suivantes :

- La racine est étiquetée par  $(q_0, r_S)$ .
- Pour tout nœud  $n$  de  $T$  étiqueté par  $(q, e)$  il existe une transition  $f \in \delta(q, \rho_S(e) \cap Prop_A)$  telle que pour toute action  $a$  :
  - pour tout état  $q_a \in f(a)$ ,  $n$  admet un fils étiqueté par  $(q_a, t)$  pour un certain  $t \in E_S$  tel que  $(s, t) \in R(a)$ ,
  - pour tout  $t \in E_S$  tel que  $(s, t) \in R(a)$  il existe un état  $q_a \in f(a)$  de l'automate tel que  $n$  admette un fils étiqueté  $(q_a, t)$ .

Un calcul est acceptant ssi pour tout chemin  $\mathcal{P}$  de ce calcul, le minimum des indices  $\Omega(q)$  pour  $q$  apparaissant infiniment souvent sur ce chemin est pair.

**Remarque.** Etant donné un automate  $A$  et un système de transitions  $S$ , on peut associer à chaque état  $q \in Q$  le prédicat  $P_q(x)$  défini pour tout état  $e \in E_S$  par  $P_q(e)$  ssi  $S[e/r_S] \in L_q(A)$ . Lors de la lecture d'un sommet  $e \in E_S$  à partir d'un état  $q \in Q$ , le choix d'une transition  $f \in \delta(q, \rho_S(e))$  revient à choisir, pour toute action  $a \in Act$ , le "type" modal (en un sens proche de celui défini en 1.2.1.2)

$$T_f = \{P_{q_1}(x) \mid q_1 \in f(a)\}$$

de telle sorte que l'ensemble  $Succ_a(e)$  satisfasse ce type. Une différence majeure cependant avec la définition 1.2.1.2, les types manipulés ici ne sont pas forcément disjoints. Une telle

contrainte, dans le cas des automates modaux, reviendrait à manipuler des automates non ambigüs dont on sait [46], dans le cas des arbres binaires, qu'ils sont strictements moins expressifs que les automates ambigüs.

Nous allons prouver que le pouvoir expressif de ces automates est exactement le pouvoir expressif du  $\mu$ -calcul.

## 4.2.2 Automates modaux et $\mu$ -formules modales

**4.2.2.1 Théorème.** *Pour toute formule positive  $\alpha$  du  $\mu$ -calcul modal il existe un automate modal équivalent.*

*Preuve.* On suppose tout d'abord que  $\alpha$  est disjonctive. Sans perte de généralité, on peut toujours supposer que toute les conjonctions apparaissant dans  $\alpha$  sont “complètes” vis-à-vis de l'alphabet  $Act$  au sens où elles sont de la forme :

$$l_1 \wedge l_2 \wedge \cdots \wedge l_n \wedge \bigwedge_{a \in Act} a \rightarrow A_a$$

pour  $l_1, \dots, l_n$  des littéraux et, pour toute action  $a \in Act$ ,  $A_a$  un ensemble fini de formules disjonctives.

En effet, si ce n'est pas le cas, on peut toujours “compléter” chaque conjonction par des formules de la forme  $a \rightarrow \emptyset \vee a \rightarrow \{\top\}$  avec  $\top \stackrel{\text{def}}{=} \nu X. \bigwedge_{a \in Act} a \rightarrow \emptyset \vee a \rightarrow X$ . Quitte à distribuer les disjonctions sur les conjonctions, on obtient bien une formule de la forme désirée.

Soit alors  $\mathcal{T}_l$  un arbre avec arêtes retour se développant en un tableau  $\mathcal{T}$  pour  $\alpha$ . On pose :  $A = \langle Q, q_0, \Sigma_p, \delta, \Omega \rangle$ , avec :

- $Q = (CN \times BV)$  où  $CN$  est l'ensemble des nœuds de choix de  $\mathcal{T}_l$  (cf. définition 3.2.1.5, et  $BV$  est l'ensemble des variables liées de  $\alpha$  augmenté du symbole  $*$ ,
- $q_0$  est la racine de  $\mathcal{T}_l$  associée au symbole  $*$ ,
- $\Sigma_p$  est l'ensemble des parties de l'ensemble  $Prop_\alpha$  des symboles propositionnels apparaissant dans  $\alpha$ ,
- pour toute partie  $P \subseteq Prop_\alpha$ ,  $\delta((n, X), P)$  est l'ensemble des fonctions  $f_{m,Y}$  (une pour chaque nœud modal  $m$  proche de  $n$ ) telles que :
  - tout symbole propositionnel apparaissant dans  $L(m)$  appartient à  $P$ ,
  - aucun symbole propositionnel apparaissant nié dans  $L(m)$  n'appartient à  $P$ <sup>3</sup>,

<sup>3</sup>ainsi, lorsque  $m$  est localement inconsistant, aucune transition n'est construite.



$Y$  étant la plus petite variable (au sens de l'ordre de dépendance) qui est régénérée sur le chemin (sans cycles car  $\alpha$  est gardée) du nœud  $n$  au nœud  $m$ , ou bien  $*$  s'il n'existe pas une telle variable, avec, pour toute action  $a \in \Sigma_a$ , la fonction  $f_{m,Y}(a)$  définie par cas :

- si  $a \rightarrow \emptyset \in L(m)$  alors  $f_{m,Y}(a) = \emptyset$ ,
- si le nœud  $m$  admet un  $a$ -fils alors  $f_{m,Y}(a) = \{(n', Y) | n' \text{ est un } a\text{-fils de } m\}$ ,

la condition préalable de “complétude” des conjonctions nous assurant qu'aucun autre cas n'est possible,

- $\Omega((n, X))$  est égal au  $\nu$ -indice de  $X$  dans  $\alpha$  (voir def. 3.1.2.4)  $\pi_\alpha^\nu(X)$  si  $X$  est effectivement une variable, égale au maximum des  $\nu$ -indices des variables liées dans  $\alpha$  sinon.

On vérifie alors facilement qu'un marquage consistant du tableau  $\mathcal{T}$  par un système  $S$  induit un calcul acceptant de l'automate ci-dessus. Réciproquement, un calcul acceptant de  $A$  sur un système  $S$  induit un marquage consistant de  $\mathcal{T}_l$  (puis  $\mathcal{T}$ ) par  $S$ . Le théorème 3.2.2.5 s'applique donc, nous donnant l'équivalence sémantique de la formule disjonctive  $\alpha$  avec l'automate ainsi construit. Le théorème 4.1.3.1 prouvant pour toute formule positive  $\alpha$  du  $\mu$ -calcul l'existence d'une forme disjonctive équivalente nous permet de conclure.  $\square$

**4.2.2.2 Corollaire (Simulation des automates alternants).** *Pour tout automate modal alternant  $A_{alt}$  il existe un automate modal (non alternant)  $A$  tel que  $L(A_{alt}) = L(A)$ .*

*Preuve.* C'est en effet une conséquence immédiate du théorème 3.3.4.1 et du théorème ci-dessus.  $\square$

**4.2.2.3 Théorème.** *A tout automate modal  $A$  on peut associer une formule  $\alpha$  (disjonctive) du  $\mu$ -calcul qui lui est équivalente.*

*Preuve.* C'est ici un cas particulier de la preuve du théorème 3.3.4.1. Remarquons tout de même que dans le cas des automates modaux non alternants, on obtient une formule disjonctive, et la preuve de leur équivalence sémantique se simplifie considérablement.  $\square$

**4.2.2.4 Corollaire.** *L'ensemble des langages de systèmes (ou d'arbres) de transitions reconnaissables au sens des automates modaux est clos par complémentation.*

### 4.3 Restrictions au cas des arbres binaires infinis

Dans cette section, on montre comment les résultats ci-dessus s'appliquent au cas des arbres binaires. En particulier, on montre que le théorème de complémentation de Rabin [40] en est une conséquence immédiate.

On suppose que  $Act$  contient au moins deux actions distinctes notées  $d$  et  $g$ . On note alors  $T = \{d, g\}^*$  l'ensemble des mots finis sur l'alphabet  $\{d, g\}$ . On dit aussi que  $u \in T$  est *préfixe* de  $v \in T$ , ce qui est noté  $u \leq_{pre} v$ , lorsqu'il existe  $w \in T$  tel que  $u.w = v$ . On vérifie que la relation  $\leq_{pre}$  ainsi définie est bien un ordre sur  $T$  appelé *ordre préfixe*.

#### 4.3.1 Définitions et résultats classiques sur les arbres binaires

Dans cette section, on se place dans le cas où  $Prop$  est fini.

**4.3.1.1 Définition.** Un *arbre binaire infini*, appelé aussi *arbre binaire*, est un arbre de transitions dont tout état admet exactement deux successeurs, un accessible via une transition étiquetée  $d \in Act$ , l'autre accessible via une transition étiquetée par  $g \in Act$ .

**Remarque.** Pour tout arbre binaire  $S$ , en identifiant les sommets de  $S$  avec les mots de  $T = \{d, g\}^*$  avec :

1. la racine  $r_S = \epsilon$  (le mot vide),
2. pour tout sommet  $u \in T$ ,  $(u, u.g) \in R_S(g)$  (resp.  $(u, u.d) \in R_S(d)$ ) où le sommet  $u.g$  (resp.  $u.d$ ) est appelé *fil droit* (resp. *fil gauche*) du sommet  $u$ ,

l'arbre binaire  $S$  est complètement déterminé par sa fonction d'étiquetage des états  $\rho_S : T \rightarrow \mathcal{P}(Prop)$ . Dans la suite, on pourra identifier modulo équivalence par isomorphisme la classe des arbres binaires infinis et l'ensemble des fonctions de  $T$  dans  $\mathcal{P}(Prop)$ . C'est cette définition d'arbres binaires qui est généralement utilisée, e.g. dans [40].

Cette définition des arbres binaires comme un cas particulier des arbres de transitions nous conduit à voir les automates d'arbres binaires comme des cas particuliers d'automates modaux. Plus précisément :

**4.3.1.2 Définition.** Un *automate d'arbre binaire* est un automate modal

$$A = \langle Q, q_0, \mathcal{P}(Prop), \delta, \Omega \rangle$$

dont la fonction de transition satisfait la propriété suivante : pour tout état  $q$ , pour toute partie  $P \subseteq Prop$ , pour toute fonction  $f : Act \rightarrow \mathcal{P}(Q)$  de  $\delta(q, P)$  on a, pour toute action  $a \in Act$ ,  $f(a) \neq \emptyset$  ssi  $a \in \{d, g\}$  et, dans ce cas,  $f(a)$  est un singleton.

On dit alors, comme dans le cas des automates modaux, qu'un arbre binaire  $S$  est reconnu par un automate binaire  $A$  ssi il existe un calcul acceptant de l'automate  $A$  sur l'arbre  $S$  au sens de la définition 4.2.1.2.

**Remarque.** On pourra utiliser dans la suite des fonctions de transitions  $\delta : Q \times \Sigma_p \rightarrow \mathcal{P}(Q \times Q)$ , en reconstruisant, pour tout état  $q$ , pour tout  $P \in \Sigma_p$ , toute paire  $(q_1, q_2) \in \delta(q, P)$  la "transition"  $f_{q_1, q_2}$  définie, pour toute action  $a \in Act$ , par  $f_{q_1, q_2}(a) \neq \emptyset$  ssi  $a = g$ , dans ce cas  $f_{q_1, q_2}(a) = q_1$ , ou bien  $a = d$  et dans ce cas  $f_{q_1, q_2}(a) = q_2$ . On retrouve ainsi la définition usuelle des automates d'arbres binaires.

La théorie des automates et les travaux de Rabin nous assurent que cette notion d'automate capture la sémantique de la logique monadique à deux successeurs. Plus précisément :

**4.3.1.3 Théorème (Rabin et al.).** *Pour toute MS-formule  $\varphi$  sur les arbres binaires il existe un automate d'arbres binaires  $A$  tel que, pour tout arbre binaire  $S$ ,  $S \models \varphi$  ssi  $S \in L(A)$ .*

*Preuve.* Voir les travaux de Rabin [40] connectant la logique monadique du second ordre sur les arbres binaires et la notion d'automates binaires, les travaux de Mostowski [32] nous permettant alors de nous ramener au cas d'automates avec critère de parité. Une étape importante de la preuve de Rabin est le théorème de complémentation :

**4.3.1.4 Théorème (Complémentation).** *Pour tout automate d'arbres binaires  $A$  il existe un automate d'arbre binaire  $A_c$  tel que, pour tout arbre binaire  $S$ ,  $S \in L(A_c)$  ssi  $S \notin L(A)$ .*

*Preuve.* Il y a plusieurs preuves de ce théorème réputé difficile [40, 17, 20, 33, 34, 5]. On en donne une nouvelle dans la partie suivante.

## 4.3.2 Une preuve du théorème de complémentation

**4.3.2.1 Définition.** Une formule disjonctive  $\alpha$  est appelée *formule disjonctive binaire* lorsque toutes les conjonctions apparaissant dans  $\alpha$  sont de la forme

$$d \rightarrow \{\alpha_1\} \wedge g \rightarrow \{\alpha_2\} \wedge \bigwedge L$$

pour  $\alpha_1$  et  $\alpha_2$  deux  $\mu$ -formules et  $L$  un ensemble (fini) de littéraux.

Il vient :

**4.3.2.2 Proposition.** *Pour tout automate d'arbres binaires  $A$  la formule disjonctive binaire  $\alpha$  obtenue à partir de  $A$  selon la preuve du théorème 4.2.2.3 est une formule disjonctive binaire et, réciproquement, pour toute formule disjonctive binaire  $\alpha$  l'automate  $A$*

extrait de  $\alpha$  selon la preuve du théorème 4.2.2.1 est un automate d'arbres binaires.

*Preuve.* Par construction. □

La preuve du théorème de complémentation de Rabin apparaît alors comme une conséquence immédiate du lemme suivant :

**4.3.2.3 Lemme.** *Pour toute formule disjonctive binaire  $\alpha$  il existe une formule disjonctive binaire  $\alpha_c$  telle que, pour tout arbre binaire  $S$ ,  $S \models \alpha_c$  ssi  $S \models \neg\alpha$ .*

*Preuve.* Soit  $\alpha$  une formule disjonctive binaire. Remarquons tout d'abord que la formule  $\alpha_c$  souhaitée n'est pas nécessairement équivalente à  $\neg\alpha$  sur la classe des systèmes de transitions. On requiert seulement qu'elle lui soit équivalente sur l'ensemble des arbres binaires. On peut alors utiliser le résultat suivant :

**4.3.2.4 Proposition.** *Pour toute formule  $\alpha$  du  $\mu$ -calcul, toute action  $a \in \{d, g\}$ , tout ensemble fini  $A$  de formules, les formules :*

1.  $\langle a \rangle\alpha$  et  $[a]\alpha$ ,
2.  $\neg a \rightarrow \{\alpha\}$  et  $a \rightarrow \{\neg\alpha\}$ ,
3.  $a \rightarrow A$  et  $a \rightarrow \{\bigwedge A\}$ .

sont équivalentes sur l'ensemble des arbres binaires.

*Preuve.* Immédiate. □

Ainsi, à partir de  $\neg\alpha$ , les lois 2 et 3 ajoutées aux lois de De Morgan nous permettent de supprimer toutes les négations obtenant ainsi une formule dont toutes les "modalités" sont de la forme  $a \rightarrow \{\alpha\}$  pour  $a \in \{d, g\}$ . Quitte alors à compléter chaque modalité  $a \rightarrow \alpha$  en  $a \rightarrow \alpha \wedge b \rightarrow \top$  (lorsque  $\{a, b\} = \{d, g\}$ ), on obtient une formule  $\alpha_1$  qui nous permet de conclure. En effet :

1. par construction, la formule  $\alpha_1$  est, sur les arbres binaires, équivalente à la formule  $\neg\alpha$ ,
2. la forme normale disjonctive de  $\alpha_1$  construite comme dans la preuve du théorème 4.1.3.1 est bien une formule disjonctive binaire; lorsque toutes les formules apparaissant dans  $\{C\}$  sont, soit des littéraux, soit des formules de la forme  $a \rightarrow \{\alpha\}$ , la règle (*mod*) se simplifie en effet en :
 
$$\frac{\{\alpha \mid a \rightarrow \{\alpha\} \in \{C\}\}}{\{C\}}$$
 pour toute action  $\alpha \in Act$  telle qu'il existe  $\alpha$  avec  $a \rightarrow \{\alpha\} \in C$

□

## Chapitre 5

# Complétude expressive du $\mu$ -calcul

Dans ce chapitre, on étudie la sémantique de la logique monadique du second ordre sur les arbres de transitions afin de pouvoir comparer le pouvoir expressif de cette logique avec le  $\mu$ -calcul modal.

Plus précisément, on montre qu'une formule de logique monadique du second ordre est équivalente à une formule du  $\mu$ -calcul modal si et seulement si sa sémantique est close par équivalence de bisimulation.

Les formules du  $\mu$ -calcul ayant une sémantique définissable en logique monadique du second ordre, ce résultat apparaît en fait comme une caractérisation du pouvoir expressif du  $\mu$ -calcul : c'est exactement l'ensemble des formules de logique monadique du second ordre dont la sémantique est close par équivalence de bisimulation.

Notre preuve est effective au sens où on démontre en particulier qu'il est décidable de savoir si une formule de *LMSO* admet comme modèle une classe d'arbre de transition close par bisimulation.

**Remarque.** Dans sa forme actuelle, ce résultat est limité au cas des systèmes de transitions dont l'ensemble des sommets est au plus dénombrable. Un travail postérieur à la rédaction de cette thèse [28], en collaboration avec Igor Walukiewicz, qui repose largement sur les développements récents de la théorie des automates sur les structures arborescentes [48], en offre une preuve alternative qui s'applique aux cas général de tous les systèmes de transitions.

## 5.1 Présentation lexicographique des arbres

On s'intéresse ici à une forme particulière d'arbres, appelés lexiques, qui peuvent être vus comme des arbres de cardinalité au plus dénombrable, dont les fils de chaque sommet sont ordonnés de telle sorte qu'ils forment un mot fini ou infini. Cette étude constitue l'une des approches possibles pour prouver le résultat principal de ce chapitre : une caractérisation algébrique du pouvoir expressif du  $\mu$ -calcul modal par rapport à la logique monadique du second ordre.

Dans tout ce qui suit :

- $\Sigma$  désigne l'ensemble  $Act \times \mathcal{P}(Prop)$  pour  $Act$  un ensemble fini d'actions et  $Prop$  un ensemble fini de constantes propositionnelles,
- $\Sigma_*$  désigne l'ensemble  $\Sigma \cup \{*\}$  pour  $*$  une étiquette n'apparaissant pas dans  $\Sigma$ .

### 5.1.1 Lexiques et arbres de transitions induits

**5.1.1.1 Définition.** Etant donné  $S = \langle E, r, R, \rho \rangle$  un arbre de transitions, on appelle *présentation lexicographique* de  $S$  une relation d'ordre  $\leq_S$  satisfaisant les propriétés suivantes :

1. la relation  $\leq_S$  est locale, i.e. pour tous sommets  $e$  et  $f \in E_S$ , si  $e \leq_S f$  alors il existe  $g \in E_S$  tel que  $e \in \text{Succ}(g)$  et  $f \in \text{Succ}(g)$ ,
2. pour tout sommet  $e \in E_S$ , l'ensemble  $\text{Succ}(e)$ , totalement ordonné par  $\leq_S$ , est isomorphe à une partie de  $\mathbb{N}$ .

**Remarque.** Si  $S$  admet une présentation lexicographique alors l'ensemble de ses états est au plus dénombrable.

Dans la suite, on vas caractériser à l'aide d'une notion d'automate la logique monadique sur les arbres de transitions munie de présentation lexicographique. Pour ce faire, on introduit la notion de lexique qui apparaît comme une définition possible de la structure logique sous-jacente aux arbres de transitions ainsi présentés.

**5.1.1.2 Définition.** Un *lexique* est un  $n$ -uplet

$$L = \langle E, r, R, O, \rho \rangle$$

pour :

1. un ensemble de sommets,
2.  $r$  une racine,

3.  $R$  une relation sur  $E \times E$  telle que le graphe  $\langle E, R \rangle$  est un arbre de racine  $r$ ,
4.  $O$  une présentation lexicographique de cet arbre.

On dit que  $e$  est un voisin gauche de  $f$  lorsque  $(e, f) \in O$ .

**Notation.** De façon analogue aux systèmes de transitions on note, pour tout sommet  $e \in E$ ,

$$\text{Succ}(e) = \{f \in E \mid (e, f) \in R\}$$

On note aussi  $\text{MSucc}(e)$  l'ensemble  $\text{Succ}(e)$  ordonnée par  $O$  sur  $\text{Succ}(e)$ . Par définition, l'ensemble  $\text{Succ}(e)$  ainsi ordonnée apparaît comme un mot, fini ou infini, sur l'alphabet  $E$ . On considère donc que  $\text{MSucc}$  est une fonction de  $E$  dans  $E^\infty$ .

**Remarque.** La donnée de  $\text{MSucc}$  définit, sans ambiguïté possible, les relations  $R$  et  $O$ . Dans la suite, on convient donc, aussi souvent que nécessaire, de considérer un lexique comme un quadruplet :

$$L = \langle E, r, \text{MSucc}, \rho \rangle$$

La logique monadique du second ordre sur les lexiques restant bien entendu définie sur le vocabulaire formé des deux relations  $R$  et  $O$ . La fonction  $\text{MSucc}$  étant définissable *LMSO* sur ce vocabulaire, on pourra tout aussi bien l'utiliser dans les formules de *LMSO*.

**5.1.1.3 Proposition.** *Etant donné  $a_0$  une action arbitrairement choisie, il existe une transduction MS-définissable bijective entre l'ensemble des lexiques dont la racine porte une étiquette de la forme  $(a_0, P)$  et l'ensemble des arbres de transitions munis d'une présentation lexicographique.*

*Preuve.* Pour une définition et pour les propriétés générales des transductions, on pourra consulter le tour d'horizon de Courcelle [11]. Pour tout lexique  $L = \langle E, r, \text{MSucc}, \rho \rangle$  on définit l'arbre de transitions  $\text{ArT}(L) = \langle E_1, r_1, R_1, \rho_1 \rangle$  induit par  $L$  en prenant :

1. l'ensemble de sommets  $E_1 = E$ ,
2. le sommet initial  $r_1 = r$ ,
3. la fonction de transition  $R_1$ , définie pour toute action  $a \in \text{Act}$ , tous sommets  $e$  et  $f \in E$  par  $(e, f) \in R_1(a)$  ssi  $f \in \text{Succ}(e)$  avec  $\pi_1(\rho(f)) = a$ , i.e. la première projection de  $\rho(f)$ ,
4. la fonction d'étiquetage  $\rho_1 : E \rightarrow \mathcal{P}(\text{Prop})$  définie pour tout sommet  $e \in E_1$  par  $\rho_1(e) = \pi_2(\rho(e))$ , i.e. la seconde projection de  $\rho(e)$ .

En prenant alors comme relation d'ordre  $\leq_{ArT(L)}$  la relation de voisinage gauche induite par  $L$  il est immédiat que c'est une présentation lexicographique de  $S$ . A l'évidence, cette construction est une bijection, l'action  $a_0$ , n'apparaissant pas dans  $ArT(L)$  devant effectivement être fixée. C'est bien une transduction définissable en  $LMSO$  car tous les composants de  $ArT(L)$  et la relation  $\leq_{ArT(L)}$  sont définissables à l'aide de  $MS$ -formules.  $\square$

**Notation.** Dans la suite, pour tout arbre de transitions  $S$  de cardinalité au plus dénombrable, toute présentation lexicographique  $\leq_S$  de  $S$ , on note  $L(S, \leq_S)$  le lexique défini comme ci-dessus, l'action  $a_0$  étiquetant la racine étant fixée une fois pour toutes.

### 5.1.2 Codage binaire des lexiques

Dans la suite, on note  $T = \{d, g\}^*$ . On dit qu'un mot  $u \in T$  est *préfixe* d'un mot  $v \in T$ , ce que l'on note  $u \leq_{pref} v$ , lorsqu'il existe  $w \in T$  tel que  $u.w = v$ . On note aussi  $u \leq_{lex} v$  le fait que  $u$  est plus petit que  $v$  selon l'ordre lexicographique sur  $T$  en posant  $g < d$ .

**5.1.2.1 Définition.** Un *arbre binaire de codage* ou plus simplement arbre de codage est une fonction  $V : T \rightarrow \Sigma_*$  telle que  $V(\epsilon) \neq *$ .

Un arbre de codage  $V$  est appelé *arbre gauche* lorsque pour tout  $u \in T$  :

1. si  $V(u.g) = *$  alors pour tout  $v \neq \epsilon$ ,  $V(u.v) = *$ ,
2. si  $V(u.g) \neq *$  alors soit  $V(u.d) \neq *$ , soit  $V(u.d.g) \neq *$ .

**5.1.2.2 Définition.** Pour tout arbre de codage gauche  $V$ , on appelle *L-sommet* tout sommet  $u \in T$  tel que  $V(u) \neq *$ , et on appelle *\*-sommet* tout sommet  $u \in T$  tel que  $V(u) = *$ . On définit alors le *lexique induit* par  $V$  en posant :

$$Lex(V) = \langle E_V, r_V, MSucc_V, \rho_V \rangle$$

avec :

1. l'ensemble d'états  $E_V = T - V^{-1}(*)$ ,
2. la racine  $r_V = \epsilon$ ,
3. pour tout sommet  $u \in E_V$ ,  $MSucc_V(u) = v_0.v_1.\dots$ , le mot composé de tous les sommets  $v \in E_V$  tels que pour tout sommet  $w \in T$ , si  $u <_{pref} w <_{pref} v$ , alors  $V(w) = *$ , et satisfaisant, pour tous indices  $i$  et  $j$ ,  $i \leq j$  ssi  $v_i \leq_{lex} v_j$ ,



4. pour tout sommet  $u \in E$ ,  $\rho_V(u) = V(u)$ .

Pour tout  $L$ -sommet  $u \in T$ , on note aussi  $\text{Succ}_V(u)$  l'ensemble des  $L$ -sommets qui sont successeurs de  $u$  dans  $\text{Lex}(V)$ . Les fonctions  $\text{Succ}_V$  et  $\text{MSucc}_V$  s'étendent sans difficulté à tout sommet  $u \in T$ , la définition ci-dessus ne faisant pas intervenir le fait que  $u$  est un  $L$ -sommet.

**5.1.2.3 Proposition.** *La fonction  $\text{Lex}$  est une transduction bijective, définissable en logique monadique du second ordre, entre l'ensemble des arbres binaires de codage gauche et l'ensemble des lexiques.*

*Preuve.* Par construction  $\text{Lex}$  est bien une transduction définissable, chaque élément du lexique  $\text{Lex}(V)$  étant définissable, à partir des sommets de  $T$  et de  $V$ , en logique monadique du second ordre. Montrons que c'est une bijection. Pour ce faire, on explicite sa réciproque notée  $\text{CodG}$ .

Soit en effet  $L = \langle E, r, \text{MSucc}, \rho \rangle$  un lexique. On définit les *coordonnées binaires gauches* de tout sommet  $e \in E$  comme l'unique mot  $c(e) \in T$  inductivement défini par :

1.  $c(r) = \epsilon$ ,
2. pour tout sommet  $e \in E$  :
  - (a) si  $\text{MSucc}(e) = \epsilon$  l'induction termine,
  - (b) si  $\text{MSucc}(e) = e_0$  on pose  $c(e_0) = c(e).g$ ,
  - (c) si  $\text{MSucc}(e) = e_0.e_1.\dots.e_{n+1}$ , pour tout  $i \in \{0, \dots, n\}$  on pose  $c(e_i) = c(e).d^i.g$  et on pose aussi  $c(e_{n+1}) = c(e).d^n$ ,
  - (d) si  $\text{MSucc}(e) = e_0.e_1.e_2.\dots$ , pour tout  $i \in \mathbb{N}$  on pose  $c(e_i) = c(e).d^i.g$ ,

On définit alors  $\text{CodG}(L) : T \rightarrow \Sigma_*$  par :

1. si  $u = \epsilon$  alors  $\text{CodG}(L)(u) = (a_0, \rho(r))$ ,
2. si  $u \neq \epsilon$  :
  - (a) s'il existe un sommet  $e \in E$  tel que  $c(e) = u$  alors, en notant  $a \in \text{Act}$  l'action étiquetant l'unique transition entrante dans le sommet  $e$ ,  $V(u) = (a, \rho(e))$ ,
  - (b) sinon  $V(u) = *$ .

Par construction,  $\text{CodG}(L)$  est bien un arbre gauche avec  $\text{Lex}(\text{CodG}(L))$  isomorphe à  $L$ . De surcroît, on vérifie facilement que pour tout arbre gauche  $V$ , on a bien

$$V = \text{CodG}(\text{Lex}(V))$$

□

L'existence de la transduction  $Lex$ ,  $MS$ -définissable, nous permet de traduire toute formule sur les lexiques en une formule sur les arbres de codage gauche et réciproquement. Plus précisément :

**5.1.2.4 Proposition.** *Pour toute formule de la logique monadique du second ordre  $\varphi$  sur le vocabulaire des lexiques (resp. sur le vocabulaire des systèmes de transitions) il existe une formule  $\psi$  sur les arbres de codage telle que, pour tout arbre de codage  $V$ ,  $V \models \psi$  ssi  $Lex(V) \models \varphi$  (resp.  $ArT(Lex(V)) \models \varphi$ ).*

*Preuve.* C'est une conséquence immédiate et classique de la théorie des modèles [11, 24]. La formule  $\psi$  s'obtient en effet à partir de  $\varphi$  en remplaçant tout symbole du vocabulaire des lexiques (resp. des systèmes de transitions) par son interprétation sur les arbres de codages (resp. par son interprétation sur les arbres de codages via son interprétation sur les lexiques). □

**5.1.2.5 Corollaire.** *Toute formule satisfaisable de la logique monadique du second ordre sur les lexiques (resp. les arbres de transitions) admettant un modèle (resp. un modèle de cardinalité au plus dénombrable) admet un modèle régulier.*

*Preuve.* C'est une conséquence immédiate de la proposition précédente et du théorème de Rabin sachant qu'un automate d'arbres binaires dont le langage associé n'est pas vide admet un moins un arbre régulier. On vérifie alors sans peine que si  $V$  est un arbre de codage régulier alors  $Lex(V)$  et  $ArT(Lex(V))$  sont réguliers. □

### 5.1.3 Automates sur les lexiques

Dans cette partie on caractérise le pouvoir d'expression de la logique monadique du second ordre sur les lexiques à l'aide d'une notion appropriée d'automates. La preuve proposée, qui repose sur le théorème de Rabin, passe par une extraction d'automates sur les lexiques à partir d'automates d'arbres binaires sur les codages de lexiques.

**5.1.3.1 Définition ( $L$ -automates).** Un  $L$ -*automate* est un quadruplet de la forme

$$A = \langle Q, q_0, \delta, \Omega \rangle$$

avec un ensemble fini d'états  $Q$ , un état initial  $q_0 \in Q$ , une fonction de parité  $\Omega : Q \rightarrow \mathbb{N}$  et une fonction de transition

$$\delta : Q \times \Sigma_p \rightarrow \mathcal{P}_{reg}(Q^\infty)$$

où  $Q^\infty$  désigne l'ensemble des mots finis et infinis sur l'alphabet  $Q$ , et  $\mathcal{P}_{reg}(Q^\infty)$  l'ensemble des parties  $MS$ -définissables de  $Q^\infty$ , i.e. les langages rationnels sur l'alphabet  $Q$ .

**5.1.3.2 Définition (Calcul).** Etant donné  $L = \langle E, r, MSucc, \rho \rangle$ , un lexique on appelle calcul de  $A$  sur  $L$  une fonction  $C : E \rightarrow Q$  telle que :

1.  $C(r) = q_0$ ,
2. pour tout sommet  $e \in E$ ,  $C(MSucc(e)) \in \delta(C(e), \rho(e))$ .

Un tel calcul est dit acceptant lorsque, pour tout chemin infini,  $e_1.e_2.\dots$  sur le lexique  $L$ , l'indice minimum de tous les états  $q$  tel que  $q = C(e_i)$  pour un nombre infini d'indices  $i$  est pair. On dit alors que le lexique  $L$  est reconnu par  $A$  et on note  $L(A)$  l'ensemble des lexiques ainsi reconnus.

**Remarque.** On vérifie, la preuve étant similaire au cas classique des arbres binaires, que le langage  $L(A)$  est définissable en logique monadique du second ordre. L'existence d'un calcul acceptant revient en effet à "deviner" à l'aide de quantifications existentielles les ensembles  $\{C^{-1}(q)\}_{q \in Q}$  qui sont en nombre fini, puis à vérifier que la fonction  $C$  induite par ces ensembles est un calcul acceptant.

**Notation.** Dans la suite, pour tout état  $q \in Q$ , on note, comme dans le cas des automates d'arbres binaires,  $L_A(q)$  le langage reconnu par l'automate  $A[q/q_0]$  obtenu à partir de  $A$  en prenant comme état initial l'état  $q$  à la place de  $q_0$ .

**5.1.3.3 Théorème.** *Pour toute formule  $\varphi$  de la logique monadique du second ordre sur les lexiques il existe un  $L$ -automate  $A_\varphi$  tel que, pour tout lexique  $L$ ,  $L \models \varphi$  ssi  $L \in L(A_\varphi)$ .*

*Preuve.* Le reste de cette section est dédié à la preuve de ce théorème.

Soit donc  $\varphi$  une formule de la logique monadique du second ordre sur les lexiques. La proposition 5.1.2.4 nous assure qu'il existe une formule  $\psi$  sur les arbres de codage gauches dont les modèles sont exactement les codages des modèles de  $\varphi$ . Le théorème de Rabin nous permet alors d'affirmer qu'il existe un automate sur les arbres de codage

$$A = \langle Q, q_0, \delta, \Omega \rangle$$

dont le langage associé est exactement l'ensemble de ces codes. La suite de la preuve consiste à mettre, en plusieurs étapes, l'automate  $A$  sous une forme telle qu'on puisse simplement en extraire un  $L$ -automate.

**5.1.3.4 Lemme.** *Il existe un automate*

$$A' = \langle Q', q'_0, \delta', \Omega' \rangle$$

reconnaissant le même langage que  $A$  et tel que pour tout état  $q \in Q'$  l'une des conditions est satisfaite :

1.  $\delta'(q, *) = \emptyset$  on dit alors que  $q$  est un  $L$ -état,
2.  $\delta'(q, l) \neq \emptyset$  ssi  $l = *$  on dit alors que  $q$  est un  $*$ -état.

*Preuve.* Soit en effet  $Q_c$  une copie (disjointe) de  $Q$ ,  $h : Q_c \rightarrow Q$  la fonction faisant correspondre à chaque copie l'état dont il est une copie et  $g = h \cup Id_Q : Q_c \cup Q \rightarrow Q$  la projection de  $Q_c \cup Q$  sur  $Q$ .

On pose alors  $Q' = Q \cup Q_c$ , en prenant comme état initial  $q'_0 = q_0$ , en posant, pour tout état  $q \in Q'$ ,  $\Omega'(q) = \Omega(g(q))$  et, pour toute étiquette  $l \in \Sigma_*$  :

1. si  $q \in Q$ ,  $\delta'(q, *) = \emptyset$  et, pour tout  $l \neq *$ ,  $\delta'(q, l) = g^{-1}(\delta(q, l))$ ,
2. si  $q \in Q_c$ ,  $\delta'(q, *) = g^{-1}(\delta(q, *))$  et, pour tout  $l \neq *$ ,  $\delta'(q, l) = \emptyset$ .

Il est facile de voir que l'automate binaire ainsi construit reconnaît le même langage d'arbres binaires que l'automate  $A$ . □

Intuitivement les  $L$ -états vont constituer les états du  $L$ -automate que l'on cherche à extraire de  $A$ . Une transformation supplémentaire est nécessaire pour pouvoir tester les conditions infinitaires sur les seuls  $L$ -états.

**5.1.3.5 Définition ( $L$ -segment).** En reprenant la notion de forme arborescente d'un automate explicitée dans la partie 3.3.3, on appelle  $L$ -segment d'un automate sous forme arborescente

$$A = \langle Q_L \cup Q_*, q_0, \delta, \Omega \rangle$$

une séquence finie d'états  $q_1 \cdots q_n$  telle que :

1. pour tout  $i \in \{1, n\}$ ,  $q_i$  est un  $L$ -état ssi  $i = 1$  ou  $i = n$ .
2. pour tout  $i \in \{1, \dots, n-1\}$ ,  $q_i \ll_A q_{i+1}$ , où, là encore,  $\ll_A$  désigne la relation de couverture associée à la forme arborescente de l'automate  $A$ .

Il vient :

**5.1.3.6 Lemme.** *Il existe un automate  $A'$  sous forme normale arborescente, reconnaissant le même langage d'arbres que  $A$  tel que :*

(P) *pour tout  $L$ -segment  $q_1 \cdots q_n$  de  $A$  il existe au plus un  $i \in \{2, \dots, n\}$  tel que  $q_i$  soit un état retour.*

*Preuve.* Sans pertes de généralité, on peut supposer que  $A$  est sous forme arborescente. Pour obtenir un automate équivalent qui satisfait en outre la propriété il faut a priori déplier un peu plus l'automate  $A$ . Une façon de procéder est la suivante. On considère tout d'abord le développement universel  $\widehat{A}$  de l'automate  $A$ . Pour chaque branche infinie de  $\widehat{A}$ , pour chaque état retour  $q$  dans  $A$  dupliqué infiniment souvent sur cette branche, il nous faut choisir une copie  $q_c$  de  $q$  sur cette branche de telle sorte que toute transition au dessous de  $q_c$  et pointant sur  $q_c$  soit redirigée sur l'état  $q_c$ . La construction de l'automate désiré peut alors se faire en suivant les deux indications suivantes :

1. pour satisfaire la propriété  $(P)$ , il suffit, pour toute paire  $(q_i, q_j)$  d'états retours appartenant à un même  $L$ -segment  $q_1.q_2.\dots.q_n$  dans  $A$  de ne pas choisir leur copies sur une même copie  $q_1^c.q_2^c.\dots.q_n^c$  du  $L$ -segment considéré,
2. pour satisfaire en même temps la propriétés des formes arborescentes portant sur les indices de parité, il suffit de procéder pour ces choix dans l'ordre des indices croissants des états retour de  $A$ , quitte à se ramener initialement à un automate  $A$  dont les indices des états non retour sont toujours supérieurs à tout indices d'états retour.

□

On suppose donc que l'automate  $A$  est sous forme normale arborescente est qu'il satisfait la propriété  $(P)$  ci-dessus. Pour toute  $L$ -branche  $q_1.q_2.\dots.q_n$  il existe au plus état retour parmi  $\{q_2, \dots, q_n\}$ . De surcroît, cet état est uniquement déterminé par  $q_n$ . On le note donc, lorsqu'il existe,  $Retour(q_n)$ .

**5.1.3.7 Proposition.** *L'automate  $A'$  obtenu à partir de  $A$  en changeant la fonction de parité  $\Omega$  en  $\Omega'$  définie pour tout état  $q \in Q$  par :*

1. si  $Retour(q)$  est défini alors  $\Omega'(q) = \Omega(Retour(q))$ ,
2. sinon  $\Omega'(q) = \Omega(q)$ ,

reconnait  $L(A)$ .

*Preuve.* En effet, pour tout arbre binaire  $V$ , tout calcul  $C : T \rightarrow Q$ , pour toute branche de  $T$  sur laquelle l'état  $q$  apparaît infiniment souvent, l'état  $Retour(q)$  apparaît aussi infiniment souvent. Or l'automate  $A$  est sous forme arborescente. On a donc  $\Omega(q) \geq \Omega(Retour(q))$ . Ainsi, changer  $\Omega(q)$  en  $\Omega(Retour(q))$  ne modifie pas la validité ou la non-validité du critère d'acceptation infinitaire. □

On suppose donc cette transformation effectuée sur l'automate  $A$ . Il vient :

**5.1.3.8 Lemme.** *Pour tout arbre de codage  $V : T \rightarrow \Sigma_*$ , tout calcul  $C : T \rightarrow Q$  de l'automate  $A$  sur l'arbre  $V$ , toute branche infinie  $u_0, u_1, \dots, u_n, \dots$ , de l'arbre  $V$ , tel que  $* \neq V(u_i)$  pour une infinité de  $i$ ,*

$$\min\{\Omega(q) \mid q = C(u_i) \text{ pour une infinité de } i\} \text{ est pair,}$$

*ssi*

$$\min\{\Omega(q) \mid q = C(u_i) \text{ pour une infinité de } i \text{ avec } V(u_i) \neq *\} \text{ est pair.}$$

*Preuve.* Immédiate sachant que pour toute branche infinie portant infiniment souvent un  $L$ -état :

1. l'indice de parité minimum est nécessairement atteint en un état retour  $q$  (par définition des automates sous forme arborescente),
2. et il existe nécessairement un  $L$ -état  $q_1$  apparaissant infiniment souvent sur cette branche tel que  $q = \text{Retour}(q_1)$  et donc, par hypothèse,  $\Omega(q_1) = \Omega(q)$ ,

ce qui nous permet de conclure. □

Autrement dit, la condition d'acceptation portant sur les branches de calcul qui lisent une infinité de  $L$ -sommets ne porte que sur les  $L$ -états. Il nous reste alors à extraire de  $A$  les transitions appropriées.

**Notation.** Dans la suite, pour tout arbre de codage  $V$ , tout  $u \in T$ , on note  $V_u$  l'arbre binaire défini à partir de  $V$  en posant, pour tout  $v \in T$ ,

$$V_u(v) = V(u.v)$$

**5.1.3.9 Lemme.** *Pour tout  $L$ -état  $q$ , pour toute étiquette  $l \in \Sigma$  il existe un langage reconnaissable  $\delta_L(q, l) \subseteq Q_L^\infty$  tel que, pour tout arbre gauche  $V$  tel que  $V(\epsilon) = l$ , pour toute fonction  $C : \text{Succ}_V(E) \rightarrow Q$ , pour tout mot  $w \in Q^\infty$  on a équivalence de :*

1. *il existe un calcul acceptant de l'automate  $A$  à partir de l'état  $q$  sur l'arbre  $V$  tel que  $C(\text{MSucc}_V(\epsilon)) = w$ ,*
2.  *$w \in \delta_L(q, l)$  et pour tout  $u \in \text{Succ}(\epsilon)$ ,  $V_u \in L_A(C(u))$ .*

*Preuve.* Soit donc  $q$  un  $L$ -état de  $A$ ,  $l \in \Sigma$  une étiquette de  $L$ -sommet. On montre qu'il existe une formule  $\varphi_{q,l}$  de la logique monadique du second ordre sur les mots de  $Q^\infty$  tel que, pour tout arbre de codage  $V$  telle que  $V(\epsilon) = l$ , toute fonction  $C : \text{Succ}_V(\epsilon) \rightarrow Q$  telle que pour tout  $u \in \text{Succ}_V(\epsilon)$ ,  $V_u \in L_A(C(u))$ ,  $C(\text{MSucc}(\epsilon)) \models \varphi_{q,l}$  ssi la fonction  $C$  s'étend en un calcul acceptant de l'automate  $A$  à partir de l'état  $q$  sur l'arbre  $V$ .

Soit donc  $V$  un arbre de codage gauche. On remarque tout d'abord que pour tout sommet  $v$  tel qu'il existe  $u \in \text{Succ}_V(\epsilon)$  avec  $u \leq_{\text{pref}} v$ , l'extension de  $C$  à  $v$  ne pose aucune difficulté car  $V_u \in L_A(C(u))$ .

Dans la suite, on note  $\{1, 2, \dots\}$  l'ensemble des nœuds sous jacent aux graphes de  $w$  qui seront interprété dans la formule  $\varphi_{q,l}$  comme, selon le cas, les sommets de  $\text{Succ}_V(\epsilon)$ , ou bien leur parent immédiat dans  $\{d, g\}^*$  lorsqu'il est défini. Le mot  $w \in Q_L$  est alors noté  $q_{i_1}.q_{i_2} \dots$ , les états  $q_i$  étant deux à deux disjoints.

La formule  $\varphi_{q,l}$  est construite comme la disjonction des quatre cas suivant :

- (1).  $\epsilon \models \varphi_{q,l}$  ssi  $V_l \in L_A(q)$  où  $V_l$  désigne l'arbre binaire définie par  $V_l(\epsilon) = l$  et, pour tout  $u \neq \epsilon$ ,  $V_l(u) = *$ .
- (2).  $q_{i_1} \models \varphi_{q,l}$  ssi  $\exists q'_1 \in Q_*$   $(q_{i_1}, q'_1) \in \delta(q, l)$  avec  $V_* \in L_A(q'_1)$  où  $V_*$  est l'arbre défini par  $V_*(u) = *$  pour tout  $u$ .
- (3).  $q_{i_1}.q_{i_2} \models \varphi_{q,l}$  ssi  $(q_{i_1}, q_{i_2}) \in \delta(q, l)$ .
- (4).  $q_{i_1}.q_{i_2}.q_{i_3} \dots \models \varphi_{q,l}$  ssi, en notant  $m$  la taille du mot  $w$ , il une partition  $X_{q'_1}, \dots, X_{q'_n}$  de  $\{2, \dots, m-1\}$  avec  $q'_1, \dots, q'_n \in Q_*$  deux à deux distinct telle que, en notant pour tout  $k \in \{2, \dots, m-1\}$ ,  $j_k$  l'unique indice tel que  $k \in X_{q'_{j_k}}$  :

1.  $(q_{i_1}, q'_{j_1}) \in \delta(q, l)$ ,
2. pour tout  $k \in \{3, \dots, m-1\}$ ,  $(q_{i_{k-1}}, q'_{j_k}) \in \delta(q'_{j_{k-1}}, *)$ ,
3. si  $m$  est fini alors  $(q_{i_{m-1}}, q_{i_m}) \in \delta(q'_{j_{m-1}}, *)$  sinon, il existe  $k \in \{1, \dots, n\}$  tel que  $\Omega(q'_{j_k})$  est pair et pour tout  $k'$  tel que  $k \leq k'$ ,
  - (a)  $\Omega(q_{j_{k'}}) \geq k$ ,
  - (b) et il existe  $k'' \in X_{q_{j_k}}$  tel que  $k' \leq k''$ .

L'ensemble des états  $Q$  étant fini, les quantification sur les états de l'automate se traduisent bien comme des conjonctions ou des disjonctions finies.  $\square$

*Suite de la preuve du théorème 5.1.3.3.* On définit alors le  $L$ -automate  $A_\varphi$  par :

$$A_\varphi = \langle Q_L, q_0, \delta_L, \Omega \rangle$$

Etant donné un lexique  $L$ , supposons que  $L \models \varphi$ . Par définition de l'automate  $A$ ,  $CodG(L) \in L(A)$ . Il existe donc un calcul acceptant  $C : T \rightarrow Q$  de l'automate  $A$  sur l'arbre  $CodG(L)$ . Mais alors les lemmes 5.1.3.9 et 5.1.3.8 nous assurent que ce calcul restreint aux seuls  $L$ -sommets de  $CodG(L)$  définit bien un calcul acceptant de  $A_\varphi$  sur  $L$ .

Réciproquement, si  $C$  est un calcul acceptant de  $A_\varphi$  sur  $L$ , on peut l'étendre, par induction sur la profondeur des sommets de  $L$  considérés en appliquant à chaque étape le lemme 5.1.3.9, à un calcul  $C_1$  de l'automate  $A$  sur  $CodG(L)$  tel que :

1. pour tout sommet  $e$  de  $L$ , en notant  $c(e)$  ses coordonnées binaires gauches,

$$C(e) = C_1(c(e))$$

2. pour toute branche de  $V$  ne contenant qu'un nombre fini de  $L$ -sommets, la condition infinitaire est satisfaite.

Or le calcul  $C$  étant un calcul acceptant, pour toute branche de  $L$  les conditions infinitaires sont satisfaites. Le point 1 nous assure donc que les conditions infinitaires sur les branches où apparaissent une infinité de  $L$ -états sont, en se restreignant à ces  $L$ -états, satisfaites. Le lemme 5.1.3.8 nous assure alors qu'ajouter les  $*$ -états ne change pas ces résultats. On conclut ainsi que  $C_1$  est un calcul acceptant de l'automate  $A$  sur  $CodG(L)$  et donc, par définition de  $A$ , le lexique  $L$ , isomorphe à  $Lex(CodG(L))$ , est bien un modèle de  $\varphi$ .  $\square$

## 5.2 Propriétés induites

Dans cette section, on étudie les propriétés des arbres de transitions de cardinalité dénombrable définissables “modulo” leurs présentations lexicographiques.

### 5.2.1 Propriétés d'arbres “modulo” une présentation lexicographique

**Notation.** Pour tout arbre de transitions  $S$ , pour toute présentation lexicographique  $\leq_S$  de  $S$  (cf. définition 5.1.1.1), pour tout état  $e \in E_S$  on note  $MSucc(\leq_S, e)$  le mot fini ou infini sur l'alphabet  $E_S$  formée des successeurs de  $e$  ordonnés par  $\leq_S$ .

**5.2.1.1 Définition.** Une propriété d'arbres de transitions est dite *MS-définissable modulo une présentation lexicographique* lorsqu'elle est de la forme

$$\exists O \varphi(O) \wedge \psi(O)$$



où  $\varphi(O)$  spécifie, en logique monadique du second ordre, que la relation  $O$  est une présentation lexicographique de  $S$ , et  $\psi(O)$  est une formule de la logique monadique du second ordre sur le vocabulaire des arbres de transitions augmenté du symbole de relation binaire  $O$ .

**Notation.** L'ensemble des propriétés d'arbres  $MS$ -définissables modulo une présentation lexicographique est noté  $LMSO_{\leq l}$ .

**Remarque.** Lorsque  $O$  n'apparaît pas dans  $\psi(O)$ , comme tout arbre de transitions de cardinalité dénombrable admet au moins un ordonnancement local, une telle formule est équivalente, sur les arbres de cardinalité dénombrable, à une formule de logique monadique du second ordre. On a donc immédiatement l'inclusion sémantique :

$$LMSO \subseteq LMSO_{\leq l}$$

### 5.2.2 Extension du théorème de Parikh aux mots infinis

Dans cette partie, on note  $\mathbb{N}_\infty$  la clôture  $\mathbb{N} \cup \{\infty\}$  de l'ensemble des entiers naturels. On étend les opérations d'addition et de multiplication à  $\mathbb{N}_\infty$  en posant, pour tout  $x \in \mathbb{N}_\infty$ ,

$$x + \infty = \infty + x = \infty$$

et

$$x \cdot \infty = \infty \cdot x = \infty$$

En particulier, on pose  $0 \cdot \infty = \infty \cdot 0 = \infty$  !

**Notation.** Dans la suite, pour tout ensemble  $Q$ , on note  $\mathbb{N}_\infty^Q$  l'ensemble des fonctions de  $Q$  dans  $\mathbb{N}_\infty$ .

**5.2.2.1 Définition (Ensembles semi-linéaires).** Etant donné  $Q$  un ensemble fini, une partie  $P \subseteq \mathbb{N}_\infty^Q$  est dite *linéaire* lorsqu'il existe  $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_m \in \mathbb{N}_\infty^Q$  tels que

$$P = \{\alpha_0 + n_1\alpha_1 + \dots + n_m\alpha_m \mid (n_1, \dots, n_m) \in \mathbb{N}^m\}$$

Une partie  $P \subseteq \mathbb{N}_\infty^Q$  est dite *semi-linéaire* lorsqu'elle est une union finie de parties linéaires.

**5.2.2.2 Définition.** Etant donné  $Q$  un alphabet fini, étant donné  $L \subseteq Q^\infty$  un langage de mots finis et infinis sur  $Q$ , on appelle *image commutative* de  $L$  l'ensemble  $\Phi(L) \subseteq \mathbb{N}_\infty^Q$  défini par :

$$IC(L) = \{q \mapsto |w|_q \mid w \in L\}$$

où, pour tout état  $q \in Q$ ,  $|w|_q$  désigne le nombre (dans  $\mathbb{N}_\infty$ ) d'occurrences de la lettre  $q$  dans le mot  $w$ .

On dit alors que deux langages  $L_1$  et  $L_2 \subseteq Q^\infty$  sont *lettres-équivalents* lorsque

$$IC(L_1) = IC(L_2)$$

**5.2.2.3 Théorème (Parikh).** *Un langage  $L \subseteq Q^\infty$  admet une image commutative semi-linéaire ssi il est lettres-équivalent à un langage régulier (de façon équivalente à un langage  $MS$ -définissable).*

*Preuve.* Dans le cas d'un langage de mots finis, ce résultat, classique, est effectivement connu sous le nom de théorème de Parikh [21]. Son extension au cas infini ne pose pas de difficultés particulières. On peut par exemple, s'appuyer sur le fait qu'un langage reconnaissable est reconnaissable à l'aide d'un automate déterministe avec critère de Muller. De cet automate, on montre facilement que l'image commutative du langage considéré est semi-linéaire. La réciproque est immédiate.  $\square$

### 5.2.3 Caractérisation par automates

On caractérise dans cette section la logique  $LMSO_{\leq l}$  à l'aide d'une notion d'automate obtenue intuitivement en prenant les images commutatives des transitions des  $L$ -automates.

**5.2.3.1 Définition ( $ML$ -Automates).** Un  $ML$ -automate d'arbres est un  $n$ -uplet

$$A = \langle Q, q_0, \delta, \Omega \rangle$$

pour  $Q$  un ensemble fini d'états,  $q_0 \in Q$  un état initial,  $\Omega : Q \rightarrow \mathbb{N}$  une fonction de parité et une fonction de transition

$$\delta : Q \times \mathcal{P}(Prop) \times Act \rightarrow Psl(Q)$$

où  $Psl(Q)$  désigne l'ensemble des parties semi-linéaires de  $\mathbb{N}_\infty^Q$ .

**5.2.3.2 Définition.** Etant donné  $S$  un système de transition, un calcul de l'automate  $A$  sur  $S$  est une fonction  $C : E_S \rightarrow Q$  telle que :

1.  $C(r_S) = q_0$ ,
2. pour tout sommet  $e \in E$ , toute action  $a \in Act$ ,  $M(C, e, a) \in \delta(C(e), \rho_S(e), a)$  pour  $M(C, e, a) \in \mathbb{N}_\infty^Q$  défini pour tout état  $q \in Q$  par :

$$M(C, e, a)(q) = |\{f \in E_S \mid (e, f) \in R_S(a) \wedge C(f) = q\}|$$

Un tel calcul est dit acceptant lorsque, pour tout chemin infini  $r_S.e_1.e_2.\dots$  de  $S$ , le minimum des indices  $\Omega(q)$  tels que  $q = C(e_i)$  pour une infinité de  $i$  est pair.

Il vient :

**5.2.3.3 Théorème.** *Pour toute formule  $\varphi \in \text{LMSO}_{\leq l}$  il existe un ML-automate  $A$  tel que, pour tout arbre de transitions  $S$ ,  $S \models \varphi$  ssi  $S \in L(A)$ .*

*Preuve.* Soit  $\varphi \in \text{LMSO}_{\leq l}$ . Par définition, la formule  $\varphi$  est de la forme

$$\exists O \varphi(O) \wedge \psi(O)$$

où  $\varphi'(O)$  spécifie, en logique monadique du second ordre, que la relation  $O$  est une présentation lexicographique d'arbre de transitions, et  $\psi(O)$  est une formule de la logique monadique sur le vocabulaire des systèmes de transitions augmenté du symbole de relation binaire  $O$ . Par application de la proposition 5.1.1.3, il existe une formule  $\psi_1$  sur les lexiques telle que, pour tout arbre de transitions  $S$ , toute présentation lexicographique  $O$  de  $S$ ,

$$\langle S, O \rangle \models \psi(O) \text{ ssi } L(S, O) \models \psi_1$$

où  $L(S, O)$  désigne le lexique induit par  $S$  et  $O$ . Soit alors  $A_1 = \langle Q, q_0, \delta_1, \Omega \rangle$  le  $L$ -automate (sur les lexiques) caractérisant au sens du théorème 5.1.3.3 la sémantique de  $\psi_1$ .

**5.2.3.4 Proposition.** *On peut supposer, tout en conservant l'équivalence précédente, que pour tout état  $q \in Q$  il existe au plus une action  $a_q \in \text{Act}$  telle que  $\delta_1(q, (a_q, P)) \neq \emptyset$  pour au moins un  $P \subseteq \text{Prop}$ .*

*Preuve.* Si ce n'est pas le cas, il suffit en effet de construire l'automate

$$A'_1 = \langle Q', q'_0, \delta'_1, \Omega' \rangle$$

avec  $Q' = Q \times \text{Act}$ ,  $q'_0 = (q_0, a_0)$ , pour toute paire  $(q, a) \in Q'$ , pour toute étiquette  $(b, P) \in \Sigma = \text{Act} \times \mathcal{P}(\text{Prop})$ ,

1. si  $a = b$  on pose  $\delta'_1((q, a), (b, P)) = \{(q_0, a_0).(q_1, a_1).\dots \in Q'^{\infty} \mid q_0.q_1.\dots \in \delta_1(q, (a, P))\}$ ,
2. sinon on pose  $\delta'_1((q, a), (b, P)) = \emptyset$ ,

et pour toute paire  $(q, a) \in Q'$ ,  $\Omega'((q, a)) = \Omega(q)$ .

On vérifie que pour tout état  $q \in Q$ , pour toute partie  $P$  de  $\text{Prop}$ , le langage

$$\delta'_1((q, a), (a, P)) \subseteq (Q \times \text{Act})^{\infty}$$

est bien reconnaissable car  $\delta_1(q, (a, P)) \subseteq Q^\infty$  est reconnaissable : l'automate  $A'_1$  ainsi construit est donc bien un  $L$ -automate. De plus, pour tout lexique  $L$  dont la racine porte une étiquette de la forme  $(a_0, P)$ , on vérifie aussi que  $L \in L(A_1)$  ssi  $L \in L(A'_1)$ .  $\square$

**Notation.** En supposant donc que le  $L$ -automate  $A_1$  satisfait la propriété précédente, on note, pour toute action  $a \in Act$ ,

$$Q_a = \{q \in Q \mid a_q = a\}$$

On construit alors le  $ML$ -automate souhaité à partir du  $L$ -automate  $A_1$  en posant :

$$A = \langle Q, q_0, \delta, \Omega \rangle$$

avec, pour tout état  $q \in Q$ , toute partie  $P \subseteq Prop$ , toute action  $a \in Act$ ,

$$\delta(q, P, a) = \{f \in \mathbb{N}_\infty^Q \mid \exists w \in \delta_1(q, (a_q, P)), \forall q_1 \in Q, f(q_1) \neq 0 \Rightarrow f(q_1) = |w|_{q_1} \wedge q_1 \in Q_a\}$$

Cet ensemble est bien une partie semi-linéaire de  $\mathbb{N}_\infty^Q$  puisque c'est l'image commutative de la projection sur l'alphabet  $Q_a$  du langage  $\delta_1(q, (a_q, P))$  par définition reconnaissable.

Il est alors facile de voir que pour tout arbre de transitions  $S$ ,  $S \in L(A)$  ssi il existe une présentation lexicographique  $O$  de  $S$  tel que  $L(S, O) \in L(A_1)$ .  $\square$

## 5.3 Applications

Dans cette section, on démontre que toute formule de la logique monadique sur les systèmes de transitions dont la sémantique est close par bisimulation est équivalente à une formule du  $\mu$ -calcul modal.

### 5.3.1 $\Omega$ -clôture d'une formule

**5.3.1.1 Théorème.** *Pour toute  $MS$ -formule  $\varphi$  il existe une formule du  $\mu$ -calcul  $\tilde{\varphi}$ , appelée  $\Omega$ -clôture de  $\varphi$ , telle que, pour tout système de transitions de cardinalité au plus dénombrable, en notant  $\tilde{S}$  l'expansion universelle de  $S$ ,  $\tilde{S} \models \varphi$  ssi  $S \models \tilde{\varphi}$ .*

*Preuve.* Soit  $A_1 = \langle Q, q_0, \delta_1, \Omega \rangle$  un  $ML$ -automate caractérisant la sémantique de la formule  $\varphi$  au sens du théorème 5.2.3.3.

En fait, on démontre le lemme suivant :

**5.3.1.2 Lemme.** *Pour tout ML-automate  $A_1$  il existe un automate modal alternant  $A$  tel que, pour tout système de transitions de cardinalité au plus dénombrable,  $\tilde{S} \in L(A_1)$  ssi  $S \in L(A)$ .*

*Preuve.* Soit en effet l'automate  $A$  défini par :

$$A = \langle Q, q_0, \delta, \Omega \rangle$$

pour  $\delta : Q \times \mathcal{P}(Prop) \rightarrow \mathcal{P}(Act \rightarrow \mathcal{P}(\mathcal{P}(Q)))$  la fonction de transition modale alternante étant décrite ci-dessous.

Etant donnés  $q \in Q$  et  $P \subseteq Prop$ , la définition des ensembles semi-linéaires et la finitude de  $Q$  nous permettent d'affirmer qu'il existe un entier  $n$  tel que, pour toute action  $a \in Act$  :

$$\delta_1(q, P, a) = T_{a,1} \cup T_{a,2} \cup \dots \cup T_{a,n}$$

où, pour tout  $i \in \{1, \dots, n\}$ ,  $T_{a,i}$  est une partie *linéaire* de  $\mathbb{N}_\infty^Q$ .

Notons au passage que pour toute partie linéaire  $T \subseteq \mathbb{N}_\infty^Q$ , pour tout état  $q \in Q$  on a équivalence de :

1. il existe  $f \in T$  tel que  $f(q) = \infty$ ,
2. pour tout  $f \in T$ ,  $f(q) = \infty$ .

**5.3.1.3 Définition (Etats infinitaires).** Un état satisfaisant l'une des deux conditions précédentes est alors appelé *état infintaire* dans  $T$ . L'ensemble des états infinitaires de  $T$  est alors noté  $Q_\infty(T)$ .

De surcroît, pour toute action  $a \in Act$ , quitte à décomposer  $\delta_1(q, P, a)$  plus finement, on peut toujours supposer que pour tout  $i \in \{1, \dots, n\}$ , pour tout état  $q \in Q$  on a équivalence de :

1. il existe  $f \in T_{a,i}$  tel que  $f(q) = 0$ ,
2. pour tout  $f \in T_{a,i}$ ,  $f(q) = 0$ .

**5.3.1.4 Définition (Etats nuls).** Un état satisfaisant l'une de ces deux conditions sera alors appelé *état nul* dans  $T$ . L'ensemble des états *non nuls* de  $T_{a,i}$  est alors noté  $Q(T_{a,i})$ .

**5.3.1.5 Définition (Transitions modales alternantes induites).** On définit l'ensemble de "transitions"  $\delta(q, P)$  comme l'ensemble des fonctions  $f : Act \rightarrow \mathcal{P}(\mathcal{P}(Q))$  telles que pour toute action  $a \in Act$ , il existe  $i \in \{1, \dots, n\}$  satisfaisant :

1.  $Q(T_{a,i}) = \bigcup \{Q' \in f(a)\}$ ,

2. pour tout ensemble  $Q' \in f(a)$ ,  $Q' \cap Q_\infty(T_{a,i}) \neq \emptyset$ , i.e. tout ensemble d'états dans  $f(a)$  contient au moins un état infinitaire.

Il nous reste alors à montrer l'équivalence de  $A_1$  et  $A$  sur les expansions de systèmes de transitions. Soit donc  $S$  un système de transitions (de cardinalité au plus dénombrable) et  $\tilde{S}$  son expansion.

On montre tout d'abord :

**5.3.1.6 Proposition.** *Si  $\tilde{S} \in L(A_1)$  alors  $\tilde{S} \in L(A)$ .*

*Preuve.* Il existe un calcul acceptant  $C : E_{\tilde{S}} \rightarrow Q$  de l'automate  $A_1$  sur  $\tilde{S}$ .

Soit alors la relation de *similitude* sur les sommets de  $\tilde{S}$  définie comme la plus petite relation d'équivalence, noté  $\simeq_c$ , telle que, pour tous sommets  $e$  et  $f \in E_S$  tel que  $e \simeq_c f$ , pour toute action  $a \in \Sigma$ , pour tout sommet  $e' \in \text{Succ}_a(e)$ , tout sommet  $f' \in \text{Succ}_a(f)$ , si les sous-arbres induit dans  $\tilde{S}$  par les sommets  $e'$  et  $f'$  sont isomorphes alors  $e' \simeq_c f'$ .

On montre alors que l'arbre induit par l'ensemble des paires

$$\{(e, C(f)) \in E_{\tilde{S}} \times Q \mid e \simeq_c f\}$$

i.e. en considérant qu'une paire  $(e', C(f'))$  est fils d'une paire  $(e, C(f))$  ssi  $e' \in \text{Succ}(e)$ , est bien un calcul acceptant de l'automate modal  $A$  sur  $\tilde{S}$ .

Notons tout d'abord que le fait que  $C$  soit un calcul acceptant de  $A_1$  sur  $\tilde{S}$  nous assure que les conditions infinitaires sont effectivement satisfaites. Il reste donc seulement à montrer que  $C$  est bien un calcul d'automate modal alternant.

Comme  $C(r_{\tilde{S}}) = q_0$  la racine de notre arbre de calcul est bien étiquetée  $(r_{\tilde{S}}, q_0)$ .

Soit un nœud de cet arbre étiqueté par  $(e, C(f))$ . Les sous-arbres de  $\tilde{S}$  induit par les sommets  $e$  et  $f$ , il suffit de considérer le cas où  $f = e$ . Considérons alors  $a \in \text{Act}$ . L'équivalence  $\simeq_c$  induit une partition  $E_1, E_2, \dots, E_p$  de l'ensemble  $\text{Succ}_a(e)$ , chaque ensemble  $E_i$  étant, par construction des  $\omega$ -expansions, infinis.

Soit alors la fonction  $f : \text{Act} \rightarrow \mathcal{P}(\mathcal{P}(Q))$  définie pour toute action  $a \in \text{Act}$  par :

$$f(a) = \{C(E_i) \mid 1 \leq i \leq qp\}$$

Montrons qu'elle appartient bien à  $\delta(C(e), \rho_S(e))$ .

En effet,  $C$  est un calcul de  $ML$ -automate. Pour toute action  $a \in \text{Act}$ , en notant, comme dans la définition 5.2.3.2,  $M(C, e, a) \in \mathbb{N}_\infty^Q$  le multi-ensemble de  $Q$  induit par  $C$  sur  $\text{Succ}_a(e)$ , on a donc bien :

$$M(C, e, a) \in \delta_1(C(e), \rho_S(e), a)$$

C'est à dire, en reprenant les notations utilisées dans la définition de  $\delta$ , qu'il existe un entier  $k \in \{1, \dots, n\}$  tel que  $M(C, e, a) \in T_{a,k}$ . Or :

1. pour tout état  $q' \in Q$  non nul dans  $T_{a,k}$ , comme  $M(C, e, a) \in T_{a,k}$ , il existe bien un sommet  $e \in \text{Succ}_a(e) \cap E_i$  tel que  $q' \in C(E_i)$ , tout état de  $C(\text{Succ}_a(e))$  étant non nul dans  $T_{a,k}$ ,
2. pour tout  $i \in \{1, \dots, p\}$ , l'ensemble  $E_i$  étant infini, il existe nécessairement un état  $q' \in C(E_i)$  infinitaire dans  $T_{a,k}$ , i.e. tel que  $q' = C(e)$  pour une infinité de  $e \in E_i$ .

□

De la même façon :

**5.3.1.7 Proposition.** *Si  $\tilde{S} \in L(A)$  alors  $\tilde{S} \in L(A_1)$ .*

*Preuve.* Soit un calcul acceptant de l'automate modal  $A$  sur  $\tilde{S}$ , c'est-à-dire un arbre  $T$  étiqueté sur  $E_S \times Q$  satisfaisant les conditions de la définition 3.3.1.2.

On considère le calcul de l'automate  $A$  obtenue à partir de du calcul précédent par clôture vis à vis de l'équivalence de similitude définit dans la preuve précédent, i.e. le calcul obtenue en considérant l'arbre induit (comme précédemment) par les paires  $(e, q) \in E_S \times Q$  telle qu'au moins un sommet de  $T$  soit étiqueté par une paire de la forme  $(f, q)$  avec  $e \simeq_c f$ . Les sous-arbres de  $\tilde{S}$  induit par ces deux sommets  $e$  et  $f$  étant isomorphes, une telle clôture est bien un calcul acceptant de l'automate modal  $A$  sur  $\tilde{S}$ .

En notant toujours  $T$  le nouvel arbre de calcul ainsi obtenue, on construit alors, par induction sur la structure arborescente de l'ensemble des sommets de  $T$  un calcul acceptant  $C : E_S \rightarrow Q$  de l'automate  $A_1$  sur  $\tilde{S}$ .

Initialement, on pose  $C(r_S) = q_0$ . La racine de  $T$  est bien étiquetée par  $(r_S, q_0)$ . Supposons construit notre calcul jusqu'à un sommet simple  $e \in E_S$  tel qu'il existe un nœud  $n_e$  de  $T$  étiqueté par  $(e, C(e))$ , et soit  $q = C(e)$ .

Par définition d'un calcul d'automate modal alternant, il existe une fonction  $g \in \delta(C(e), \rho_S(e))$  telle que pour toute action  $a \in Act$  :

1. pour tout sommet  $f \in \text{Succ}_a(e)$  il existe  $Q' \in g(a)$  tel que toute étiquette de la forme  $(f, q')$  pour  $q' \in Q'$  apparaît sur au moins un fils de  $n_e$ ,
2. pour tout ensemble  $Q' \in g(a)$  il existe  $f \in \text{Succ}_a(e)$  tel que, là encore, toute étiquette de la forme  $(f, q')$  pour  $q' \in Q'$  apparaît sur au moins un fils de  $n_e$ ,

Etant fixé  $a \in Act$ , on pose à nouveau, de la même façon que dans la preuve précédente,

$$\text{Succ}_a(e) = E_1 \cup E_2 \cup \dots \cup E_p$$

les  $E_1, E_2, \dots, E_n$  étant les classes d'équivalence, infinie, induite par l'équivalence de similitude  $\simeq_c$  sur  $\text{Succ}_a(e)$ .

Par construction de la fonction de transition  $\delta$  il existe une partie linéaire  $T \subseteq \delta_1(q, \rho_S(e), a)$  telle que :

1.  $Q(T) = \bigcup \{Q' \in g(a)\}$ ,
2. pour tout ensemble  $Q' \in g(a)$ ,  $Q' \cap Q_\infty(T) \neq \emptyset$ .

Soit donc une fonction  $h : Q \rightarrow \mathbb{N}_\infty$  choisie *arbitrairement* dans  $T$ . Par construction, pour tout état  $q' \in Q$ ,

1.  $q' \in Q_\infty(T)$  ssi  $h(q') = \infty$ ,
2.  $q' \in Q(T)$  ssi  $h(q') \neq 0$ .

La construction de  $C$  procède alors de la façon suivante.

1. Pour tout  $Q' \in g(a)$ , tout  $q' \in Q'$ , il existe au moins un indice  $i \in \{1, \dots, p\}$  tels que, pour tout sommet  $e' \in E_i$ , toutes les paires de la forme  $(e', q')$  étiquettent des fils de  $n_e$ . Chaque ensemble  $E_i$  étant infini, il existe une famille de parties disjointes de  $\text{Succ}_a(e)$

$$\{U_{q'}\}_{q' \in \bigcup \{Q' \in g(a)\}}$$

telle que, pour tout  $Q' \in g(a)$ , pour tout état  $q' \in Q'$ ,

$$h(q') = |U_{q'}|$$

et, pour tout sommet  $f \in U_{q'}$  l'étiquette  $(f, q')$  apparaît sur au moins un sommet  $n_f$  fils de  $n_e$ . On pose donc, pour tout sommet  $f \in U_{q'}$ ,  $C(f) = q'$ .

2. pour tout sommet  $f \in \text{Succ}_a(e)$  n'appartenant pas aux ensembles  $U_{q'}$  il existe tout de même  $Q' \in g(a)$  et un état  $q' \in Q'$  infinitaire tel qu'au moins un fils  $n_f$  de  $n_e$  porte l'étiquette  $(f, q')$ . On pose alors  $C(f) = q'$  en ajoutant  $f$  à l'ensemble  $U_{q'}$ .

Autrement dit, on a étendu la fonction  $C$  à  $\text{Succ}_a(e)$  de telle sorte que pour tout état  $q \in Q$ ,

$$|C^{-1}(q) \cap \text{Succ}_a(e)| = h(q)$$



en disposant, pour tout sommet  $f \in \text{Succ}_a(e)$  d'un sommet  $n_f$  de  $T$  étiqueté par  $(f, C(f))$ , ce qui clos l'induction.

La fonction  $C$  ainsi défini est bien un calcul de l'automate  $A_1$  sur  $\tilde{S}$ . Comme chaque branche infinie de ce calcul coïncide avec au moins une branche infinie du calcul acceptant de  $A$  sur  $\tilde{S}$ , les conditions infinitaires sont aussi satisfaites.  $\square$

*Fin de la preuve du théorème 5.3.1.1.* Le théorème 3.3.4.1 affirmant qu'il existe une formule du  $\mu$ -calcul modal dont les modèles sont exactement les systèmes reconnus par l'automate  $A$  nous permet de conclure.  $\square$

### 5.3.2 Caractérisation du pouvoir expressif du $\mu$ -calcul modal

On explicite ici l'un des corollaires majeurs du théorème précédent.

**5.3.2.1 Définition.** Une  $MS$ -formule  $\varphi$  est dite *close par bisimulation* lorsque, pour tous systèmes de transitions (de cardinalité dénombrable)  $S_1$  et  $S_2$ , si  $S_1$  et  $S_2$  sont en bisimulation alors  $S_1 \models \varphi$  ssi  $S_2 \models \varphi$ .

**Remarque.** La définition de la notion d'expansion nous permet de donner une formulation équivalente de la définition précédente. On a en effet :  $\varphi$  est close par bisimulation ssi, pour tout système de transitions  $S$  de cardinalité au plus dénombrable,  $S \models \varphi$  ssi  $\tilde{S} \models \varphi$ .

Il vient :

**5.3.2.2 Corollaire.** Une formule  $\varphi$  de la logique monadique du second ordre est close par bisimulation ssi elle est équivalente à une formule du  $\mu$ -calcul modal.

*Preuve.* Il est bien connu que les formules du  $\mu$ -calcul modal sont closes par bisimulation. La réciproque quant à elle est une conséquence immédiate du théorème 5.3.1.1.  $\square$

Autrement dit, le  $\mu$ -calcul modal est le plus grand ensemble de formules "préservant" l'équivalence de bisimulation dont la sémantique est définissable en logique monadique du second ordre.

**Remarque.** Si une  $MS$ -formule  $\varphi$  est close par bisimulation alors elle est nécessairement équivalente (sur les systèmes dénombrables) à sa  $\Omega$ -clôture  $\tilde{\varphi}$ . Autrement dit,  $\varphi$  est close par bisimulation ssi la formule  $\varphi \Leftrightarrow \tilde{\varphi}$  est valide. La validité d'une telle équivalence étant décidable dans le cas restreint des arbres de transitions, savoir si une formule de LMSO admet, sur les arbres, un ensemble de modèle clos par équivalence de bisimulation est donc décidable.



# Références

- [1] Abramsky (S.). – Domain theory in logical form. *In: 2<sup>nd</sup> IEEE Symposium on Logic In Computer Science.* – (Version longue dans *Annals of pure and applied logic*, 51 (1991)).
- [2] Abramsky (S.). – A domain equation for bisimulation. *Information and Computation*, vol. 92, n° 2, 1991, pp. 161–218.
- [3] Arnold (A.). – *Systèmes de transitions finis et sémantique des processus communicants.* – Masson, 1992, *Etudes et Recherches en Informatiques.*
- [4] Arnold (A.). – Verification and comparison of transition systems. *In: Theor. And Pract. Of Soft. Dev.*, éd. par Gaudel (M.-C.) et Jouannaud (J. P.). – LNCS 668.
- [5] Arnold (A.). – An initial semantics for the  $\mu$ -calculus on trees and rabin’s complementation lemma. *Theoretical Computer Science*, vol. 148, 1995, pp. 121–132.
- [6] Arnold (A.) et Dicky (A.). – An algebraic characterization of transition system equivalences. *Information and Computation*, vol. 82, 1989, pp. 198–229.
- [7] Benthem (J. van). – *Languages in Action. Catégories, Lambdas and Dynamic Logic.* – North-Holland, 1991, *Studies in Logic*, volume 130.
- [8] Bernholtz (O.) et Grumberg (O.). – Branching time temporal logic and amorphous tree automata. *In: 4th Conf. on Concurrency Theory.* pp. 262–277. – LNCS 715.
- [9] Castellani (I.). – Bisimulations and abstraction homomorphisms. *Journal of Computer and System Sciences*, vol. 34, 1987, pp. 210–235.
- [10] Courcelle (B.). – The monadic second order logic of graph I: Recognizable sets of finite graphs. *Inf. and Comp.*, vol. 85, 1990, pp. 12–75.
- [11] Courcelle (B.). – Monadic second order definable graph transductions: A survey. *Theoretical Computer Science*, vol. 126, 1994, pp. 53–75.

- [12] Courcelle (B.). – *The Monadic Second-Order Of Graph X: Linear Orderings*. – Rapport technique, LaBRI, Université de Bordeaux I, 1994. A paraître dans TCS.
- [13] Courcelle (B.). – The monadic second-order logic of graphs IX: Machines and their behaviours. *Theoretical Computer Science*, vol. 149, October 1995.
- [14] Courcelle (B.) et Walukiewicz (I.). – Monadic second-order logic, graph coverings and unfoldings of transition systems. 1995. – Présenté à CSL'95.
- [15] Ebbinghaus (H.D.), Flum (J.) et Thomas (W.). – *Mathematical Logic*. – Springer-Verlag, 1984, *Undergraduate Texts in Mathematics*.
- [16] Ehrenfeucht (A.). – An application of games to the completeness problem for formalized theories. *Fund. Math.*, vol. 49, 1961.
- [17] Emerson (E. A.) et Jutla (C. S.). – Tree automata, mu-calculus and determinacy. *In: 6<sup>th</sup> IEEE Symp. on Logic in Comp. Science*.
- [18] Fraïssé (R.). – Sur quelques classifications des systèmes de relations. *Publications Scientifiques de l'Université d'Alger*, vol. A, n° 1, 1954.
- [19] Gunter (C.A.) et Scott (D.S.). – Semantic domains. *In: Handbook of Theor. Comp. Science (vol. B)*, éd. par Van Leeuwen (J.), pp. 633–674. – Elsevier, 1990.
- [20] Gurevich (Y.) et Harrington (L.). – Trees, automata and games. *In: 14<sup>th</sup> ACM Symp. on Theoer. of Comput.*, pp. 60–65.
- [21] Harrisson (M. A.). – *Introduction to Formal Language Theory*. – Addison-Wesley, 1978, *Series In Computer Science*.
- [22] Hennessy (M.) et Milner (R.). – Observing nondeterminism and concurrency. *In: 7<sup>th</sup> Int. Coll. on Aut., Lang. and Prog.*, éd. par de Backer (J.) et van Leeuwen (M.). – LNCS 85.
- [23] Hennessy (M.) et Milner (R.). – Algebraic laws for nondeterminism and concurrency. *J. Assoc. Comput. Mach.*, vol. 32, 1985, pp. 137–161.
- [24] Hodges (W.). – *Model theory*. – Cambridge University Press, 1993, *Encyclopedia of mathematics and its application*, volume 42.
- [25] Janin (D.). – Some results about logical descriptions of non deterministic behaviours. *In: XIII<sup>th</sup> Found. of Soft. Tech. and Theo. Comp. Sci.* – LNCS 761.

- [26] Janin (D.). – *Propriétés logiques du non déterminisme et  $\mu$ -calcul modal*. – Thèse de Doctorat, Ecole Doctorale de Mathématiques et Informatiques, Université de Bordeaux I, 1996.
- [27] Janin (D.) et Walukiewicz (I.). – Automata for the modal  $\mu$ -calculus and related results. *In: Math. Found. of Comp. Sci. 95*. – LNCS 969.
- [28] Janin (D.) et Walukiewicz (I.). – On the expressive completeness of the modal  $\mu$ -calculus w.r.t. monadic second order logic. *In: 11<sup>th</sup> IEEE Symp. on Logic in Comp. Science*. – Soumis.
- [29] Johnstone (P. T.). – *Stone spaces*. – Cambridge University Press, 1982, *Cambridge studies in advanced mathematics*, volume 3.
- [30] Kozen (D.). – Results on the propositional  $\mu$ -calculus. *Theoretical Computer Science*, vol. 27, 1983, pp. 333–354.
- [31] Mostowski (A. W.). – Regular expressions for infinite trees and a standard form of automata. *In: Computation Theory*. – LNCS 208.
- [32] Mostowski (A.W.). – Hierarchies of weak automata on weak monadic formulas. *Theoretical Computer Science*, vol. 83, 1991.
- [33] Muchnik (A. A.). – Games on infinite trees and automata with dead-ends: a new proof for the decidability of the monadic second order theory of two successors. *Bull. EATCS*, vol. 42, 1992, pp. 220–267. – (traduction d'un article en Russe de 1984).
- [34] Muller (D. E.) et Schupp (P. E.). – Alternating automata on infinite trees. *Theoretical Computer Science*, vol. 54, 1987, pp. 267–276.
- [35] Muller (D. E.) et Schupp (P. E.). – Simultating alternating tree automata by non-deterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, vol. 141, 1995, pp. 67–107.
- [36] Niwiński (D.). – Fixed point vs. infinite generation. *In: 3<sup>rd</sup> IEEE Symp. on Logic in Comp. Science*.
- [37] Palmgren (E.) et Stoltenberg-Hansen (V.). – Logically presented domains. *In: 10<sup>th</sup> IEEE Symp. on Logic in Comp. Science*.
- [38] Park (D.). – Concurrency and automata on infinite sequences. *In: 5th GI Conf. on Theoret. Comput. Sci.* pp. 167–183. – Karlsruhe, 1981.

- [39] Plotkin (G. D.). – A powerdomain construction. *SIAM Journal on Computing*, vol. 5, 1976.
- [40] Rabin (M. O.). – Decidability of second order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, vol. 141, 1969.
- [41] Robinson (E.). – Logical aspects of denotational semantics. In: *Category Theory and Computer Science*. – LNCS 283.
- [42] Smyth (M. B.). – Power domains and predicate transformers: a topological view. In: *ICALP'83*. – LNCS 154.
- [43] Streett (R. S.) et Emerson (E. A.). – An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, vol. 81, 1989.
- [44] Thomas (W.). – Automata on infinite object. In: *Handbook of Theor. Comp. Science (vol. B)*, éd. par Van Leeuwen (J.), pp. 133–191. – Elsevier, 1990.
- [45] Thomas (W.). – On the Ehrenfeucht-Fraïssé game in theoretical computer science. In: *Theor. And Pract. Of Soft. Dev.*, éd. par Gaudel (M.-C.) et Jouannaud (J. P.). – LNCS 668.
- [46] Walukiewicz (I.). – On the ambiguity problem, 1994. Notes non publiées.
- [47] Walukiewicz (I.). – Completeness of kozen's axiomatization of the propositional  $\mu$ -calculus. In: *10th IEEE Symp. on Logic in Comput. Sci.*
- [48] Walukiewicz (I.). – Monadic second order logic on tree-like structures. In: *Symp. on Theor. Aspect of Comp. Science*. – A paraître.
- [49] Winskel (G.). – A note on powerdomains and modalities. In: *Int. FCT-Conf*. – LNCS 158.

# Annexe A

## Structures, formules et jeux logiques

On rappelle ici quelques notions générales de théorie des modèles nous permettant en particulier de fixer les notations et la terminologie utilisées ici. Cette présentation, bien évidemment succincte, ne pourrait se substituer aux présentations classiques telles que celles qu'on peut trouver dans les premiers chapitres du *Mathematical Logic* de Ebbinghaus, Flum et Thomas [15] ou bien le *Model Theory* de Hodges [24].

### A.1 Structures, vocabulaires et formules logique

On appelle *vocabulaire* ou *signature* une liste  $L$  de symboles de constantes et de symboles de relations. Une  $L$ -structure  $S$  est alors tout  $n$ -uplet composé de :

1. un ensemble de points  $E_S$  appelé *domaine* ou bien *univers* de la structure  $S$ ,
2. pour tout symbole constant  $c \in L$ , un élément  $c_S \in E_S$  appelé *interprétation* de la constante  $c$  dans la structure  $S$ ,
3. pour tout symbole relationnel  $R \in L$  d'arité  $n$ , une partie  $R_S \subseteq E_S^n$  appelée *interprétation* de la relation  $R$  dans la structure  $S$ .

Pour une  $L$ -structure  $S$ , on notera souvent  $\rho_S$  la fonction associant à tout élément  $e \in E_S$  l'ensemble des symboles de prédicats  $p \in L$  tels que  $e \in p_S$ . Lorsque  $p \in \rho_S(e)$  on dit que la propriété  $p$  est satisfaite par le point  $e$ .

Etant donné  $\{x, y, z, \dots\}$  un ensemble de variables appelées variables du premier ordre, étant donné  $\{X, Y, Z, \dots\}$  un ensemble de variables appelées variables ensemblistes, une

formule de la logique monadique du second ordre, appelée aussi *MS*-formule, est une expression  $\varphi$  de la forme :

$$\varphi = R(t_1, t_2, \dots, t_k) | t \in X | \neg \varphi_1 | \varphi_1 \wedge \varphi_2 | \forall x \varphi_1 | \forall X \varphi_1$$

pour :

1. tout symbole de relation  $R$  d'arité  $k$ , les expressions  $t, t_1, t_2, \dots, t_k$  dénotant des variables du premier ordre ou constantes de  $L$ ,
2. toute variable du premier ordre  $x$ , toute variable ensembliste  $X$ ,
3. toute *MS*-formule  $\varphi_1$  et  $\varphi_2$ .

L'ensemble des *MS*-formules est noté *LMSO*.

Dans une formule  $\varphi$ , une occurrence de la variable  $x$  (resp.  $X$ ) est dite *liée* lorsqu'elle apparaît dans une sous-formule de la forme  $\forall x \psi$  (resp.  $\forall X \psi$ ). Dans le cas contraire, elle est dite *libre*.

Pour simplifier l'écriture d'une formule, on introduit les constantes et constructeurs supplémentaires suivants dont on donne l'interprétation d'un point de vue sémantique :

1. la disjonction :  $\varphi_1 \vee \varphi_2$  interprétée par la formule  $\neg((\neg \varphi_1) \wedge (\neg \varphi_2))$ ,
2. les quantifications existentielles :  $\exists x \varphi$  (resp.  $\exists X \varphi$ ) interprétée par la formule  $\neg(\forall x \neg \varphi)$  (resp.  $\neg(\forall X \neg \varphi)$ ),
3. l'implication :  $\varphi_1 \Rightarrow \varphi_2$  interprétée par la formule  $\neg \varphi_1 \vee \varphi_2$ ,
4. l'équivalence :  $\varphi_1 \Leftrightarrow \varphi_2$  interprétée comme la formule  $(\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)$ .

## Modèles d'une formule

Étant donné une *MS*-formule  $\varphi$ , étant donné

$$\mathbf{X} = (X_1, \dots, X_m) \text{ (resp. } \mathbf{x} = (x_1, x_2, \dots, x_n))$$

un vecteur de variables de prédicats (resp. de variables du premier ordre) deux à deux distinctes contenant au moins toutes les variables libres de  $\varphi$ , un *modèle* de  $\varphi(\mathbf{X}, \mathbf{x})$  est un triplet de la forme  $M, \mathbf{A}, \mathbf{a}$  pour :

1.  $M$  une structure plate ,
2.  $\mathbf{A} = (A_1, \dots, A_m)$  un vecteur de parties de  $E_M$ ,



3.  $\mathbf{a} = (a_1, \dots, a_n)$  un vecteur de points de  $E_M$ ,

tel que  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  est dérivable à l'aide des règles suivantes :

1. lorsque  $\varphi = R(t_1, \dots, t_k)$ ,  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  ssi, en notant pour tout  $i \in \{1, \dots, k\}$ ,  $t_{iM} = c_M$  si  $t_i = c$  et  $t_{iM} = a_j$  si  $t_i = x_j$ ,  $(t_{1M}, \dots, t_{kM}) \in R_M$ ,
2. lorsque  $\varphi = t \in X$ ,  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  ssi, avec les mêmes notations,  $t_M \in A_i$  avec  $X = X_i$ ,
3. lorsque  $\varphi = \neg\psi$ ,  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  ssi  $M \models \psi[\mathbf{A}, \mathbf{a}]$  est faux,
4. lorsque  $\varphi = \varphi_1 \wedge \varphi_2$ ,  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  ssi  $M \models \varphi_1[\mathbf{A}, \mathbf{a}]$  et  $M \models \varphi_2[\mathbf{A}, \mathbf{a}]$ ,
5. lorsque  $\varphi = \forall x\psi$ ,  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  ssi pour tout  $b \in E_M$ ,  $M \models (\psi[x_{n+1}/x])[\mathbf{A}, \mathbf{a}']$ , où  $\psi[x_{n+1}/x]$  désigne la formule  $\psi$  dans laquelle toutes les occurrences libres de la variable  $x$  ont été remplacées par une variable  $x_{n+1}$  distincte de  $x_1, \dots, x_n$  et  $\mathbf{a}' = (a_1, \dots, a_n, b)$ ,
6. lorsque  $\varphi = \forall X\psi$ ,  $M \models \varphi[\mathbf{A}, \mathbf{a}]$  ssi pour tout  $B \subseteq E_M$ ,  $M \models (\psi[X_{m+1}/X])[\mathbf{A}', \mathbf{a}]$  avec les mêmes conventions d'écriture et  $\mathbf{A}' = (A_1, \dots, A_m, B)$ .

On dit que deux formules  $\varphi$  et  $\psi$  sont *équivalentes* sur une classe de structures  $\mathcal{C}$ , ou encore  *$\mathcal{C}$ -équivalentes*, lorsqu'elles admettent les mêmes modèles sur la classe  $\mathcal{C}$ .

Deux structures  $S_1$  et  $S_2$  satisfaisant les mêmes formules d'un ensemble  $\Phi \subseteq LMSO$  sont dites *non séparables* ou bien *indiscernables* par  $\Phi$ .

## A.2 Formules du premier ordre, égalité et jeux

On introduit les constantes  $\top$  et  $\perp$  ainsi que le prédicat d'égalité de la façon suivante :

1. les constantes vrai et faux :  $\top$  et  $\perp$  respectivement interprétées comme les formules  $\forall x \exists X x \in X$  et  $\exists x \forall X x \in X$ ,
2. le prédicat d'égalité :  $t_1 = t_2$  interprété comme la formule  $\forall X t_1 \in X \Leftrightarrow t_2 \in X$ .

**Remarque.** On vérifie que les constantes  $\top$  et  $\perp$  admettent respectivement toutes structures et aucune structure comme modèles. On vérifie aussi que pour tous termes  $t_1$  et  $t_2$ ,  $M \models t_1 = t_2$  ssi  $t_{1M} = t_{2M}$ .

Une *MS*-formule  $\varphi$  est alors dite du *premier ordre sans égalité* (resp. du *premier ordre avec égalité*) lorsqu'elle est construite sans quantificateur du second ordre à l'exception de ceux utilisés pour définir  $\top$  et  $\perp$  (resp.  $\top$ ,  $\perp$  et le prédicat d'égalité).

L'ensemble des formules du premier ordre avec égalité est noté  $LPO$ , l'ensemble des formules du premier ordre sans égalité est noté  $LPOSEG$ .

Deux structures satisfaisant les mêmes formules de la logique du premier ordre avec égalité sont dites *élémentairement équivalentes*.

On doit à Fraïssé [18] une caractérisation algébrique de l'équivalence élémentaire qui, reprise par Ehrenfeucht [16] sous la forme de jeux, se révèle un outil puissant de démonstration. Utilisés informellement dans cette thèse sous le nom de jeux de Ehrenfeucht et Fraïssé, ou jeux de  $E\&F$ , on en donne ici une présentation rapide et légèrement étendue afin de traiter le cas de la logique du premier ordre sans prédicat d'égalité. Le lecteur souhaitant avoir plus de détails pourra consulter la présentation de Thomas [45].

Soit  $\Phi(\mathbf{x})$  un ensemble de formules du premier ordre sur le vocabulaire  $L$ , sans quantifications, pouvant contenir le prédicat d'égalité et admettant au plus  $k$  variables libres parmi les variables de  $\mathbf{x} = (x_1, \dots, x_k)$ .

Étant donné deux  $L$ -structures  $M$  et  $N$ , étant donné deux vecteurs de points  $\mathbf{a} = (a_1, \dots, a_k) \in E_M^k$  et  $\mathbf{b} = (b_1, \dots, b_k) \in E_N^k$ , on dit que les structures  $\langle M, \mathbf{a} \rangle$  et  $\langle N, \mathbf{b} \rangle$  sont  $\Phi(\mathbf{x})$ -équivalentes, lorsque, pour toute formule  $\varphi(\mathbf{x}) \in \Phi(\mathbf{x})$ ,  $M \models \varphi[\mathbf{a}]$  ssi  $N \models \varphi[\mathbf{b}]$ .

**Remarque.** Les formules de  $\Phi(\mathbf{x})$  étant sans quantifications, en notant  $M(\mathbf{a})$  et  $N(\mathbf{b})$  les sous-structures de  $M$  et  $N$  induites respectivement par les points de  $\mathbf{a}$  et  $\mathbf{b}$ , les structures  $\langle M, \mathbf{a} \rangle$  et  $\langle N, \mathbf{b} \rangle$  sont  $\Phi(\mathbf{x})$ -équivalentes ssi les structures  $\langle M(\mathbf{a}), \mathbf{a} \rangle$  et  $\langle N(\mathbf{b}), \mathbf{b} \rangle$  sont  $\Phi(\mathbf{x})$ -équivalentes. On notera donc dans la suite

$$M(\mathbf{a}) \simeq_{\Phi(\mathbf{x})} N(\mathbf{b})$$

cette équivalence.

Un *jeu de Ehrenfeucht et Fraïssé* en  $k$  coups sur les structures  $M$  et  $N$ , à partir de l'ensemble de formules  $\Phi(\mathbf{x})$ , est alors défini comme un jeu entre deux joueurs  $A$  et  $B$  se déroulant en  $k$  tours de jeu de la façon suivante : pour chaque tour de jeu,

1. le joueur  $A$  choisit un point  $e$  sur l'une des structures  $M$  ou  $N$ ,
2. le joueur  $B$  choisit un point  $f$  sur l'autre structure.

À la fin des  $k$  tours de jeu, les joueurs  $A$  et  $B$  ont effectué ensemble et dans l'ordre, une suite de  $k$  choix de points  $\mathbf{a} = (a_1, a_2, \dots, a_k)$  sur la structure  $M$  et une suite de  $k$  choix de points  $\mathbf{b} = (b_1, b_2, \dots, b_k)$  sur la structure  $N$ , certain point pouvant avoir été choisi plusieurs fois.

Le joueur  $B$  est déclaré gagnant lorsque  $M(\mathbf{a}) \simeq_{\Phi(\mathbf{x})} N(\mathbf{b})$ , sinon, le joueur  $A$  est déclaré gagnant.

Sur un tel jeu, on dit que le joueur  $B$  a une stratégie gagnante ssi il lui est possible de gagner toute partie.

En notant  $L(\Phi(\mathbf{x}))$  l'ensemble des formules obtenues en quantifiant arbitrairement toutes les variables libres de n'importe quelle formule  $\varphi(\mathbf{x})$  de  $\Phi(\mathbf{x})$  on a équivalence de :

1. pour toute formule  $\psi \in L(\Phi(x))$ ,  $M \models \psi$  ssi  $N \models \psi$ ,
2. le joueur  $B$  a une stratégie gagnante pour le jeu de Ehrenfeucht et Fraïssé en  $k$  coups, à partir de l'ensemble de formules  $\Phi(x)$ .

**Remarque.** Dans le cas où  $\Phi(\mathbf{x})$  est l'ensemble de toutes les formules sans quantificateurs dont les variables sont à choisir parmi  $\mathbf{x}$ , pour toutes  $L$ -structures  $M$  et  $N$ , on a équivalence de :

1. pour toute formule close  $\psi$  ayant au plus  $k$  quantificateurs,  $M \models \psi$  ssi  $N \models \psi$ ,
2. le joueur  $B$  a une stratégie gagnante pour le jeu de Ehrenfeucht et Fraïssé en  $k$  coups, à partir de l'ensemble de formules  $\Phi(x)$ .

Notons que dans ce cas, pour tous vecteurs  $\mathbf{a} \in E_M^k$  et  $\mathbf{b} \in E_N^k$  on a  $M(\mathbf{a}) \simeq_{\Phi(x)} N(\mathbf{b})$  ssi la relation  $R = \{(a_i, b_i)\}_{i \in \{1, \dots, k\}}$  induit un isomorphisme entre  $E_M(\mathbf{a})$  et  $E_M(\mathbf{b})$ . C'est en fait la condition usuelle des jeux de Ehrenfeucht et Fraïssé. Avec cette modification il apparaît que deux structures  $M$  et  $N$  sont élémentairement équivalentes ssi pour tout entier  $k$  le joueur  $B$  admet une stratégie gagnante pour le jeu en  $k$  coups.



## Annexe B

# Présentation logique des domaines

Dans cette annexe, on définit une présentation logique des domaines. Les domaines y sont en effet construits comme des ensembles de théories logiques consistantes ordonnées par implication inverse. On montre, fait banal mais souvent oublié ou ignoré, que le théorème de compacité de la logique du premier ordre nous permet de garantir facilement que de tels ensembles ordonnés seront effectivement des domaines. Quoique découverte indépendamment, notre approche étend, avec un point de vue légèrement différent, les travaux de Palmgren et Stoltenberg-Hansen [37].

On ne peut que citer aussi les travaux d'Abramsky [1] ou Robinson [41] qui connectent la théorie des domaines avec la théorie de la démonstration. Pour ce qui est des rapports avec la logique, notre approche est cependant plus générale car Abramsky ou Robinson suppose implicitement et systématiquement que ce que nous appelons les domaines de théories et les domaines de modèles sont isomorphes.

La pertinence de ces travaux reste cependant à examiner avec plus de profondeur. Il semble en effet que tout ce qui est présenté ici était largement connu, à défaut d'être explicité, par le fondateur de la théorie des domaines : Scott lui-même [29].

On retiendra de cette présentation qu'elle apparaît comme un outil puissant qui devrait permettre de reconstruire ou retrouver sans peine bon nombre de constructions de domaines et de résultats topologiques en informatique théorique.

## B.1 Définitions usuelles des domaines sémantiques

Dans cette section, on rappelle les définitions usuelles portant sur les domaines [19]. On insiste en particulier sur la notion de clôture par idéal qui est un outil fondamental pour la construction de domaine en explicitant ses propriétés essentielles.

### B.1.1 Complétudes et idéaux

Soit  $\langle D, \leq \rangle$  un ensemble partiellement ordonné.

**B.1.1.1 Définition.** Une partie  $E \subseteq D$  est dite *dirigée* lorsque, pour tous éléments  $x$  et  $y \in E$  il existe  $z \in E$  tel que  $x \leq z$  et  $y \leq z$ .

**B.1.1.2 Définition ( $D$ -complétude et  $C$ -complétude).** L'ensemble  $\langle D, \leq \rangle$  est dit  *$d$ -complet* ou encore *dirigé complet* lorsque toute partie dirigée  $E \subseteq D$  admet une borne supérieure notée  $\sqcup E$ . En particulier, comme la partie vide est dirigée, tout ensemble  $d$ -complet admet un minimum.

L'ensemble  $\langle D, \leq \rangle$  est dit  *$c$ -complet* ou encore *conditionnellement complet* lorsque toute partie  $E \subseteq D$  bornée admet une borne supérieure toujours notée  $\sqcup E$ . En particulier, la partie vide étant bornée, tout ensemble  $c$ -complet admet un minimum.

**B.1.1.3 Définition.** On appelle *idéal* de  $D$  toute partie  $I \subseteq D$  telle que :

1.  $I$  est non vide,
2.  $I$  est une partie commençante, i.e. pour tout  $x \in D$ , s'il existe  $y \in I$  tel que  $x \leq y$  alors  $x \in I$ ,
3.  $I$  est dirigée, i.e. pour tout  $x$  et  $y \in D$  il existe  $z \in I$  majorant  $x$  et  $y$ .

On appelle alors *clôture par idéaux* de  $D$  l'ensemble  $I(D)$  des idéaux de  $D$  ordonné par inclusion.

**B.1.1.4 Proposition.** Lorsque l'ensemble  $D$  admet un minimum, la clôture par idéal  $\langle I(D), \subseteq \rangle$  de  $\langle D, \leq \rangle$  est un ensemble  $d$ -complet. Dans cet ensemble d'idéaux, le supremum de toute partie dirigée est l'union des idéaux qui la composent. De plus, ce résultat tient toujours lorsque  $\leq$  n'est qu'un préordre sur  $D$ .

*Preuve.* Soit  $\mathcal{I}$  une partie dirigée de  $\langle I(D), \subseteq \rangle$ . Si  $\mathcal{I}$  est vide, alors le supremum de  $\mathcal{I}$  est l'idéal  $\{x_0\}$  pour  $x_0$  l'élément minimum de  $D$ . Lorsque  $\mathcal{I}$  n'est pas vide, on montre que  $\bigcup \mathcal{I}$  est bien un idéal, et donc, c'est le supremum  $\sqcup \mathcal{I}$  de la partie dirigée  $\mathcal{I}$ .

1.  $\mathcal{I}$  n'est pas vide, donc par définition d'un idéal,  $\bigcup \mathcal{I}$  n'est pas vide.

2. Soit  $x \in \bigcup \mathcal{I}$  et soit  $I_x \in \mathcal{I}$  tel que  $x \in I_x$ . Comme  $I_x$  est un idéal, pour tout  $y \in \bigcup \mathcal{I}$  tel que  $y \leq x$  on a bien  $y \in I_x$  et donc  $y \in \bigcup \mathcal{I}$ . Autrement dit,  $\bigcup \mathcal{I}$  est une partie commençante de  $D$ .
3. Soient  $x$  et  $y \in \bigcup \mathcal{I}$  et deux idéaux  $I_x$  et  $I_y \in \mathcal{I}$  les contenant. Comme  $\mathcal{I}$  est une partie dirigée, il existe  $J \in \mathcal{I}$  tel que  $I_x \cup I_y \subseteq J$ , c'est à dire  $x$  et  $y \in J$ . On en déduit donc, comme  $J$  est un idéal qu'il existe bien un élément  $z \in J$  a fortiori dans  $\bigcup \mathcal{I}$  tel que  $x \leq z$  et  $y \leq z$ . Autrement dit,  $\bigcup \mathcal{I}$  est bien dirigée.

L'extension au cas où  $\leq$  n'est qu'un préordre est immédiate, l'extension de la définition d'un idéal à ce cas assurant que tout idéal est clos par l'équivalence induite par ce préordre.  $\square$

**B.1.1.5 Définition.** Pour tout ensemble ordonné  $D$ , on appelle *injection canonique* de  $D$  dans  $I(D)$  la fonction  $\iota : D \rightarrow I(D)$  définie pour tout  $x \in D$  par  $\iota(x) = \{y \in D \mid y \leq x\}$ .

Cette injection canonique est bien entendu un plongement de  $\langle D, \leq \rangle$  dans  $\langle I(D), \subseteq \rangle$ , i.e. pour tous éléments  $x$  et  $y \in D$ ,  $x \leq y$  ssi  $\iota(x) \subseteq \iota(y)$ .

**Remarque.** Remarquons cependant que cette clôture par idéaux n'est pas, en général, le plus petit ensemble ordonné  $d$ -complet dans lequel  $\langle D, \leq \rangle$  se plonge. Pour s'en convaincre, il suffit de considérer  $\mathbb{N} \cup \{\infty\}$  ordonné de façon usuelle. C'est bien un ensemble  $d$ -complet, mais sa clôture par idéaux construit deux "infinis" : l'idéal  $\mathbb{N}$  et l'idéal  $\mathbb{N} \cup \{\infty\}$ .

## B.1.2 Compacité et domaine

**B.1.2.1 Définition.** Un élément  $x \in D$  est dit *compact* lorsque, pour toute partie dirigée  $E \subseteq D$  admettant une borne supérieure, si  $x \leq \sqcup E$  alors il existe  $y \in E$  tel que  $x \leq y$ . L'ensemble des éléments compacts de  $D$  est noté  $\text{Cp}(D)$ .

**B.1.2.2 Définition.** Un ensemble  $B \subseteq D$  est appelé *base génératrice* ou plus simplement *base* de  $D$  lorsque, pour tout élément  $x \in D$ ,

$$x = \sqcup \{y \in B \mid y \leq x\}$$

**B.1.2.3 Définition (Domaine).** Un domaine de Scott-Ershov est un ensemble partiellement ordonné,  $d$ -complet, dont l'ensemble des éléments compacts forme une base génératrice.

**B.1.2.4 Théorème.** Pour tout ensemble ordonné  $D$  admettant un minimum, la clôture par idéaux  $\langle I(D), \subseteq \rangle$  de  $D$  est un domaine dont les éléments compacts sont tous les idéaux

de la forme  $\iota(x)$  avec  $x \in D$  pour  $\iota$  l'injection canonique de  $D$  dans  $I(D)$ . De surcroît, lorsque tous les éléments de  $D$  sont compacts, ce domaine est effectivement le plus petit domaine dans lequel  $D$  se plonge dans la mesure où ce plongement préserve la compacité.

*Preuve.* On a vu dans la proposition B.1.1.4 que  $I(D)$  ordonné par inclusion est bien  $d$ -complet. Il est immédiat que  $\iota(D)$  est une base génératrice de  $I(D)$ . Soit alors  $\mathcal{I} \subseteq I(D)$  une partie dirigée de  $I(D)$  et un idéal  $I \in \iota(D)$  tel que  $I \subseteq \sqcup \mathcal{I} = \bigcup \mathcal{I}$ . Par définition, il existe  $x \in D$  tel que  $I = \iota(x)$ . On montre alors qu'il existe un idéal  $J \in \mathcal{I}$  tel que  $I \subseteq J$ . En effet, si ce n'est pas le cas, pour tout  $J \in \mathcal{I}$  on a  $x \notin J$  et donc  $x \notin \bigcup \mathcal{I}$  ce qui est absurde. Ainsi, tout élément de  $\iota(D)$  est compact. Réciproquement, soit  $I$  un élément compact de  $I(D)$ , on a bien  $I \in \iota(D)$ . En effet,  $\mathcal{I} = \{\iota(x) \mid x \in I\}$  est une partie dirigée de  $I(D)$  avec  $I \subseteq \bigcup \mathcal{I}$ . Il existe donc, comme  $I$  est compact, un élément  $x \in I$  tel que  $I \subseteq \iota(x)$ . Or, avec  $x \in I$  on a aussi, par construction,  $\iota(x) \subseteq I$ , d'où on déduit  $I = \iota(x)$ .

Supposons alors que tous les éléments de  $D$  sont compacts. Soit  $\langle D_1, \leq \rangle$  un domaine et un plongement  $f : D \rightarrow D_1$ , i.e. tel que, pour tous  $x$  et  $y \in D$ ,  $x \leq y$  ssi  $f(x) \leq f(y)$ , satisfaisant en outre la condition que pour tout élément de  $D$ ,  $f(d)$  est compact. Cette fonction s'étend naturellement en une fonction  $g : I(D) \rightarrow D_1$  en posant, pour tout idéal  $I \in I(D)$ ,

$$g(I) = \sqcup f(I)$$

Pour tous idéaux  $I$  et  $J \in I(D)$ , si  $I \subseteq J$  alors  $g(I) \leq g(J)$ . Il nous reste à montrer la réciproque. Supposons donc que  $g(I) \leq g(J)$ . On a donc, pour tout  $x \in I$ ,  $f(x) \leq g(J) = \sqcup f(J)$ . Mais, par hypothèse, les éléments de  $f(D)$  sont compacts dans  $D_1$  et, comme  $J$  est dirigé,  $f(J)$  est dirigée. Il existe donc, pour tout  $x \in I$ ,  $y_x \in J$  tel que  $f(x) \leq f(y_x)$  et donc, par hypothèse,  $x \leq y_x \in J$ . Ainsi, pour tout  $x \in I$ ,  $\iota(x) \subseteq J$  et donc  $I = \bigcup_{x \in I} \iota(x) \subseteq J$ .  $\square$

**B.1.2.5 Corollaire.** *Tout domaine est isomorphe à la clôture par idéaux de l'ensemble de ses éléments compacts. Tout ensemble ordonné est isomorphe à l'ensemble des éléments compacts d'un domaine auquel on a éventuellement enlevé le minimum.*

**Remarque.** L'intuition sous-jacente à la notion d'éléments compacts pourrait être qu'ils représentent des approximations finies des autres points. Mais cette intuition de finitude ne tient pas puisque que l'ensemble ordonné défini par les éléments compacts d'un domaine



peut être arbitrairement complexe et irrégulier.

**B.1.2.6 Définition (Continuité).** Etant donné deux domaines  $\langle D_1, \leq \rangle$  et  $\langle D_2, \leq \rangle$ , on dit qu'une fonction  $f : D_1 \rightarrow D_2$  est *continue* lorsque, pour toute partie dirigée  $E \subseteq D_1$ ,  $f(\sqcup E) = \sqcup f(E)$ . En particulier, une telle fonction est croissante.

### B.1.3 “Power-domains”

On termine cette présentation par une définition générale de la notion de “power-domain” suivie des définitions usuelles des “power-domains” continus de Hoare, Smyth et Plotkin.

**B.1.3.1 Définition (“Power-domain”).** Etant donné  $\langle D, \leq \rangle$  un domaine, on appelle “power-domain” de  $D$  un couple  $\langle PD, \{\cdot\}, \cup, \rangle$  pour :

1. un domaine  $PD$ ,
2. une fonction “singleton”  $\{\cdot\} : D \rightarrow PD$ ,
3. une fonction d’“union”  $\cup : PD \times PD \rightarrow PD$ .

On dira que  $PD$  admet l’élément “vide” lorsqu’il existe un élément  $0 \in PD$  qui est le neutre de l’opération d’“union”. Bien entendu, un tel élément, s’il existe, est unique.

**B.1.3.2 Définition (“Power-domain” continu).** Un power-domain  $PD$  de  $D$  est dit continu lorsque l’opération de singleton est continue et l’opération d’union bi-continue.

On suppose dorénavant que  $\langle D, \leq \rangle$  est un domaine dont on note  $Cp(D)$  l’ensemble des éléments compacts.

Sur  $\mathcal{P}_{fin}(Cp(D))$  on définit les pré-ordres de Hoare, Smyth et Plotkin de la façon suivante : pour toute parties finies  $P$  et  $Q \subseteq Cp(D)$ ,

1.  $P \sqsubseteq_{hoare} Q$  lorsque pour tout  $x \in P$  il existe  $y \in Q$  tel que  $x \leq y$ ,
2.  $P \sqsubseteq_{smyth} Q$  lorsque pour tout  $y \in Q$  il existe  $x \in P$  tel que  $x \leq y$ ,
3.  $P \sqsubseteq_{plotkin} Q$  lorsque  $P \sqsubseteq_{hoare} Q$  et  $P \sqsubseteq_{smyth} Q$ .

**B.1.3.3 Théorème.** *Les clôtures par idéaux des domaines*

$$\langle \mathcal{P}_{fin}(Cp(D)), \sqsubseteq_{hoare} \rangle, \langle \mathcal{P}_{fin}(Cp(D)), \sqsubseteq_{smyth} \rangle \text{ et } \langle \mathcal{P}_{fin}^*(Cp(D)), \sqsubseteq_{plotkin} \rangle$$

sont des “power-domains” continus de  $\langle D, \leq \rangle$  appelés respectivement “power-domains” de Hoare, Smyth et Plotkin.

Dans ces domaines, le singleton d'un élément  $x \in D$  est défini comme le plus petit idéal contenant  $\{\{y\} \in \mathcal{P}(Cp(D)) \mid y \leq x\}$ .

De même, l'union de deux idéaux  $I$  et  $J$  est défini comme le plus petit idéal contenant  $\{U \cup V \mid U \in I \wedge V \in J\}$ .

*Preuve.* Classique [19]. □

## B.2 Construction logique des domaines

Dans cette section, on montre comment les domaines sémantiques peuvent apparaître en théorie des modèles du premier ordre. Une approche similaire est définie dans [37]. Dans la fin de cette de section, on explicite cette similitude.

### B.2.1 Définition

Soit  $L$  un vocabulaire et  $\mathcal{C}$  une classe non vide de  $L$ -structures *définissable en logique du premier ordre*, i.e. il existe une formule  $\varphi_{\mathcal{C}}$  de logique du premier ordre telle que

$$\mathcal{C} = \{M \mid M \models \varphi_{\mathcal{C}}\}$$

et soit  $\Phi$  un ensemble de formules du premier ordre sur le vocabulaire  $L$  tel qu'au moins une formule  $\varphi \in \Phi$  admet un modèle sur  $\mathcal{C}$ .

**Notation.** Dans la suite, étant donnée  $M$  une  $L$ -structure, on note  $Th(M)$  l'ensemble des formules  $\varphi$  de la logique du premier ordre sur le vocabulaire  $L$  telles que  $M \models \varphi$ . Étant aussi donné  $T \subseteq LPO$  un ensemble de formules sur le même vocabulaire, on note  $mod(T)$  la classe des  $L$ -structures  $M$  telles que, pour toute formule  $\varphi \in T$ ,  $M \models \varphi$ .

**B.2.1.1 Définition (Préordre de spécialisation relativement à  $\Phi$ ).** Sur  $\mathcal{C}$ , le *préordre de spécialisation* de Scott [29], noté  $\preceq_{\Phi}$ , est défini, pour toutes  $L$ -structures  $M_1$  et  $M_2 \in \mathcal{C}$  par,

$$M_1 \preceq_{\Phi} M_2 \text{ lorsque } Th(M_1) \cap \Phi \subseteq Th(M_2)$$

i.e. toute formule appartenant à  $\Phi$  satisfaite par  $M_1$  est satisfaite par  $M_2$ . L'équivalence induite par ce préordre, appelée  *$\Phi$ -indiscernabilité*, est notée  $\simeq_{\Phi}$ .

**Notation.** On note  $\mathcal{C}/\Phi$  l'ensemble des classes de  $L$ -structures  $\Phi$ -indiscernables. L'ordre induit par  $\preceq_{\Phi}$  sur  $\mathcal{C}/\Phi$ , appelé *ordre de spécialisation*, est noté  $\leq_{\Phi}$ .

**B.2.1.2 Définition (Préordre sémantique relativement à  $\mathcal{C}$ ).** On étend la relation de satisfaction logique  $\models$  à toute structure  $M \in \mathcal{C}$  et toute partie  $T \subseteq \Phi$ , appelée aussi *théorie*, par

$$M \models T \text{ lorsque } \forall \varphi \in T, M \models \varphi$$

où  $M$  désigne n'importe quelle  $L$ -structure. Le *préordre sémantique*  $\preceq_{\mathcal{C}}$  est défini, pour tous ensembles de formules  $T_1$  et  $T_2 \subseteq \Phi$  par :

$$T_1 \preceq_{\mathcal{C}} T_2 \text{ lorsque } \text{mod}(T_2) \cap \mathcal{C} \subseteq \text{mod}(T_1)$$

On dit parfois que  $T_2$  est un *raffinement* de  $T_1$ . L'équivalence induite par ce préordre est notée  $\simeq_{\mathcal{C}}$ .

**B.2.1.3 Définition.** On appelle *clôture d'une théorie*  $T \subseteq \Phi$  vis-à-vis de  $\mathcal{C}$  la théorie

$$c_{\mathcal{C}}(T) = \bigcup \{T' \subseteq \Phi \mid T' \simeq_{\mathcal{C}} T\}$$

**B.2.1.4 Proposition.** *Pour toute théorie  $T \subseteq \Phi$ , on a  $c_{\mathcal{C}}(T) \simeq_{\mathcal{C}} T$ . Ainsi, toute classe de théories  $\mathcal{C}$ -équivalentes admet un représentant canonique : la clôture de toute les théories qu'elle contient. Sur l'ensemble de ces représentants, le préordre sémantique  $\preceq_{\mathcal{C}}$  coïncide avec l'inclusion.*

*Preuve.* Immédiate. □

**Notation.** On note  $T(\Phi)/\mathcal{C}$  l'ensemble de ces classes de théories  $\mathcal{C}$ -consistantes équivalentes sur  $\mathcal{C}$ . L'ordre induit par  $\preceq_{\mathcal{C}}$ , appelé *ordre sémantique*, est noté  $\leq_{\mathcal{C}}$ . La proposition précédente nous permet d'affirmer que cet ensemble ainsi ordonné est isomorphe à l'ensemble des clôtures des théories  $\mathcal{C}$ -consistantes ordonné par inclusion. Cet ensemble de clôtures de théories est alors appelé *présentation canonique* de  $T(\Phi)/\mathcal{C}$ . En notant  $TD(\Phi)$  l'ensemble des théories  $\mathcal{C}$ -consistantes dirigées vis à vis du préordre  $\preceq_{\mathcal{C}}$ , on note aussi  $TD(\Phi)/\mathcal{C}$  l'ensemble des clôtures de ces théories.

**B.2.1.5 Définition.** On dit qu'une théorie  $T \subseteq \Phi$  admet un représentant fini lorsqu'il existe une théorie finie  $T' \subseteq \Phi$  telle que  $T \simeq_{\mathcal{C}} T'$ .

**Remarque.** Dans le cas d'une théorie dirigée  $T$  la théorie  $T$  admet un représentant fini ssi elle admet un maximum vis à vis de  $\preceq_{\mathcal{C}}$ .

Il vient :

**B.2.1.6 Théorème.** *L'ensemble  $T(\Phi)/\mathcal{C}$  (resp. l'ensemble  $TD(\Phi)/\mathcal{C}$ ) ordonné par inclusion est un domaine de Scott-Ershov appelé le domaine des théories de  $\Phi$  modulo  $\mathcal{C}$  (resp.*

le domaine des théories dirigées de  $\Phi$  modulo  $\mathcal{C}$ ). L'ensemble de ses éléments compacts est exactement l'ensemble des éléments de  $T(\Phi)/\mathcal{C}$  (resp. des éléments de  $TD(\Phi)/\mathcal{C}$ ) admettant un représentant fini. De surcroît, le domaine  $T(\Phi)/\mathcal{C}$  est conditionnellement complet.

*Preuve.* Il est immédiat que  $T(\Phi)/\mathcal{C}$  (resp.  $TD(\Phi)/\mathcal{C}$ ) ordonné par  $\leq_{\mathcal{C}}$  est  $d$ -complet, le supremum d'une partie dirigée étant la classe des théories équivalentes à l'union des théories apparaissant dans cette partie, le théorème de compacité nous assurant qu'une telle union de théories est  $\mathcal{C}$ -consistante. De plus, il est tout aussi évident que l'ensemble des classes de théories finies (resp. des classes de théories dirigées finies) est une base génératrice de  $T(\Phi)/\mathcal{C}$  (resp.  $TD(\Phi)/\mathcal{C}$ ). Il nous reste donc à montrer que c'est bien une base d'éléments compacts. Au passage, il suffit d'en faire la preuve dans le cas le plus général de  $T(\Phi)/\mathcal{C}$ .

Soit donc  $T = \{\varphi_1, \dots, \varphi_n\}$  une théorie  $\mathcal{C}$ -consistante finie, et soit  $\mathcal{I}$  une partie dirigée de théories telle que  $T \preceq_{\mathcal{C}} \sup \mathcal{I} \simeq_{\mathcal{C}} \bigcup \mathcal{I}$ . Autrement dit, en notant  $\varphi_{\mathcal{C}}$  la formule définissant  $\mathcal{C}$ , la théorie

$$\left\{ \bigvee_{i \in \{1, \dots, n\}} \neg \varphi_i, \varphi_{\mathcal{C}} \right\} \cup \bigcup \mathcal{I}$$

est inconsistante. Il existe donc, par compacité de la logique du premier ordre, une partie finie  $P \subseteq \bigcup \mathcal{I}$  telle que la théorie

$$\left\{ \bigvee_{i \in \{1, \dots, n\}} \neg \varphi_i, \varphi_{\mathcal{C}} \right\} \cup P$$

est inconsistante, c'est à dire c'est à dire

$$T \preceq_{\mathcal{C}} P$$

Or, comme  $P \subseteq \bigcup \mathcal{I}$ , il existe bien, pour toute formule  $\varphi \in P$  une théorie  $T_{\varphi} \in \mathcal{I}$  contenant  $\varphi$ . En appliquant alors le fait que  $\mathcal{I}$  est dirigée et que  $P$  est finie, il existe bien une théorie  $T_s \in \mathcal{I}$  telle que, pour toute formule  $\varphi \in P$ ,  $T_{\varphi} \preceq_{\mathcal{C}} T_s$ . On en déduit alors que  $T \preceq_{\mathcal{C}} T_s$  en usant du fait que  $T \preceq_{\mathcal{C}} P \preceq_{\mathcal{C}} \bigcup_{\varphi \in P} T_{\varphi}$ .  $\square$

**B.2.1.7 Définition.** L'ensemble de formules  $\Phi$  est dit *conditionnellement clos par conjonction* sur l'ensemble de structures  $\mathcal{C}$  lorsque les conditions suivantes sont satisfaites :

1.  $\varphi_{\mathcal{C}} \in \Phi$  d'où, pour tout  $\varphi \in \Phi$ ,  $\varphi_{\mathcal{C}} \preceq_{\mathcal{C}} \varphi$ ,
2. pour toute formule  $\psi \in \Phi$  il existe  $M \in \mathcal{C}$  tel que  $M \models \psi$ ,
3. si  $\{\varphi_1, \varphi_2\}$  est  $\mathcal{C}$ -consistant, i.e. la formule  $\varphi_1 \wedge \varphi_2$  admet un modèle sur  $\mathcal{C}$  alors il existe  $\psi \in \Phi$  telle que  $\psi \simeq_{\mathcal{C}} \varphi_1 \wedge \varphi_2$ .

**B.2.1.8 Proposition.** *Lorsque  $\Phi$  est conditionnellement clos par conjonction, les domaines  $T(\Phi)/\mathcal{C}$  et  $TD(\Phi)/\mathcal{C}$  sont isomorphes, les éléments compacts de ces domaines étant les classes des théories équivalentes à une formule de  $\Phi$ .*

*Preuve.* Immédiate. □

## B.2.2 Domaines de théories et domaines de modèles

Dans la partie précédente on a construit le domaine des théories de  $\Phi$ . Dans cette partie, on étudie les propriétés de l'ensemble  $\mathcal{C}/\Phi$  qui, lorsqu'il est effectivement un domaine, appelé ici *domaine des modèles*, semble [29] être historiquement le premier type de domaine considéré par Scott.

**B.2.2.1 Définition.** Soit  $Th_\Phi$  la fonction de  $\mathcal{C}/\Phi$  dans (la représentation canonique de)  $T(\Phi)/\mathcal{C}$  définie, pour tout modèle  $M \in \mathcal{C}$  par  $Th_\Phi([M]) = Th(M) \cap \Phi$ .

**Remarque.** Cette fonction est bien définie et c'est de plus un plongement de  $\mathcal{C}/\Phi$  dans  $T(\Phi)/\mathcal{C}$ , i.e. pour tous  $[M]$  et  $[N] \in \mathcal{C}/\Phi$ ,  $[M] \leq_\Phi [N]$  ssi  $Th_\Phi([M]) \subseteq Th_\Phi([N])$ .

Il vient :

**B.2.2.2 Théorème.** *L'ensemble  $\mathcal{C}/\Phi$  ordonné par l'ordre de spécialisation  $\leq_\Phi$  est d-complet ssi il admet un minimum.*

*Preuve.* Le sens direct est immédiat. Réciproquement, sans perte de généralité, i.e. sans que cela modifie le préordre de spécialisation, on peut supposer que  $\Phi$  est clos par conjonction et contient la formule  $\varphi_{\mathcal{C}}$  caractérisant  $\mathcal{C}$ .

Soit  $E \subseteq \mathcal{C}/\Phi$  une partie dirigée selon l'ordre de spécialisation. Lorsque  $E$  est vide le supremum de  $E$  est bien défini, c'est le minimum de  $\mathcal{C}/\Phi$ . Dans le cas contraire, le théorème de compacité nous permet de conclure. Posons en effet :

$$Th_\Phi(E) = \bigcup_{[M] \in E} Th_\Phi([M])$$

On montre alors qu'il existe un modèle  $M_E \in \mathcal{C}$  tel que  $Th_\Phi([M_E]) = Th_\Phi(E)$ , c'est à dire que la théorie

$$T = Th_\Phi(E) \cup \{\neg\varphi \mid \varphi \in \Phi - Th_\Phi(E)\}$$

est  $\mathcal{C}$ -consistante. Soit en effet  $T_1$  une partie finie de  $T$ . Pour toute formule  $\varphi \in T \cap Th_\Phi(E)$  il existe, par construction de  $Th_\Phi(E)$ , un modèle  $M_\varphi$  de  $\varphi$  tel que  $[M_\varphi] \in E$ . Or  $E$  est dirigée, il existe donc un majorant  $[M_1] \in E$  de  $\{[M_\varphi]\}_{\varphi \in T \cap Th_\Phi(E)}$ . De plus,  $Th_\Phi(M_1) \subseteq Th_\Phi(E)$ , il vient donc  $M_1 \models T_1$ . Autrement dit, toute partie finie de  $T$  admet un modèle,

le théorème de compacité nous assure que  $T$  admet aussi un modèle  $M_E$ . Comme  $\varphi_C \in T$ , on a de surcroît  $M_E \in \mathcal{C}$ . Par construction,  $Th_\Phi([M_E]) = Th_\Phi(E)$  donc  $[M_E]$  est un majorant de  $E$ . De plus, pour tout majorant  $[M]$  de  $E$ , on a  $M \models Th_\Phi(E) = Th_\Phi(M_E)$ , donc  $M_E \preceq_{\mathcal{C}} M$ . Ainsi,  $[M_E]$  est bien le supremum de  $E$ .  $\square$

**Remarque.** En fait, la démonstration précédente nous montre aussi que la fonction

$$Th_\Phi : \mathcal{C}/\Phi \rightarrow T(\Phi)/\mathcal{C}$$

est continue. En effet, pour toute partie dirigée  $E$  de  $\mathcal{C}/\Phi$ , on vient de montrer que  $Th_\Phi(\sqcup E) = \bigcup_{[M] \in E} Th_\Phi(M)$  et, dans la représentation canonique de  $T(\Phi)/\mathcal{C}$ , la borne supérieure de  $\{Th_\Phi(M)\}_{[M] \in E}$  qui est dirigée est bien l'union des théories qui la composent.

La définition suivante apparaît comme une caractérisation logique simple du fait que  $\mathcal{C}/\varphi$  et  $T(\Phi)/\mathcal{C}$  sont isomorphes.

**B.2.2.3 Définition.** L'ensemble  $\Phi$  *couvre simplement*  $\mathcal{C}$  lorsque, pour toute formule  $\varphi \in \Phi$ , toute partie finie  $A$  de  $\Phi$ , si  $\bigvee A \preceq \varphi$  alors il existe  $\psi \in A$  telle que  $\psi \preceq_{\mathcal{C}} \varphi$ .

**Remarque.** Lorsque  $\Phi$  couvre simplement  $\mathcal{C}$  la propriété de couverture simple reste vraie lorsque  $A$  est infinie. En effet, soient  $A$  une partie infinie de  $\Phi$  et une formule  $\varphi \in \Phi$  telles que

$$\bigvee A \preceq_{\mathcal{C}} \varphi$$

La théorie  $\{\varphi_C, \varphi\} \cup \{\neg\psi \mid \psi \in A\}$  est inconsistante. Il existe donc une partie finie  $A' \subseteq A$  tel que la théorie  $\{\varphi_C, \varphi\} \cup \{\neg\psi \mid \psi \in A'\}$  est inconsistante, c'est à dire

$$\bigvee A' \preceq_{\mathcal{C}} \varphi$$

Mais alors, comme  $\Phi$  couvre simplement  $\mathcal{C}$  il existe bien  $\psi \in A'$  tel que  $\psi \preceq_{\mathcal{C}} \varphi$ .

Il vient alors :

**B.2.2.4 Théorème.** Lorsque  $\Phi$  est conditionnellement clos par conjonction,  $\Phi$  couvre simplement  $\mathcal{C}$  ssi la fonction

$$Th_\Phi : \mathcal{C}/\Phi \rightarrow T(\Phi)/\mathcal{C}$$

est un isomorphisme.

*Preuve.* Supposons que  $Th_\Phi$  est un isomorphisme. Pour toute formule  $\varphi \in \Phi$  il existe  $M_\varphi \in \mathcal{C}$  tel que  $M_\varphi \models \varphi$  et, pour toute formule  $\psi \in \Phi$ , si  $M_\varphi \models \psi$  alors  $\psi \preceq_{\mathcal{C}} \varphi$ . En effet,

il suffit de prendre

$$[M_\varphi] = Th_\Phi^{-1}(\{\psi \in \Phi \mid \psi \preceq_C \varphi\})$$

Soit alors un ensemble fini de formules  $A \subseteq \Phi$  et  $\varphi \in \Phi$  telle que :

$$\bigvee A \preceq_C \varphi$$

Par définition, on a immédiatement  $M_\varphi \models \bigvee A$ , il existe donc  $\psi \in A$  tel que  $M_\varphi \models \psi$  ce dont on déduit, par construction de  $M_\varphi$ ,  $\psi \preceq_C \varphi$ .

Réciproquement, supposons que  $\Phi$  couvre simplement  $\mathcal{C}$ . Pour montrer que  $Th_\Phi$  est un isomorphisme, il suffit de montrer que tout élément compact de  $T(\Phi)/\mathcal{C}$  admet par  $Th_\Phi$  un antécédent dans  $\mathcal{C}/\Phi$ . En effet, si c'est le cas, pour toute théorie  $T \in T(\Phi)/\mathcal{C}$  le théorème précédent nous permet d'affirmer que  $Th^{-1}(T)$  existe avec

$$Th^{-1}(T) = \sqcup \{Th_{-1}(c_C(\varphi)) \mid \varphi \in T\}$$

Comme  $\Phi$  est supposée conditionnellement close par conjonction, les éléments compacts de  $T(\Phi)/\mathcal{C}$  sont exactement les théories de la forme  $c_C(\varphi) = \{\psi \in \Phi \mid \psi \preceq_C \varphi\}$  pour  $\varphi$  variant dans  $\mathcal{C}$ .

Soit donc  $\varphi \in \Phi$ . On pose

$$mod_C(\varphi) = \{M \in \mathcal{C} \mid M \models \varphi\}$$

Montrons alors que  $mod_C(\varphi)$  admet un minimum. En effet, si ce n'est pas le cas, pour tout modèle  $M$  de  $\varphi$  il existe un modèle  $N_M$  de  $\varphi$  et une formule  $\psi_M$  tels que  $M \models \psi_M$  alors que  $N_M \not\models \psi_M$ , i.e.  $M \not\preceq_\Phi N$ . Il vient immédiatement

$$\bigvee_{M \in mod_C(\varphi)} \psi_M \preceq \varphi$$

Mais, par hypothèse de couverture simple, il existe  $M \in mod_C(\varphi)$  tel que  $\psi_M \preceq \varphi$ . On en déduit en particulier, comme  $N_M \models \varphi$ , que  $N_M \models \psi_M$  ce qui est contraire aux hypothèses.

Soit donc  $M_\varphi$  le minimum (modulo  $\simeq_C$ ) de  $mod_C(\varphi)$ . Par construction,  $M_\varphi \models \varphi$ . Soit alors  $\psi$  telle que  $M_\varphi \models \psi$ . On a nécessairement  $\psi \preceq_C \varphi$ . En effet, si ce n'est pas le cas, il existe  $M$  tel que  $M \models \varphi$  alors que  $M \not\models \psi$ . Mais alors  $M \in mod_C(\varphi)$  et donc, par construction de  $M_\varphi$ ,  $M_\varphi \preceq_\Phi M$ , d'où on déduit  $M \models \psi$  ce qui est absurde. Autrement dit,

$$Th_\Phi([M_\varphi]) = \{\psi \in \Phi \mid \psi \preceq_C \varphi\}$$

□

### B.2.3 Connexion avec l'approche de Palmgren et Stoltenberg-Hansen

Nous montrons ici comment notre approche de la présentation logique des domaines est fortement reliée à celle de Palmgren et Stoltenberg-Hansen [37].

Dans la suite,  $\Phi(\mathbf{x})$  désigne un ensemble de formules du premier ordre construites sur le vocabulaire  $L$  et les variables  $\mathbf{x} = x_1, x_2, \dots, x_n$ .

**B.2.3.1 Définition.** Etant donné une  $L$ -structure  $M$ , on dit que  $\varphi(\mathbf{x})$  est *réalisable* sur  $M$  lorsqu'il existe un  $n$ -uplet  $\mathbf{a} = a_1, \dots, a_n \in E_M^n$  de l'univers de  $M$  tels que

$$M, \mathbf{a} \models \varphi(\mathbf{x})$$

Le  $n$ -uplet  $\mathbf{a}$  est alors appelé une *réalisation* de  $\Phi(\mathbf{x})$  dans  $M$ . Par extension, un ensemble de formules  $T(\mathbf{x}) \subseteq \Phi(\mathbf{x})$  est dit aussi *réalisable* sur  $M$  lorsqu'il existe un  $n$ -uplet  $\mathbf{a} \in E_M^n$  qui soit une réalisation sur  $M$  de toute formule  $\varphi(\mathbf{x}) \in T(\mathbf{x})$ .

Un *type* de  $\Phi(\mathbf{x})$  est un ensemble de formules  $T(\mathbf{x}) \subseteq \Phi(\mathbf{x})$  dont toute partie finie est réalisable sur au moins une  $L$ -structure  $M$ .

**Remarque.** Le théorème de compacité de la logique du premier ordre nous assure qu'un tel ensemble de formules est un type ssi il est réalisable sur au moins une  $L$ -structure  $M$ . Il suffit en effet d'ajouter au vocabulaire  $L$  une constante par variable de  $\mathbf{x}$ . On note alors  $T(\Phi(\mathbf{x}))$  l'ensemble des types de  $\Phi(\mathbf{x})$ .

**B.2.3.2 Définition.** En notant, pour tout type  $T(\mathbf{x}) \subseteq \Phi(\mathbf{x})$ ,  $T_M$  l'ensemble des réalisations du type  $T(\mathbf{x})$  sur une structure  $M$  donnée, en notant  $\preceq_M$  le préordre sur  $T(\Phi(\mathbf{x}))$  défini pour tout type  $T_1(\mathbf{x})$  et  $T_2(\mathbf{x}) \in T(\Phi(\mathbf{x}))$  par :

$$T_1(\mathbf{x}) \preceq_M T_2(\mathbf{x}) \text{ lorsque } (T_2)_M \subseteq (T_1)_M$$

on dit que l'ensemble  $\Phi(\mathbf{x})$  est *conditionnellement clos par conjonction* sur la structure  $M$  lorsque :

1. toute formule  $\varphi(\mathbf{x}) \in \Phi(\mathbf{x})$  est réalisable dans  $M$ ,
2. il existe une formule  $\varphi_0(\mathbf{x}) \in \Phi(\mathbf{x})$  telle que, pour toute formule  $\varphi(\mathbf{x})$ ,  $\varphi_0(\mathbf{x}) \preceq_M \varphi(\mathbf{x})$ ,
3. pour toutes formules  $\varphi_1(\mathbf{x})$  et  $\varphi_2(\mathbf{x}) \in \Phi(\mathbf{x})$  telles qu'il existe au moins une réalisation commune de  $\varphi_1(\mathbf{x})$  et  $\varphi_2(\mathbf{x})$  sur  $M$ , i.e.  $\{\varphi_1(\mathbf{x})\}_M \cap \{\varphi_2(\mathbf{x})\}_M \neq \emptyset$ , il existe une formule  $\psi(\mathbf{x})$  telle que  $\{\psi(\mathbf{x})\}_M \simeq_M \{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})\}_M$  où  $\simeq_M$  désigne l'équivalence sur  $T(\Phi(\mathbf{x}))$  induite par le préordre  $\preceq_M$ .



**B.2.3.3 Théorème (Palmgren et Stoltenberg-Hansen).** *Pour tout ensemble  $\Phi(\mathbf{x})$  conditionnellement clos par conjonction sur une  $L$ -structure  $M$  supposée  $\omega$ -saturée, i.e. telle que tout type réalisable de logique du premier ordre admet une réalisation sur  $M$ , en notant  $T(\Phi(\mathbf{x}))/M$  l'ensemble  $T(\Phi(\mathbf{x}))$  des types réalisables de  $\Phi(\mathbf{x})$  quotienté par  $\simeq_M$  et ordonné par  $\preceq_M$ , en notant  $\Phi(x)/M$  l'ensemble de ces classes ayant un représentant de la forme  $\{\varphi(\mathbf{x})\}$  pour  $\varphi(\mathbf{x}) \in \Phi(\mathbf{x})$ , l'ensemble  $T(\Phi(\mathbf{x}))/M$  est un domaine de Scott-Ershov dont les éléments compacts sont exactement les éléments de  $\Phi(\mathbf{x})/M$ .*

*Preuve.* C'est ici le théorème central de [37]. C'est aussi une conséquence immédiate du théorème B.2.1.6 comme on peut le voir de la façon suivante.

Soit  $L(\mathbf{x})$  le vocabulaire  $L$  augmenté des variables  $x_1, \dots, x_n$  dès lors considérées comme des constantes. Soit  $\varphi_0(\mathbf{x})$  une formule minimum vis-à-vis du préordre  $\preceq_M$  dont l'existence est assurée par le fait que  $\Phi(\mathbf{x})$  est conditionnellement clos par conjonction sur  $M$ . Soit  $\mathcal{C}$  la classe des  $L(\mathbf{x})$ -structures qui sont modèles de  $\varphi_0(\mathbf{x})$ . Il est alors facile de voir que l'ensemble  $T(\Phi(\mathbf{x}))/\mathcal{C}$  des  $L(\mathbf{x})$ -théories consistantes de  $\Phi(\mathbf{x})$  quotienté et ordonné par  $\preceq_{\mathcal{C}}$  est isomorphe à l'ensemble  $T(\Phi(\mathbf{x}))/M$  des types de  $\Phi(x)$  quotienté et ordonné par  $\preceq_M$ . □



# Index

$\Phi$ -indiscernabilité .....	114	bien nommée .....	37
$\mu$ -indice .....	38	bisimilaires .....	24
$\mu$ -signature .....	40	bisimulation .....	23
$\mu$ -trace .....	46	branchement fini .....	18
$\mu$ -variable .....	37	$c$ -complet .....	110
$\nu$ -indice .....	38	$\mathcal{C}$ -équivalentes .....	105
$\nu$ -signature .....	41	calcul modal alternant .....	51
$\nu$ -trace .....	46	calcul .....	71
$\nu$ -variable .....	37	chemin fini .....	19
$\Omega$ -clôture .....	92	close .....	34
$\omega$ -expansions .....	23	close par bisimulation .....	97
$a$ -fils .....	18	clôture d'une théorie .....	115
$a$ -nœud .....	44	clôture par idéaux .....	110
$a$ -successeur .....	18	compact .....	111
acceptant .....	51	cond. clos par conjonction .....	116,120
accessible .....	19	conditionnellement complet .....	110
actions .....	17	consistant .....	47
angélique .....	12	continue (fonction) .....	113
arbre avec arêtes retours .....	67	coordonnées binaires gauches .....	81
arbre binaire .....	74	couvre simplement .....	118
arbre binaire de codage .....	80	$d$ -complet .....	110
arbre binaire infini .....	74	dirigée (partie) .....	110
arbre de transitions .....	21	dirigé complet (partie) .....	110
arbre gauche .....	80	disjonctive .....	8
automate d'arbres binaires .....	74	disjonctive (formule modale) .....	30
automate modal .....	71	disjonctive (formule du $\mu$ -calcul) .....	62
automate modal alternant .....	33,50	disjonctive binaire (formule) .....	75
base .....	111	domaine .....	103
base génératrice .....	111	domaine des modèles .....	117

- domaine des théories ..... 115
- domaine des théories dirigées ..... 116
- définition de la variable  $X$  ..... 37
- démoniaque ..... 12
- développement universel ..... 21
- développement ..... 19
- élémentairement équivalentes ..... 106
- équivalence de bisimulation ..... 24
- équivalence de simulation ..... 23
- équivalentes par bisimulation ..... 7
- équivalentes ..... 105
- état ..... 18
- état infinitaire ..... 93
- état initial ..... 18
- état nul ..... 93
- état retour ..... 53
- expansion (d'un système) ..... 21
- expansion  $\alpha$ -universelle ..... 23
- expansion (d'une formule) ..... 39
- fil ..... 18
- fil droit ..... 74
- fil gauche ..... 74
- fonction de liaison ..... 37
- fonction de liaison généralisée ..... 39,40
- forme normale arborescente ..... 53
- forme normale disjonctive ..... 5
- formules modales paramétrées ..... 27
- formules modales plates ..... 6
- formules gardées ..... 42
- formules homogènes ..... 3
- idéal ..... 110
- image commutative ..... 89
- immédiatement inconsistant ..... 56
- indiscernables ..... 105
- injection canonique ..... 111
- interprétation ..... 103
- jeu de Ehrenfeucht et Fraïssé ..... 106
- $L$ -automate ..... 82
- $L$ -segment ..... 84
- $L$ -sommet ..... 80
- $L$ -structure ..... 103
- lettres-équivalents ..... 90
- lexique ..... 78
- lexique induit ..... 80
- libre (variable) ..... 34,104
- liée (variable) ..... 34,104
- linéaire (ensemble) ..... 89
- littéral ..... 30,34
- longueur ..... 19
- $ML$ -automate ..... 90
- marquage ..... 46
- modèle ..... 104
- non séparables ..... 105
- nœud de choix ..... 44
- nœud modal ..... 44
- ordre de dépendance ..... 37
- ordre de spécialisation ..... 114
- ordre préfixe ..... 74
- ordre sémantique ..... 115
- $P$ -homogène (formule) ..... 3
- positive ..... 34
- power-domain ..... 113
- prédicat de typage ..... 2
- préfixe ..... 74,80
- premier ordre avec égalité ..... 105
- premier ordre sans égalité ..... 105
- préordre de spécialisation ..... 114
- préordre sémantique ..... 115
- présentation canonique ..... 115
- présentation lexicographique ..... 78

préserve la bisimulation .....	29
proches (noeuds dans un tableau) ....	44
quantification typée .....	2
raffinement .....	115
réalisable .....	120
réalisation .....	120
reconnu .....	51
régénération .....	45
relation de bisimulation .....	7
relation de transition .....	18
semi-linéaire (ensemble) .....	89
signature .....	103
simplement équivalents .....	65
simulation .....	23
sommet .....	18
sommet initial .....	18
source d'un retour .....	53
structure plate .....	2
successeurs .....	18
système de transitions .....	18
tableau .....	33,43
théorie .....	115
trace (dans un tableau) .....	45
trace (d'un automate) .....	51
type .....	2,120
type modal .....	7
univers .....	2,103
variables séparées .....	37
vocabulaire .....	103

