

N° d'ordre: 2911

THÈSE

présentée

devant l'université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Thomas COLCOMBET

Équipe d'accueil : Galion

École doctorale : MATISSE

Composante universitaire : IFSIC/IRISA

Titre de la thèse :

Représentations et propriétés de structures infinies

soutenue le 18 mars 2004 devant la commission d'examen

M. :	Erich	GRÄDEL	Président
MM. :	Bruno	COURCELLE	Rapporteurs
	Wolfgang	THOMAS	
MM. :	Damian	NIWIŃSKI	Examineurs
	Jean-Claude	RAOULT	
	Didier	CAUCAL	

Remerciements

Je tiens à remercier tout d'abord les membres du Jury de cette thèse. En premier lieu, je remercie Bruno Courcelle et Wolfgang Thomas qui ont tous deux accepté de rapporter ce travail en se pliant à un calendrier difficile. Bruno Courcelle, par ses nombreuses remarques et conseils a beaucoup participé à la qualité de ce travail. Je remercie Erich Grädel qui a tant fait pour la notoriété de la thématique. Je remercie Damian Niwiński qui en plus de m'accueillir à Varsovie et de me transmettre tant de connaissances, a accepté de participer à ce Jury. Je remercie Jean-Claude Raoult, qui a aussi gentiment accepté de faire partie de ce jury.

Enfin, je tiens à remercier Didier Caucal qui, entre autres facéties, a bien voulu diriger mes travaux de thèse. Ses conceptions de la recherche, de l'esprit de la recherche, ont une influence certaine sur mon futur.

Je remercie Tanguy Urvoy qui est un merveilleux collègue et ami. Je remercie Karine Heydemann qui a su aussi me soutenir en de nombreuses occasions.

Je remercie les différentes personnes avec lesquelles j'ai eu l'occasion de travailler et de publier durant cette thèse, Arnaud Carayol, Pascal Fradet et Christophe Löding.

Je remercie mes parents, qui m'ont laissé m'orienter vers un métier que j'aime, quitte à y blanchir quelques cheveux.

Pour finir, je remercie Marion, pour son soutien indéfectible. Sans elle, ce pénible accouchement n'aurait pu aboutir.

Table des matières

1	Introduction	1
2	Définitions et notions de base	8
2.1	Structures	9
2.1.1	Cas général	9
2.1.2	Graphes	10
2.1.3	Les automates	11
2.1.4	Arbres et termes	12
2.2	Logiques	15
2.2.1	Logique du premier ordre	15
2.2.2	Logique monadique	17
2.2.3	Extensions de la logique monadique	20
2.2.4	Construction de Feferman-Vaught	21
2.3	Opérations sur les structures	25
2.3.1	Modification finie	26
2.3.2	Interprétation logique	26
2.3.3	Union disjointe	29
2.3.4	Produits	30
2.3.5	Transduction monadique	35
2.3.6	Dépliage	36
2.3.7	Structure arborescente	37
2.4	Propriétés structurelles	42
2.4.1	La largeur arborescente	42
2.4.2	Graphe uniformément creux	44
2.4.3	Sous-graphes bipartis complets	45
3	Systèmes d'équations	47
3.1	Ordres partiels complets et algèbres	48
3.1.1	Définitions	48
3.1.2	Quelques ordres partiels complets	52
3.1.3	Algèbres et algèbres continues	54

3.2	Systèmes d'équations	56
3.2.1	Une première définition	56
3.2.2	Systèmes d'équations comme graphes	57
3.3	μ -définissabilité et μ -calcul	59
3.3.1	Généralités sur les points fixes	60
3.3.2	Syntaxe et sémantique du μ -calcul	63
3.3.3	Décomposition du μ -calcul	65
3.3.4	Extension vectorielle du μ -calcul	65
3.3.5	Termes marqués	67
3.3.6	μ -définissabilité	68
3.4	Transducteurs déterministes	72
3.4.1	Définition	72
3.4.2	Normalisation	75
3.4.3	Propriétés	77
3.4.4	Lien avec les systèmes d'équations	80
3.4.5	Utilisation des transducteurs déterministes	83
4	Graphes et structures sur les piles	89
4.1	Langages rationnels et systèmes d'équations	90
4.1.1	Automates	90
4.1.2	Automates déterministes	90
4.2	Les graphes préfixes	91
4.3	Les structures HR-équationnelles	94
4.3.1	Première approche : grammaires de structures	94
4.3.2	Représentation équationnelle	95
4.3.3	Propriétés des graphes HR-équationnels	99
4.4	Graphes préfixes reconnaissables et extensions	101
4.4.1	Représentation équationnelle	103
4.4.1.1	Opérateurs sur les graphes	103
4.4.1.2	Extension aux structures	105
4.4.1.3	Un opérateur de fusion supplémentaire	109
4.4.2	Représentation interne	116
4.4.2.1	Graphes préfixes reconnaissables	116
4.4.2.2	Les relations PR	118
4.4.3	Représentation par transformations	121
4.4.3.1	Premier contact	121
4.4.3.2	Des transductions monadiques vers les relations PR122	
4.4.4	Application à la hiérarchie de Caucal	123
4.4.5	Liens entre VR et HR	123
4.4.5.1	Intuitions sur la preuve	124
4.4.5.2	Normalisation	125

4.4.5.3	Premières propriétés	126
4.4.5.4	Hypothèses de récurrence	127
4.4.5.5	Le résultat	128
5	Structures automatiques sur les termes	129
5.1	Langages de termes	129
5.1.1	Automates d'arbres	129
5.1.1.1	Le cas général	130
5.1.1.2	Automates descendants déterministes	131
5.1.2	Représentation équationnelle	131
5.1.2.1	Langages rationnels	131
5.1.2.2	Automates descendants déterministes	131
5.2	Structures terme-automatiques	132
5.2.1	Présentation terme-automatique	133
5.2.1.1	Relations terme-synchronisées	133
5.2.1.2	Structures à présentation terme-automatique	134
5.2.2	Représentation équationnelle	135
5.2.2.1	Définition	135
5.2.2.2	Normalisation	136
5.2.2.3	Équivalence avec la représentation automatique	138
5.2.3	Représentation logique	141
5.2.3.1	Interprétation des parties	141
5.2.3.2	D'un système d'équations à la représentation logique	143
5.2.3.3	De la représentation logique à un système d'équations	146
5.2.4	Structures automatiques	149
6	Graphes de réécriture suffixe de termes	151
6.1	Représentation interne	151
6.1.1	Contexte	152
6.1.2	Graphes de réécriture suffixe de termes	153
6.2	Représentation équationnelle	155
6.3	Équivalence de représentations	156
6.3.1	Des équations aux récritures	158
6.3.2	De la réécriture aux équations	159
6.4	Premier ordre avec accessibilité	160
6.5	Largeur arborescente	164
6.5.1	Croissance pour la largeur arborescente	164
6.5.2	Élimination des produits	166
6.5.3	Décidabilité et effectivité	171
6.6	Largeur de clique	172

6.6.1	Contexte	172
6.6.2	Un arbre de théorie monadique indécidable	173
6.6.3	Opérateur de retrait	174
6.6.4	La construction	175
7	Conclusion et perspectives	180
A	Annexes	188
A.1	Complément sur les transducteurs déterministes	188
A.2	Complément sur le théorème 16	193
A.3	Complément sur le théorème 26	196

Chapitre 1

Introduction

Contexte

L'apparition de l'informatique a été accompagnée et a même précédé d'intenses efforts de formalisation dans l'objectif de préciser ce qu'est un «calcul». L'étude de cette question fit rapidement apparaître que cette notion était intimement reliée à celle de preuve mathématique. Depuis lors, les rapprochements entre informatique et logique mathématique n'ont fait que grandir en nombre et en importance. Citons, d'un point de vue pratique, l'apparition de langages de programmation dits logiques permettant d'exprimer l'exécution d'un programme comme la démonstration d'un énoncé logique. Citons d'un point de vue théorique, les travaux de Fagin qui donnèrent naissance à la complexité descriptive; une théorie identifiant la notion informatique de classe de complexité à celle de définissabilité dans une certaine logique.

Une autre branche dont les applications sont des plus prometteuses est l'utilisation de la logique dans le cadre de la vérification. Dans une telle approche, la logique permet d'exprimer des propriétés auxquelles doit se conformer un programme (ou tout autre objet). Ces techniques de développement peuvent être classées suivant leur niveau d'automatisme. Nous nous intéressons ici à la vérification totalement automatique de propriétés. Celle-ci est bien entendue indécidable dans la plupart des cas. Néanmoins, il est possible d'établir automatiquement des propriétés pour certains systèmes plus simples que les machines de Turing. Décider de la vacuité d'un langage rationnel à partir de la donnée d'un automate le reconnaissant peut être vu comme un cas simple de telle vérification. Pour pouvoir donner toute sa généralité à cette vérification, il convient de paramétrer l'étude par un formalisme de description de l'objet à vérifier et une syntaxe des propriétés à vérifier. C'est ce que propose l'approche logique.

Logique

Deux notions logiques fondamentales nous intéressent tout particulièrement, celle de structure et celle de formule.

Les structures (relationnelles) permettent de décrire l'objet d'étude. Dans le contexte de la vérification, la structure décrit le programme à vérifier. Plus précisément, une structure est un ensemble d'éléments munis de relations. Dans le cas d'un automate par exemple, chacun des éléments représente un état et les relations codent les transitions, les ensembles des états initiaux et finaux. Pour l'étude de comportement, chaque élément est une configuration et les relations décrivent les transitions du programme.

Les formules permettent quant à elles de décrire les propriétés que l'on cherche à établir. Ainsi, on se fixe un langage de formules, c'est-à-dire un ensemble de mots. À chaque formule logique est attribuée une sémantique qui précise pour toute structure si elle «satisfait la formule» ou non. On dit dans ce cas que la structure est un modèle de la formule. Une formule logique peut, par exemple, énoncer l'existence d'un chemin entre deux sommets d'un graphe. Dans le cadre des automates, cela permet d'exprimer la non vacuité du langage accepté et, dans le cadre de la vérification de programmes, cela peut servir à exprimer la possibilité d'atteindre une configuration non valide. Une logique est décrite par la donnée de l'ensemble des formules et par la relation de satisfaction.

Dans la pratique, de très nombreuses logiques ont été développées, variant par leur expressivité ou leur objectif. Deux principales variantes concernent la vérification.

Les logiques temporelles expriment des propriétés sur les graphes orientés (appelés dans ce contexte «structures de Kripke»). Ces graphes représentent des systèmes de transition, et une formule de la logique exprime des contraintes sur le comportement du système. Les logiques temporelles se décomposent en logiques du temps linéaire (comme LTL) pour lesquelles une exécution est une suite de configurations et les logiques du temps arborescent (comme CTL, CTL* ou le μ -calcul modal) qui peuvent faire référence simultanément à toutes les exécutions possibles. Ces logiques peuvent être enrichies par l'ajout de modalités de retour permettant à une formule d'inverser le temps et de faire référence au «passé».

La logique du premier ordre et ses variantes. Ces logiques peuvent faire référence aux éléments de l'univers (resp. aux sommets) au moyen de variables et tester si ces éléments sont en relation dans la structure.

Une grande différence par rapport aux logiques temporelles est que la logique du premier ordre ne peut exprimer que des propriétés locales, ce qui est insuffisant dans beaucoup de cas.

La logique (du second ordre) monadique est une extension particulièrement intéressante de la logique du premier ordre. Elle enrichit cette dernière par la pos-

sibilité d'utiliser des variables représentant des parties de l'univers de la structure (resp. des ensembles de sommets du graphe). Cette augmentation d'expressivité permet, entre autre, d'exprimer l'existence d'un chemin entre deux sommets. Cette propriété n'est plus locale.

Structures finies

Lorsque les informaticiens se sont penchés sur la logique, et plus précisément le problème de la vérification de modèle (le *model checking*), c'est tout naturellement sur la vérification de structures finies et de famille de structures finies qu'ils se sont d'abord attachés. Ces études ont été à l'origine de nombreux résultats et de nombreuses pistes de recherche.

En général, déterminer si une structure finie satisfait une certaine formule logique est décidable. Il suffit pour cela que la logique ne soit pas intrinsèquement trop puissante. On peut néanmoins s'intéresser aux questions d'existence d'un modèle pour une formule (le problème de satisfaisabilité), de complexité du problème de satisfaction et de la satisfaisabilité au sein d'un ensemble de structures finies.

Le problème de la satisfaisabilité consiste à déterminer si une formule logique donnée admet un modèle fini. Ce problème est décidable pour les logiques temporelles (LTL, CTL, μ -calcul modal ou μ -calcul modal avec modalité de retour). Quand il est question de logique du premier ordre, l'existence d'un modèle fini est indécidable (Traktenbrot). Naturellement, pour des logiques plus expressives comme la logique monadique, la situation n'est pas plus favorable.

Cependant, quand il s'agit de déterminer l'existence d'un modèle fini possédant certaines contraintes géométriques, le problème peut redevenir décidable. Ainsi, déterminer s'il existe un arbre fini satisfaisant une propriété exprimée en logique monadique est décidable. Ce résultat se généralise aux structures qui ne s'éloignent pas trop d'un arbre. Pour cela, on introduit la notion de largeur arborescente qui est une mesure de la ressemblance d'une structure à un arbre. Pour k fixé, déterminer l'existence d'un modèle fini de largeur arborescente inférieure à k pour une formule monadique est décidable. Une autre mesure de la ressemblance entre une structure et un arbre est la largeur de clique¹. Là encore, décider de l'existence d'un modèle de largeur de clique bornée pour une formule monadique est décidable. En fait, un résultat de Seese établit que déterminer l'existence d'un modèle pour une formule de la logique du second ordre gardée (une extension de la logique monadique) parmi une famille de graphes finis ne peut être décidable que si cette famille est de largeur arborescente bornée. Une relation similaire entre

1. Le terme de ressemblance peut sembler dans ce cas excessif car cette mesure n'a rien de «visuelle». Elle correspond néanmoins à la possibilité de décrire la structure à partir d'un arbre.

la logique monadique et les familles de structures de largeur de clique bornée est connue sous le nom de «conjecture de Seese».

Si l'on s'intéresse aux questions de complexité, là encore les notions de largeur arborescente et de largeur de clique entrent en compte. En effet, étant données une structure et une formule logique du premier ordre, déterminer si la structure est un modèle de la formule est PSPACE-complet. Il en est de même pour la logique monadique. Si la formule est fixée, alors le problème se trouve dans la hiérarchie polynomiale pour ces deux logiques². Pourtant, pour une formule fixée de la logique monadique et un entier k , déterminer si une structure de largeur arborescente inférieure à k est un modèle de la formule n'est que linéaire dans la taille de la structure, ce qui correspond à un remarquable gain en complexité. Le phénomène reste valable pour la largeur de clique.

Simultanément des travaux se sont attachés à la description de familles de structures finies. Ceci peut être fait par caractérisation logique (e.g. l'ensemble des modèles d'une formule), par utilisation de propriétés géométriques (planarité, exclusion de mineurs . . .) ou par une approche algébrique. L'approche algébrique consiste à se fixer un alphabet (éventuellement infini) d'opérations permettant de transformer et de combiner des structures. Une structure complexe peut être alors décrite comme évaluation d'une expression mettant en jeu ces opérations. Cela permet de représenter des ensembles de structures par des ensembles de termes.

Deux familles d'opérations sont usuellement considérées dans ce contexte : les opérateurs HR et les opérateurs VR (qui travaillent sur les graphes mais peuvent être étendus aux structures). Les opérateurs HR capturent exactement la notion de largeur arborescente : pour tout entier k , il existe un ensemble fini d'opérateurs HR tels qu'un graphe est de largeur arborescente inférieure à k si et seulement si il est (à isomorphisme près) évaluation d'une expression n'utilisant que ces opérateurs. Une relation identique relie les opérateurs VR et la largeur de clique (en fait, il s'agit de la définition de la largeur de clique).

Cette approche se combine avec la notion de reconnaissabilité sur les termes finis. Au moyen d'automates finis (ou de définition logique, ou d'expressions rationnelles ou de grammaires), il est possible de définir des ensembles de termes, appelés langages. Lorsque l'on évalue les termes d'un langage, cela permet de définir une famille de structures.

Structures infinies

Avec les structures (et les graphes) infinies, une nouvelle problématique apparaît : comment décrire finiment un objet infini.

Plusieurs approches sont possibles :

- la représentation interne consiste à décrire explicitement l'univers et les

2. Le niveau dans la hiérarchie correspond au nombre d'alternances de quantifications.

relations,

- la représentation équationnelle décrit une structure infinie comme la solution d'un système d'équations utilisant un ensemble d'opérateurs fixé,
- la représentation par transformations à pour point de départ une structure infinie connue, et revient à décrire de nouvelles structures comme des transformations de cette structure,
- la définissabilité logique consiste à décrire une (ou des) structure(s) comme le (ou les) modèle(s) d'une formule dans une logique donnée. Peu de résultats sont connus dans ce domaine (e.g. Blumensath).

Historiquement, les travaux de Müller et Schupp sur les graphes des automates à pile ont initié les recherches sur les familles de structures infinies. La représentation que les auteurs choisissent est interne: chaque sommet d'un graphe est la configuration d'un automate à pile (au sens classique de la théorie des langages), et ce graphe contient un arc entre deux sommets si une transition de l'automate permet de passer de la configuration du premier à celle du second. Les auteurs montrent par une analyse géométrique que ces graphes ont une théorie monadique décidable. Des résultats ultérieurs de Caucal montrent qu'il est possible de représenter ces graphes par transformations. Cette représentation donne une démonstration simplifiée du résultat de décidabilité de la théorie monadique.

Les travaux de Courcelle furent quant à eux inspirés par l'approche algébrique. Il adapte les opérateurs HR pour traiter les structures d'univers dénombrable. En particulier, il s'agit de rendre ces opérateurs continus. Les graphes HR-équationnels sont alors définis comme les solutions de systèmes finis d'équations utilisant ces opérateurs. Courcelle montre que ces structures ont une théorie au second ordre gardé décidable.

Enfin, Caucal décrit la famille des graphes préfixes reconnaissables par des systèmes de réécriture de mots et par transformation d'un graphe particulier. Une conséquence directe de cette caractérisation est la décidabilité de leur théorie monadique. Barthelmann montre que cette famille de graphes admet une représentation équationnelle au moyen des opérateurs VR, en fait, l'extension naturelle aux graphes infinis des opérateurs VR. Barthelmann montre aussi que les graphes de cette famille possédant une largeur arborescente bornée sont exactement les graphes HR-équationnels. Blumensath donne quant-à lui une représentation par définissabilité en logique du second ordre gardée de cette même famille.

Toutes ces familles partagent une représentation à base de pile. Ce n'est plus le cas des structures ci-dessous.

Les structures automatiques ont été introduites par Senizergues puis Khousainov et Nerode. Elles sont décrites au moyen d'une représentation interne. Chaque élément de l'univers est un mot, et les relations associées aux symboles sont décrites par des relations synchronisées. Ces structures possèdent une théorie au premier ordre décidable.

Elles sont étendues aux structures termes-automatiques par Blumensath et Grädel. Des définitions antérieures semblables peuvent être trouvées dans les travaux de Dauchet et Tison. Nous verrons dans ce travail qu'il est possible de donner une représentation équationnelle à cette famille de structures. Toutes ces familles possèdent une théorie au premier ordre décidable.

Enfin, les familles de graphes décrites par réécriture suffixe de termes sont introduites par Löding. Là encore, des familles similaires antérieures ont été étudiées par Dauchet et Tison. Löding montre que, quand restreints aux graphes de largeur arborescente bornée, ces graphes coïncident avec les graphes des automates à piles de Müller et Schupp.

Contributions

Le présent travail étudie la plupart des familles de structures précédemment décrites sous l'angle des systèmes d'équations.

Pour cette raison, la première adaptation concerne les transducteurs déterministes avec anticipations; il s'agit d'un outil permettant de transformer les systèmes d'équations. Ces transducteurs ont été étudiés en détail dans le cadre des termes finis, et nous adaptons cet outil au cas des termes infinis. Ces travaux recouvrent partiellement le contenu d'une collaboration entre C. Löding et l'auteur [CL04]. Il s'agit de l'outil essentiel utilisé dans le reste du mémoire.

La seconde contribution (chapitre 4) concerne plusieurs équivalences généralisant des résultats connus sur les graphes préfixes reconnaissables; en particulier, la possibilité d'ajouter un opérateur de fusion dans les systèmes d'équations VR. Il s'agit d'une collaboration entre A. Carayol et l'auteur [CC03]. Nous redémontrons aussi le résultat de Barthelmann concernant les graphes préfixes reconnaissables de largeur arborescente bornée en utilisant les transducteurs déterministes.

La troisième contribution n'a pas fait l'objet de publication; il s'agit d'une représentation équationnelle de la famille des graphes dits terme-automatiques. Elle est au cœur du chapitre 5.

La quatrième et dernière contribution concerne l'introduction de la famille des graphes VRA-équationnels (chapitre 6), des différentes représentations qui peuvent en être données ainsi que de ses propriétés géométriques. Cette famille admet une représentation interne sous forme de système de réécriture suffixe de termes (elle est donc similaire à la famille de graphes introduits par Löding, bien que les deux soient incomparables). Nous montrons ici que cette famille restreinte aux graphes de largeur arborescente bornée coïncide avec la famille des graphes HR-équationnels. Ces résultats ont été partiellement publiés dans [Col02a]. Enfin, nous établissons qu'une extension de ce résultat aux graphes de largeur de clique bornée n'est pas vérifiée. Ce dernier résultat est aussi issu d'une collaboration entre C. Löding et l'auteur.

L'auteur s'est aussi intéressé au cours de sa thèse aux systèmes de réécriture de termes modulo associativité et commutativité [Col02b], ainsi qu'au sujet plus éloigné de la programmation par aspects en collaboration avec P. Fradet [CF00]. Ces travaux ne sont pas présentés dans ce document.

Plan

Le reste de ce mémoire est découpé en cinq chapitres.

Le chapitre 2 présente les définitions de base concernant les structures, les graphes, la logique et les propriétés géométriques de graphes.

Le chapitre 3 est dédié à l'étude de systèmes d'équations. On s'intéresse tout particulièrement au développement d'outils permettant de manipuler ces systèmes : les transducteurs déterministes.

Le chapitre 4 commence l'étude des structures et des graphes infinis proprement dite. Il a trait aux structures pouvant être décrites par réécriture suffixe (ou préfixe) de mots; chaque élément de l'univers est un mot, et une relation relie plusieurs mots si ceux-ci s'obtiennent les uns des autres par remplacement suffixe (resp. préfixe). Ces familles de structures admettent des représentations internes, équationnelles et par transformations, et nous établissons ces équivalences. Certains résultats présentés sont déjà connus dans le contexte des structures finies et des systèmes d'équations finis. Le principal apport consiste en une utilisation systématique des transducteurs déterministes permettant d'étendre ces résultats aux systèmes d'équations infinis.

Le chapitre 5 étudie une famille de structures dont l'univers est un ensemble de termes finis, et dont les relations sont décrites au moyens d'automates. Il s'agit des structures terme-automatiques. Nous donnons, là encore, une triple équivalence de représentations pour ces structures.

Le chapitre 6 s'intéresse aux graphes dont les sommets sont des termes, et dont les arcs sont décrits par réécriture suffixe, c'est-à-dire en remplaçant un sous-terme par un autre. Nous établissons l'équivalence entre une représentation équationnelle et une représentation interne de ces graphes, et nous nous intéressons à une étude géométrique de cette famille.

Chapitre 2

Définitions et notions de base

La notion de structure relationnelle est issue de la logique et en particulier de la théorie des modèles. Une structure est formée d'un ensemble d'éléments, son univers, et de relations liant ces éléments.

Les structures relationnelles peuvent donc être vues comme des sortes de bases de données relationnelles. La différence est que les structures relationnelles ne sont pas étudiées sous l'angle du stockage d'information, mais plutôt dans l'optique d'étudier les interactions entre relations. Dans le cas présent, la vision base de données disparaît totalement puisque l'on s'intéresse à des structures infinies.

Les structures relationnelles peuvent aussi être vues comme une généralisation de la notion de graphe : l'ensemble des sommets est l'univers et les arcs sont représentés par des relations binaires. Sous cet angle, il est souvent fait référence à la notion d'*hypergraphe*. On parle alors d'hyperarcs reliant n sommets au lieu d'arcs (qui n'en relient que deux).

Dans la suite de ce travail, il est question de structures plutôt que d'hypergraphes bien que certaines terminologies seront empruntées à ces derniers. Il s'agit d'un choix arbitraire qui dénote une orientation plus logique qu'algorithmique des résultats présentés. En revanche, la plupart des exemples seront puisés dans les graphes ; ces derniers possèdent l'avantage d'être plus aisément représentables.

Le reste de ce chapitre est décomposé comme suit. Nous commençons par introduire les définitions de structures, graphes et termes au paragraphe 2.1. Puis au paragraphe 2.2 nous présentons les formalismes logiques qui nous seront utiles par la suite. Nous introduirons alors diverses transformations de structures et de graphes au paragraphe 2.3. Enfin, le paragraphe 2.4 a pour but de présenter des critères géométriques permettant d'étudier les graphes infinis.

2.1 Structures

Nous présentons dans ce paragraphe la notion de structure et la terminologie s'y rattachant ainsi que les variantes que sont les graphes, les automates, les arbres et les termes.

2.1.1 Cas général

Une *signature* est un ensemble fini Σ de *symboles relationnels*, chaque symbole R possédant une *arité* entière notée $|R|$. Une *structure relationnelle* (ou plus simplement une structure) \mathcal{S} sur une signature Σ est une paire $(\mathcal{U}, \text{Vals})$ où \mathcal{U} est un ensemble appelé *univers*, et Vals est une application associant à chaque symbole R de la signature son *interprétation*, c'est à dire une partie de $\mathcal{U}^{|R|}$. Nous emploierons dorénavant la notation classique $\mathcal{U}_{\mathcal{S}}$ pour représenter l'univers de la structure \mathcal{S} , et $R^{\mathcal{S}}$ en lieu et place de $\text{Vals}(R)$. De plus, pour $u_1, \dots, u_{|R|}$ éléments de l'univers, on note $R^{\mathcal{S}}(u_1, \dots, u_{|R|})$ au lieu de $(u_1, \dots, u_{|R|}) \in R^{\mathcal{S}}$. On dit dans ce cas qu'un *hyperarc* R relie les éléments $u_1, \dots, u_{|R|}$.

La notion de signature est pratique dans les définitions mais devient rapidement gênante à l'usage. En particulier, il est fréquent de vouloir que les signatures de différentes structures soient égales. Cela permet par exemple de comparer ou de combiner des structures. On définit donc l'extension d'une structure à une nouvelle signature (plus grande) en fixant pour tous les nouveaux symboles une interprétation vide. On suppose implicitement qu'une telle extension est effectuée quand nécessaire. Ainsi, étant données deux structures, il est toujours possible de les considérer comme de même signature (en les étendant toutes deux à l'union de leur signature). Une conséquence de ce choix est que lorsque l'on parle d'un symbole sans préciser la signature dont il est issu, cela signifie qu'il appartient à une signature étendant les signatures de toutes les structures en présence. Un autre abus du même ordre consiste à considérer des structures sur une signature infinie. En réalité, il s'agit de structures de signature finie contenue dans la signature infinie.

Une *structure avec quotient* \mathcal{S} est une structure sur une signature contenant un symbole binaire \sim (de notation infixé) et telle que l'interprétation de \sim est une congruence pour toutes les relations de la structure. Plus précisément, $\sim^{\mathcal{S}}$ est une relation d'équivalence sur $\mathcal{U}_{\mathcal{S}}$ et pour tout symbole relationnel R d'arité n et tous éléments $u_1 \sim^{\mathcal{S}} v_1, \dots, u_n \sim^{\mathcal{S}} v_n$, $R^{\mathcal{S}}(u_1, \dots, u_n)$ si et seulement si $R^{\mathcal{S}}(v_1, \dots, v_n)$.

Un *isomorphisme* de structures avec quotient entre \mathcal{S} et \mathcal{S}' est une relation $\Upsilon \subseteq \mathcal{U}_{\mathcal{S}} \times \mathcal{U}_{\mathcal{S}'}$ telle que pour tout symbole relationnel R d'arité n (éventuellement \sim) et tous $E(u_1, v_1), \dots, E(u_n, v_n)$, $R^{\mathcal{S}}(u_1, \dots, u_n)$ si et seulement si $R^{\mathcal{S}'}(v_1, \dots, v_n)$. On note alors $\mathcal{S} \approx_{\Upsilon} \mathcal{S}'$ et on dit que \mathcal{S} et \mathcal{S}' sont isomorphes si il existe un tel isomorphisme. La notation $\mathcal{S} \approx \mathcal{S}'$ est utilisée quand l'isomorphisme

n'a pas d'importance. Toute structure est isomorphe à elle-même, et l'isomorphisme correspondant est simplement l'interprétation de \sim . Plus généralement la relation \approx est une relation d'équivalence entre structures avec quotient.

Étant donnée une structure avec quotient \mathcal{S} , il lui est associée sa *structure quotient* \mathcal{S}/\sim dont l'univers est l'ensemble des classes d'équivalences de $\sim^{\mathcal{S}}$, et dont l'interprétation d'un symbole R d'arité n est l'ensemble des (c_1, \dots, c_n) tels qu'il existe $u_1 \in c_1, \dots, u_n \in c_n$ vérifiant $R^{\mathcal{S}}(u_1, \dots, u_n)$. La définition d'une structure avec quotient est telle qu'il est équivalent de définir l'interprétation de R comme l'ensemble des (c_1, \dots, c_n) tels que pour tout $u_1 \in c_1, \dots, u_n \in c_n$ on a $R^{\mathcal{S}}(u_1, \dots, u_n)$. La relation qui à chaque élément associe sa classe d'équivalence est alors un isomorphisme entre \mathcal{S} et \mathcal{S}/\sim .

Les structures sans quotient s'identifient au cas particulier des structures avec quotient pour lesquelles l'interprétation de \sim est simplement l'identité. Cette équivalence est implicite dans la suite de ce travail et le choix de la présence d'un quotient dépend du contexte. Remarquons que pour toute structure avec quotient, sa structure quotient est en ce sens une structure sans quotient. Pour finir, notons que dans le cas des structures sans quotient, un isomorphisme de structure est bien une bijection entre univers comme on peut s'y attendre.

2.1.2 Graphes

Les graphes sont définis dans ce travail comme des cas particuliers de structures. Étant donné un alphabet A d'étiquettes a, b, \dots , un *graphe* \mathcal{G} est une structure de signature $\{\overset{a}{\rightarrow} \mid a \in A\}$, où chaque symbole $\overset{a}{\rightarrow}$ est binaire. L'univers d'un graphe est noté $V_{\mathcal{G}}$ et ses éléments se nomment *sommets*. La notation $(\overset{a}{\rightarrow})^{\mathcal{G}}(s, s')$ est remplacée par $s \overset{a}{\xrightarrow{\mathcal{G}}} s'$ et on dit qu'il y a un *arc* étiqueté par a entre s et s' . Un arc d'un sommet vers lui-même est appelé une *boucle*.

Un graphe \mathcal{G} est *déterministe* si pour tout sommet s et toute étiquette a , il existe au plus un sommet s' tel que $s \overset{a}{\xrightarrow{\mathcal{G}}} s'$. Un graphe est *co-déterministe* si le graphe obtenu en retournant tous les arcs est déterministe, c'est à dire que pour tout sommet s et toute étiquette a , il existe au plus un s' tel que $s' \overset{a}{\xrightarrow{\mathcal{G}}} s$. Le *degré sortant* $d^+(s)$ d'un sommet s dans un graphe \mathcal{G} est le nombre d'arcs qui en sont issus. Il peut avoir une valeur dans \mathbb{N} ou ∞ s'il existe une infinité de tels arcs. Formellement, $d^+(s)$ est le cardinal de $\{(a, s') \in A \times V_{\mathcal{G}} \mid s \overset{a}{\xrightarrow{\mathcal{G}}} s'\}$. Le *degré entrant* $d^-(s)$ se définit similairement comme le cardinal de $\{(s', a) \in V_{\mathcal{G}} \times A \mid s' \overset{a}{\xrightarrow{\mathcal{G}}} s\}$. Le *degré* $d(s)$ d'un sommet s est la somme de ses degrés entrants et sortants.

Un *chemin* dans un graphe est un mot $\pi = s_0 a_1 s_1 \dots a_n s_n$ de $V_{\mathcal{G}}(AV_{\mathcal{G}})^*$ tel que pour tout i entre 1 et n , $s_{i-1} \overset{a_i}{\xrightarrow{\mathcal{G}}} s_i$. Le sommet s_0 est l'*origine* du chemin, noté $orig(\pi)$, et s_n sa *destination* notée $dest(\pi)$. On s'intéresse aussi aux étiquettes

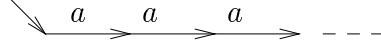


FIG. 2.1 – La demi-droite des entiers enracinée Δ_1 .

apparaissant sur un chemin, c'est à dire au mot $a_1 a_2 \dots a_n$. Pour un mot w de A^* , la notation $s \xrightarrow[\mathcal{G}]{w} s'$ désigne l'existence d'un chemin étiqueté par w entre s et s' . Étant donné un graphe \mathcal{G} et deux ensembles de sommets $I \subseteq V_{\mathcal{G}}$ et $F \subseteq V_{\mathcal{G}}$, on appelle *trace* de \mathcal{G} entre I et F le langage $\mathcal{L}_{I,F}(\mathcal{G})$ défini par

$$\mathcal{L}_{I,F}(\mathcal{G}) = \{w \mid \exists s \in I, \exists s' \in F, s \xrightarrow[\mathcal{G}]{w} s'\} .$$

Dans le cas d'un graphe fini \mathcal{G} , pour tout I et tout F , $\mathcal{L}_{I,F}(\mathcal{G})$ est un langage rationnel. Ce n'est plus nécessairement le cas si le graphe est infini.

Un graphe \mathcal{G} est *enraciné* si en plus de sa signature normale de graphe, il possède un symbole unaire *racine* interprété comme un singleton. Par abus de notation, on identifie $racine^{\mathcal{G}}$ avec son seul élément et on appelle cet élément la *racine* du graphe. Graphiquement, ce sommet est identifié par une flèche entrante. Un sommet d'un graphe enraciné est *accessible* si il existe un chemin d'origine *racine* dont il est la destination.

Exemple 2.1 La figure 2.1 présente la demi-droite des entiers enracinée Δ_1 , c'est-à-dire le graphe enraciné de sommets $V = \mathbb{N}$, de racine 0, et dont les arcs étiquetés par a sont :

$$\xrightarrow[\Delta_1]{a} = \{(n, n+1) \mid n \in \mathbb{N}\} .$$

Ce graphe est déterministe et co-déterministe. Il existe un chemin entre deux sommets n et n' si et seulement si $n \leq n'$ et ce chemin est étiqueté par le mot $a^{n'-n}$.

2.1.3 Les automates

Un automate est un graphe servant de description pour un langage.

Formellement, un *automate* \mathcal{A} est un graphe enraciné dont la signature est enrichie d'un symbole unaire *final*. Les sommets s tels que $final^{\mathcal{A}}(s)$ sont appelés sommets *finals* et sont graphiquement représentés par une flèche sortante. Le langage accepté par un automate d'étiquettes A est le langage $\mathcal{L}(\mathcal{A})$ de A^* défini par :

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}_{racine^{\mathcal{A}}, final^{\mathcal{A}}}(\mathcal{A}) = \{w \mid \exists s \in V_{\mathcal{A}}. racine^{\mathcal{A}} \xrightarrow[\mathcal{A}]{w} s, final^{\mathcal{A}}(s)\} .$$

Exemple 2.2 La figure 2.2 représente un automate infini, appelé l'escalier. Le langage qu'il accepte est $\{a^n b^{n+1} c^n \mid n \in \mathbb{N}\}$.

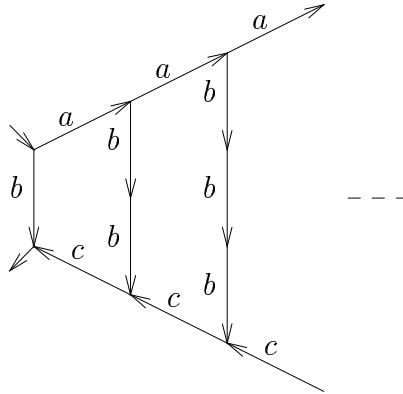


FIG. 2.2 – Un automate.

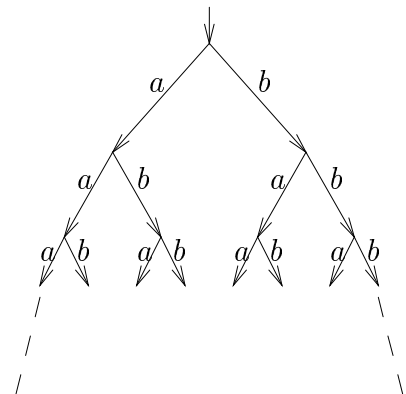
2.1.4 Arbres et termes

$$V_{\Delta_2} = \{a, b\}^*$$

$$\xrightarrow[\Delta_2]{a} = \{(u, ua) \mid u \in V\}$$

$$\xrightarrow[\Delta_2]{b} = \{(u, ub) \mid u \in V\}$$

(a)



(b)

FIG. 2.3 – L'arbre binaire complet, définition (a) et représentation (b).

Les *arbres* sont des cas particuliers de graphes enracinés satisfaisant de plus que tout sommet est destination d'un et un seul chemin à partir de la racine. Les sommets d'un arbre sont appelés *nœuds*. Si pour une certaine étiquette a , deux nœuds s et s' satisfont $s \xrightarrow{a} s'$ alors on dit que s est le *père* de s' (il ne peut en exister qu'un d'après la définition d'un arbre) et que s' est un *fil*s de s . Si un nœud n'a pas de fils, alors il s'agit d'une *feuille* de l'arbre.

Exemple 2.3 Un exemple important d'arbre infini est l'arbre binaire complet, noté Δ_2 et présenté à la figure 2.3.

On appelle *branche* d'un arbre, un ensemble maximal (pour l'inclusion) de

nœuds contenant la racine et tel que tout élément de l'ensemble ne possède au plus qu'un fils dans l'ensemble. Le langage des branches d'un arbre t étiqueté sur A , noté $\mathcal{LB}(t)$ est l'ensemble

$$\mathcal{LB}(t) = \left\{ w \in A^* \mid \exists u \in V_t, \text{ racine} \xrightarrow[t]{w} u \right\} .$$

Les branches d'un arbre apportent une information importante sur celui-ci. En fait, sous l'hypothèse de déterminisme, ce langage s'identifie à l'ensemble des nœuds de l'arbre comme le montre la propriété suivante.

Propriété 2.4 *Tout arbre déterministe t est isomorphe au graphe enraciné de sommets $\mathcal{LB}(t)$, de racine ε et dont les arcs étiquetés par a sont définis par :*

$$\xrightarrow{a} = \{(w, wa) \mid w \in \mathcal{LB}(t), wa \in \mathcal{LB}(t)\} .$$

Corollaire 2.5 *Pour tout ensemble d'étiquettes A , l'application \mathcal{LB} est une bijection entre les arbres déterministes étiquetés par A et les langages de A^* clos par préfixe.*

Cette propriété permet de manipuler les arbres déterministes comme des langages et de manipuler les nœuds comme des mots de ce langage.

Il s'agit maintenant de définir ce qu'est un terme. Pour cela, il est nécessaire de disposer de symboles fonctionnels. Un *symbole fonctionnel* f est un symbole accompagné d'une *arité* notée $|f|$ qui est un entier naturel. On note souvent lors de l'introduction de ces symboles $f : n$ pour préciser que l'arité de f est n . Les symboles fonctionnels ne doivent pas être confondus avec les symboles relationnels qui eux sont notés en lettres capitales. Un *alphabet gradué* est un ensemble fini de symboles fonctionnels. Les définitions qui suivent permettent de donner un sens à un terme comme $f(g(a), b)$ sur l'alphabet gradué $\{f : 2, g : 1, a : 0, b : 0\}$.

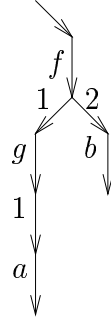
Un *terme* sur un alphabet gradué \mathcal{F} (ou plus simplement un \mathcal{F} -terme) est un arbre t étiqueté par $\mathcal{F} \uplus \mathbb{N}$ et satisfaisant les trois conditions suivantes :

- il existe un arc étiqueté par une lettre de \mathcal{F} issu de la racine de l'arbre,
- pour tout arc $s \xrightarrow[t]{f} s'$ avec $f \in \mathcal{F}$, alors, il s'agit du seul arc d'origine s , et il existe exactement $|f|$ arcs issus de s' étiquetés respectivement par $1, \dots, |f|$,
- pour tout arc $s \xrightarrow[t]{n} s'$ avec $n \in \mathbb{N}$, il existe un arc étiqueté par une lettre de \mathcal{F} issu de s' .

L'ensemble des termes, finis ou infinis, sur l'alphabet \mathcal{F} est noté $\mathcal{T}^\omega(\mathcal{F})$. L'ensemble des termes finis sur \mathcal{F} est noté $\mathcal{T}(\mathcal{F})$.

Dans cette définition des termes, deux types de sommets apparaissent : ceux origine des lettres de \mathcal{F} et ceux destination de tels arcs. On appelle les premiers *sommets principaux*. Étant donné un terme t et v un de ses sommets principaux,

t^v représente le graphe restreint aux sommets accessibles à partir de v et dont la racine est v . Il s'agit d'un terme que l'on appelle *sous-terme* à la position v . On le note t^v . À titre d'exemple, le graphe suivant



est un terme fini sur $\{f : 2, g : 1, a : 0, b : 0\}$. On remarque d'après la définition des termes, que ceux-ci sont des arbres déterministes. D'après la propriété 2.5, ils peuvent donc être identifiés avec leur langage de branches. Par exemple

$$\mathcal{LB}(f(g(a), b)) = \{\varepsilon, f, f1, f1g, f1g1, f1g1a, f2, f2b\}.$$

Il s'agit maintenant de formaliser comment ce terme peut être identifié à $f(g(a), b)$. Soit f un symbole fonctionnel d'arité n et n termes t_1, \dots, t_n , on définit le terme $t = f(t_1, \dots, t_n)$ par :

- l'ensemble de nœuds est $V = \{(i, u) \mid u \in V_{t_i}\} \cup \{0, 1\}$,
- la racine est 0,
- pour tout symbole fonctionnel $g \neq f$, $\frac{g}{t} = \left\{ ((i, u), (i, v)) \mid u \xrightarrow{t_i} v \right\}$,
- $\frac{f}{t} = \left\{ ((i, u), (i, v)) \mid u \xrightarrow{t_i} v \right\} \cup \{(0, 1)\}$,
- pour tout $k > n$, $\frac{k}{t} = \left\{ ((i, u), (i, v)) \mid u \xrightarrow{t_i} v \right\}$,
- pour tout $1 \leq k \leq n$, $\frac{k}{t} = \left\{ ((i, u), (i, v)) \mid u \xrightarrow{t_i} v \right\} \cup \{(1, (k, \text{racine}^{t_k}))\}$.

Ainsi, le terme précédent satisfait l'équation suivante :

$$\begin{array}{c} \text{Tree Diagram} \end{array} = f \left(\begin{array}{c} \text{Tree Diagram } g(a) \end{array}, \begin{array}{c} \text{Tree Diagram } b \end{array} \right) = f(g(a), b).$$

En utilisant la représentation par langage de branches des termes, on obtient une définition équivalente beaucoup plus simple :

$$\mathcal{LB}(f(t_1, \dots, t_{|f|})) = \varepsilon + f + f1\mathcal{LB}(t_1) + \dots + f|f|\mathcal{LB}(t_{|f|}) .$$

Cette construction permet d'associer un terme fini à toute expression bien formée à partir d'un alphabet gradué. En fait, la réciproque est aussi vraie : à tout terme fini sur un alphabet gradué correspond une unique expression bien formée sur cet alphabet. Pour cette raison, on dit que l'algèbre des termes finis est *libre*. Cette propriété sera présentée plus en détail au paragraphe 3.1.3.

2.2 Logiques

Les logiques permettent d'exprimer des propriétés sur les structures. De nombreux formalismes existent permettant de disposer d'une grande diversité d'expressivités. Dans ce travail, nous nous concentrons sur la logique du premier ordre (paragraphe 2.2.1) sur la logique du second ordre monadique (paragraphe 2.2.2) et quelques-unes de ses variantes (paragraphe 2.2.3). Enfin, nous présenterons la construction de Feferman-Vaught qui permet la décomposition de formules logiques (paragraphe 2.2.4).

2.2.1 Logique du premier ordre

On se fixe un ensemble dénombrable \mathcal{V} de *variables* du *premier ordre* notées $x, y \dots$. Étant donnée une signature Σ , les *formules du premier ordre* sur Σ (on omet le plus souvent de préciser la signature Σ quand il n'y a pas d'ambiguïtés.) sont définies récursivement au moyen des constructions ci-dessous :

la constante vraie : vrai ,

la conjonction : $\phi \wedge \psi$ où ϕ et ψ sont des formules du premier ordre,

la négation : $\neg\phi$ où ϕ est une formule,

la quantification existentielle du premier ordre : $\exists x.\phi$ où x est une variable du premier ordre et ϕ une formule du premier ordre,

l'égalité de variables du premier ordre : $x = y$ avec x et y variables du premier ordre,

la relation entre variables du premier ordre : $R(x_1, \dots, x_{|R|})$ où R est un symbole relationnel de Σ et $x_1, \dots, x_{|R|}$ sont des variables du premier ordre.

Les autres connecteurs classiques ne sont que des combinaisons des précédents :

la constante fausse : faux équivaut à \neg vrai ,

la disjonction : $\phi \vee \psi$ équivaut à $\neg((\neg\phi) \wedge (\neg\psi))$,

l'implication : $\phi \Rightarrow \psi$ équivaut à $(\neg\phi) \vee \psi$,

l'équivalence : $\phi \Leftrightarrow \psi$ équivaut à $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$,

la quantification universelle du premier ordre : $\forall x.\phi$ équivaut à $\neg(\exists x.\neg\phi)$.

Étant donnée une formule ϕ , on définit l'ensemble $FV(\phi)$ des *variables libres* par récurrence sur la structure de ϕ comme suit :

$$\begin{array}{llll} FV(\mathbf{vrai}) & = & \emptyset & FV(\phi \wedge \psi) & = & FV(\phi) \cup FV(\psi) \\ FV(\neg\phi) & = & FV(\phi) & FV(\exists x.\phi) & = & FV(\phi) - \{x\} \\ FV(x = y) & = & \{x, y\} & FV(R(x_1, \dots, x_{|R|})) & = & \{x_1, \dots, x_{|R|}\} . \end{array}$$

Une formule est *close* si elle n'a aucune variable libre. Par abus de langage et pour un ensemble $V \subseteq \mathcal{V}$ de noms de variables, on dit qu'une formule ϕ a pour variables libres V si V contient $FV(\phi)$.

Étant donnée une structure \mathcal{S} , une formule du premier ordre ϕ de variables libres V et une application γ de V dans $\mathcal{U}_{\mathcal{S}}$ appelée *valuation*, on définit par récurrence sur ϕ la relation de satisfaction de ϕ par \mathcal{S} dans le contexte γ , notée $\gamma, \mathcal{S} \models \phi$ comme suit.

$$\begin{array}{ll} \gamma, \mathcal{S} \models \mathbf{vrai} , & \\ \gamma, \mathcal{S} \models \phi \wedge \psi & \text{si } \gamma, \mathcal{S} \models \phi \text{ et } \gamma, \mathcal{S} \models \psi , \\ \gamma, \mathcal{S} \models \neg\phi & \text{si } \gamma, \mathcal{S} \not\models \phi , \\ \gamma, \mathcal{S} \models R(x_1, \dots, x_{|R|}) & \text{si } R^{\mathcal{S}}(\gamma(x_1), \dots, \gamma(x_{|R|})) , \\ \gamma, \mathcal{S} \models x = y & \text{si } \gamma(x) = \gamma(y) , \\ \gamma, \mathcal{S} \models \exists x.\phi & \text{si il existe } u \in \mathcal{U}_{\mathcal{S}} \text{ tel que } \gamma, x \rightarrow u, \mathcal{S} \models \phi \\ & \text{où } \gamma, x \rightarrow u \text{ représente la fonction partout égale} \\ & \text{à } \gamma \text{ sauf pour } x \text{ où elle vaut } u . \end{array}$$

Pour ϕ une formule close, si $\emptyset, \mathcal{S} \models \phi$ (\emptyset représente la seule application de domaine \emptyset), on dit que \mathcal{S} est un *modèle* de ϕ ou plus simplement que \mathcal{S} *satisfait* ϕ . On le note simplement $\mathcal{S} \models \phi$. Une structure \mathcal{S} possède une *théorie au premier ordre* décidable s'il existe un algorithme prenant en entrée une formule close ϕ et décide de sa satisfaction par \mathcal{S} .

Il est parfois utile de rendre explicite les variables libres d'une formule. Ainsi, on notera $\phi(x_1, \dots, x_n)$ une formule logique de variables libres x_1, \dots, x_n . De plus, on s'autorisera de noter $\mathcal{S} \models \phi(u_1, \dots, u_n)$ pour une structure \mathcal{S} et n éléments de son univers u_1, \dots, u_n à la place de $\gamma, \mathcal{S} \models \phi(x_1, \dots, x_n)$ avec pour tout $i \in [n]$, $\gamma(x_i) = u_i$.

Exemple 2.6 À titre d'exemple, nous donnons une formule du premier ordre qui exprime le fait, pour un graphe, d'être déterministe, c'est à dire que tout graphe \mathcal{G} étiqueté par A est déterministe si et seulement si $\mathcal{G} \models \mathbf{deterministe}[A]$ avec

$$\mathbf{deterministe}[A] = \bigwedge_{a \in A} \forall x. \forall y. \forall z. \left((x \xrightarrow{a} y \wedge x \xrightarrow{a} z) \Rightarrow y = z \right) .$$

On remarque sur cet exemple que l'on s'autorise quelques raccourcis de notation, en particulier l'usage de \wedge n'a de sens que parce que \wedge est associatif et commutatif. Il est donc possible de laisser non spécifié l'ordre sur A ainsi que le parenthésage.

Différents sous-ensembles de la logique du premier ordre jouent un rôle particulier dans le reste de ce travail. En voici la liste :

- les *expressions booléennes* (ou *formules booléennes*) sont des formules sans quantificateur, elles sont formées des connecteurs \wedge , \vee et \neg , des constantes vrai et faux, et des prédicats de relations $R(x_1, \dots, x_{|R|})$,
- les *expressions booléennes positives* (ou *formules booléennes positives*) sont des expressions booléennes sans négation. Elles sont donc formées uniquement des connecteurs \wedge et \vee , des constantes vrai et faux, et des prédicats de relations $R(x_1, \dots, x_{|R|})$,
- les *formules existentielles* sont des formules sans quantification universelle et sans négation. Elles sont donc formées des connecteurs \wedge et \vee , des constantes vrai et faux, des prédicats de relations $R(x_1, \dots, x_{|R|})$ et de la quantification existentielle \exists .

2.2.2 Logique monadique

La logique monadique étend la logique du premier ordre par des variables interprétées comme des parties de l'univers. Il est possible de quantifier existentiellement et universellement sur ces ensembles. Cela correspond à une importante augmentation d'expressivité par rapport à la logique du premier ordre. La terminologie de logique monadique est en fait une simplification remplaçant l'expression «logique au second ordre monadique» qui signifie que l'utilisation du second ordre est restreinte au cas monadique, c'est à dire aux ensembles. Cette logique pourrait aussi être qualifiée de «premier ordre ensembliste», une dénomination rendant sa nature plus explicite.

On se fixe un ensemble dénombrable \mathcal{V}_m de *variables du second ordre monadique* (ou plus simplement de *variables monadiques*). Elles sont notées en lettres capitales: $X, Y \dots$. Les formules de logique monadique (ou plus simplement *formules monadiques*) sont les formules du premier ordre étendues par la quantification existentielle du second ordre monadique $\exists X.\Phi$ et la proposition atomique d'appartenance $x \in X$. Comme précédemment, on s'autorise à utiliser la quantification universelle du second ordre monadique $\forall X.\Phi$ comme un raccourci équivalent à $\neg\exists X.\neg\Phi$.

Les définitions du paragraphe précédent sont réutilisables sans grandes modifications. Ainsi, les variables libres peuvent être monadiques tout autant que du premier ordre. La relation $\gamma, \mathcal{S} \models \Phi$ est quant à elle étendue de la manière

suivante : le contexte γ , en plus de toujours associer à chaque variable libre du premier ordre un élément de l'univers, associe à chaque variable monadique un sous-ensemble de l'univers. La définition de la satisfaction est alors complétée par les règles suivantes :

$$\begin{array}{ll} \gamma, \mathcal{S} \models \exists X. \Phi & \text{s'il existe } U \subseteq \mathcal{U}_{\mathcal{S}} \text{ tel que } \gamma, X \rightarrow U, \mathcal{S} \models \Phi, \\ \gamma, \mathcal{S} \models x \in X & \text{si } \gamma(x) \in \gamma(X). \end{array}$$

Exemple 2.7 À titre d'exemple, nous présentons une formule de la logique monadique dont la satisfaction équivaut à la forte connexité d'un graphe étiqueté par A . Cette formule exprime le fait que tout ensemble non vide de sommets clos par les relations arc contient tous les sommets du graphe.

$$\text{f-connexite}[A] = \forall X. \left(X \neq \emptyset \wedge \forall x \in X. \forall y. \bigwedge_{a \in A} (x \xrightarrow{a} y \Rightarrow y \in X) \right) \Rightarrow (\forall x. x \in X) .$$

Là encore, il y a un abus de notation : l'usage de \emptyset . Plus généralement, on s'autorise à utiliser les opérateurs ensemblistes (union, intersection, complémentaire, vide) et les comparaisons (égalité, inclusion). Bien entendu, toutes ces opérations peuvent être traduites en formules monadiques.

L'exemple suivant généralise cette technique et montre qu'il est possible d'exprimer en logique monadique le fait pour deux sommets d'être reliés par un chemin étiqueté par un mot appartenant à un langage rationnel donné.

Exemple 2.8 (accessibilité rationnelle) Pour L un langage rationnel, la formule de logique monadique $\text{accessible}[L](x, y)$ est définie par récurrence sur la définition de L :

- si $L = \emptyset$ alors $\text{accessible}[\emptyset](x, y) = \text{faux}$ car aucun mot n'appartient à L ,
- si $L = \{\varepsilon\}$ alors $\text{accessible}[\varepsilon](x, y) = (x = y)$ car le seul mot de L est le mot vide, et qu'il n'existe un chemin étiqueté par le mot vide entre deux sommets que si ceux-ci sont égaux,
- si $L = a$ alors $\text{accessible}[a](x, y) = x \xrightarrow{a} y$ car il existe un chemin étiqueté par a entre deux sommets ssi ceux-ci sont reliés par un arc étiqueté par a ,
- si $L = L' + L''$ alors

$$\begin{aligned} \text{accessibilite}[L' + L''](x, y) = \\ \text{accessibilite}[L'](x, y) \vee \text{accessibilite}[L''](x, y) . \end{aligned}$$

En effet, il existe un chemin étiqueté par un mot de L entre x et y s'il existe un mot de L' étiquetant un chemin reliant ces deux sommets, ou bien s'il existe un mot de L'' étiquetant un chemin reliant ces deux sommets,

si $L = L'.L''$ alors similairement,

$$\begin{aligned} \text{accessibilite}[L'.L''](x, y) = \\ \exists z. \text{accessibilite}[L'](x, z) \wedge \text{accessibilite}[L''](z, y), \end{aligned}$$

enfin si $L = L'^*$ alors on pose

$$\begin{aligned} \text{accessibilite}[L'^*](x, y) = \\ \forall X. \left(\begin{array}{l} x \in X \\ \wedge \forall z. \forall z'. (z \in X \wedge \text{accessibilite}[L'](z, z')) \\ \Rightarrow z' \in X \end{array} \right) \\ \Rightarrow y \in X. \end{aligned}$$

Cette formule exprime le fait que tout ensemble X de sommets du graphe close par la relation $\text{accessibilite}[L']$ et contenant x doit aussi contenir y .

Deux résultats fondamentaux de l'informatique théorique ont ouvert la voie à l'étude de la théorie monadique sur les structures infinies. Il s'agit tout d'abord du résultat de Büchi [Büc60], puis de son extension par Rabin [Rab69].

Théorème 1 (Büchi) *La demi-droite des entiers a une théorie monadique décidable.*

Théorème 2 (Rabin) *L'arbre binaire complet a une théorie monadique décidable.*

Ces deux démonstrations sont complexes. Une preuve détaillée de ces résultats ainsi que des références sur le sujet se trouvent dans les travaux de Thomas [Tho97]. Des résultats similaires seront aussi énoncés dans le contexte du μ -calcul (voir théorème 9).

D'autres techniques permettent de montrer l'indécidabilité de la théorie monadique sur certaines structures. Citons deux résultats classiques de cet ordre. Le premier fait référence à la *grille infinie*, qui se définit comme le graphe de sommets $\mathbb{N} \times \mathbb{N}$ et tel que les arcs (non étiquetés) ont pour interprétation $\{((i, j), (i, j + 1)) \mid i, j \in \mathbb{N}\} \cup \{((i, j), (i + 1, j)) \mid i, j \in \mathbb{N}\}$.

Propriété 2.9 *La grille infinie a une théorie monadique indécidable.*

La démonstration de ce résultat peut s'obtenir par réduction du problème de l'arrêt des machines de Turing. Étant donnée une telle machine, on code son exécution de la manière suivante : chaque ruban de la machine correspond à une ligne de la grille. Passer d'une ligne à la suivante correspond à une étape d'exécution de la machine. Une formule monadique est alors capable de déterminer si la machine de Turing atteint un état final ou non. Ainsi, pour toute machine de Turing, il

existe une formule monadique telle que la grille infinie en est un modèle si et seulement si la machine s'arrête.

Propriété 2.10 *La structure ayant les entiers naturels comme univers et munie des relations «plus 1» et «fois deux» a une théorie monadique indécidable.*

En fait, cette propriété est plus délicate à obtenir que la précédente mais fonctionne sur le même principe. Il s'agit de retrouver dans la structure une famille de grilles aussi grande que l'on veut (dans les deux dimensions).

Un dernier cas de structure possédant une théorie monadique indécidable est donné par la propriété 2.11

Propriété 2.11 *L'union disjointe de tous les graphes finis étiquetés par un ensemble non vide a une théorie monadique indécidable.*

Ce résultat s'obtient en montrant qu'il est possible de reconnaître les grilles rectangulaires au sein de ces graphes. Là encore, chaque grille est susceptible d'être utilisée comme support de calcul d'une machine de Turing. La largeur de la grille correspond à la longueur du ruban utilisé durant l'exécution, et la hauteur correspond au nombre d'étapes de calcul.

2.2.3 Extensions de la logique monadique

Différentes extensions de la logique monadique ont été considérées. La présentation qui en est faite ici n'a aucune ambition d'exhaustivité.

Bien entendu, l'extension la plus directe consiste à utiliser la totalité de la logique du second-ordre, c'est à dire la possibilité d'utiliser des variables dont l'interprétation est une relation d'arité quelconque. La satisfaction de cette logique devient alors indécidable dès que l'univers de la structure est infini. Il s'agit donc d'une extension d'un intérêt relatif dans le cadre d'une étude sur les structures infinies.

Une situation intermédiaire entre la logique monadique et la logique du second-ordre est la *logique du second-ordre gardée*. Il s'agit dans ce cas d'autoriser l'utilisation de variables dont l'interprétation est une relation contenue dans une relation déjà présente dans la structure. Cette logique permet en particulier de quantifier sur des ensembles d'arcs dans un graphe. Le lien entre cette logique et la logique monadique a été largement étudié [Cou97, EGO00, Cou03].

Une autre possibilité d'extension consiste à ajouter des prédicats de cardinalité à la logique monadique, en particulier la possibilité d'exprimer la finitude de l'interprétation d'une variable monadique [Cou90]. La finitude d'un ensemble d'éléments ne peut pas être exprimée en logique monadique dans le cas général, et il s'agit donc d'une extension stricte de la logique. En revanche, sur les arbres de degré fini, la logique monadique et la logique monadique augmentée d'un prédicat de cardinalité sont de même expressivité. En effet, il est possible d'exprimer en

$$\begin{aligned} \text{est-prefixe}(X) &= \forall x.\forall y.((\bigvee_{a \in A} x \xrightarrow{a} y) \wedge y \in X) \Rightarrow x \in X \\ \text{prefixe}(X, Y) &= X \subseteq Y \wedge \text{est-prefixe}(Y) \\ &\quad \wedge \forall Z.((X \subseteq Z \wedge \text{est-prefixe}(Z)) \Rightarrow Y \subseteq Z) \\ \text{infini}(X) &= \exists W. \quad \text{racine} \in W \\ &\quad \wedge \forall Y.(\text{prefixe}(X, Y) \Rightarrow W \subseteq Y) \\ &\quad \wedge \forall x \in W, \exists y \in W. \bigvee_{a \in A} x \xrightarrow{a} y \\ \text{fini}(X) &= \neg \text{infini}(X) \end{aligned}$$

FIG. 2.4 – Expression de la finitude dans un arbre de degré fini en logique monadique.

logique monadique qu'un ensemble de nœuds d'un arbre de degré fini est infini comme le montre la remarque suivante.

Remarque 2.12 *Un codage est présenté à la figure 2.4.*

Remarquons tout d'abord que si X est fini, alors l'ensemble $\text{prefixe}(X)$ des sommets dont est issu un chemin aboutissant dans X est lui-même fini. Le lemme de König nous précise qu'un arbre est infini si et seulement si il contient un sommet de degré infini ou une branche infinie. Dans le cas présent, $\text{prefixe}(X)$ est fini si et seulement s'il ne contient pas de branche infinie.

La formule $\text{est-prefixe}(X)$ est satisfaite si tout père d'un nœud de X appartient à X . L'ensemble $\text{prefixe}(X)$ est alors le plus petit ensemble Y contenant X et satisfaisant $\text{est-prefixe}(Y)$, ce qu'exprime le prédicat $\text{prefixe}(X, Y)$. La formule $\text{infini}(X)$ détermine alors l'existence d'une branche infinie W incluse dans $\text{prefixe}(X)$.

2.2.4 Construction de Feferman-Vaught

Une construction fondamentale pour la suite de ce travail est celle introduite par Feferman et Vaught dans leur étude du produit de structures [FV59] (voir aussi paragraphe 2.3.4). Cette construction ramène la vérification d'une propriété d'un produit de structure à des vérifications sur les structures composantes d'un nombre fini de propriétés, qui ne dépendent que de la formule définissant la propriété initiale.

$$\begin{array}{ll}
\phi \wedge \mathbf{vrai} \Leftrightarrow_s \phi & \phi \vee \mathbf{faux} \Leftrightarrow_s \phi \\
\phi \wedge \phi \Leftrightarrow_s \phi & \phi \vee \phi \Leftrightarrow_s \phi \\
\phi \wedge \psi \Leftrightarrow_s \psi \wedge \phi & \phi \vee \psi \Leftrightarrow_s \psi \vee \phi \\
\phi \wedge (\psi \vee \psi') \Leftrightarrow_s (\phi \wedge \psi) \vee (\phi \wedge \psi') & \phi \vee (\psi \wedge \psi') \Leftrightarrow_s (\phi \vee \psi) \wedge (\phi \vee \psi') \\
\phi \wedge (\phi' \wedge \phi'') \Leftrightarrow_s (\phi \wedge \phi') \wedge \phi'' & \phi \vee (\phi' \vee \phi'') \Leftrightarrow_s (\phi \vee \phi') \vee \phi''
\end{array}$$

$$\begin{array}{ll}
\exists x.(\phi \vee \psi) \Leftrightarrow_s (\exists x.\phi) \vee (\exists x.\psi) & \\
\forall x.(\phi \wedge \psi) \Leftrightarrow_s (\forall x.\phi) \wedge (\forall x.\psi) & \\
\exists x.(\phi \wedge \psi) \Leftrightarrow_s \psi \wedge \exists x.\phi & \text{si } x \notin FV(\psi) \\
\forall x.(\phi \vee \psi) \Leftrightarrow_s \psi \vee \forall x.\phi & \text{si } x \notin FV(\psi)
\end{array}$$

FIG. 2.5 – Quelques invariants logiques.

Dans notre contexte nous considérons cette construction d'un point de vue syntaxique, sans s'intéresser particulièrement au cas du produit. Sous cet angle il s'agit d'une transformation de formules logiques: si il est possible de séparer l'ensemble des variables d'une formule en deux ensembles tels qu'aucun symbole relationnel ne fasse référence à une variable de ces deux ensembles simultanément. alors il est possible de transformer syntaxiquement la formule en une combinaison de formules logiques n'utilisant des variables que de l'un des deux ensembles.

Pour simplifier, on suppose ici que les formules ne contiennent pas de négations, ce qui est sans conséquence pour la suite.

Précisons tout d'abord les transformations syntaxiques autorisées. Ainsi, on définit la relation d'*équivalence syntaxique* \Leftrightarrow_s entre formules comme la plus petite relation satisfaisant les invariants de la figure 2.5, réflexive, symétrique, transitive et compositionnelle (dans le sens où remplacer une sous-formule par une sous-formule équivalente permet d'obtenir une formule équivalente: cela équivaut à dire que les transformations peuvent être appliquées à l'intérieur des formules).

La relation \Leftrightarrow_s préserve les variables libres. Bien entendu, pour \mathcal{S} une Σ -structure, ϕ et ψ telles que $\phi \Leftrightarrow_s \psi$ et γ une valuation de $FV(\phi)$, on a $\gamma, \mathcal{S} \models \phi$ si et seulement si $\gamma, \mathcal{S} \models \psi$. Ainsi, la relation \Leftrightarrow_s est bien «valide».

Soit ϕ une formule logique. Appelons $Vars(\phi)$ l'ensemble des variables apparaissant dans ϕ (cette définition n'a de sens que parce que \Leftrightarrow_s ne contient pas

l' α -conversion). Soit $Rels(\phi)$ l'ensemble des prédicats relationnels apparaissant dans ϕ . On vérifie sans peine que $Vars(\phi)$ et $Rels(\phi)$ sont préservés par \Leftrightarrow_s . Enfin, pour Σ un alphabet relationnel et X un ensemble de variables, on note $\Sigma(X)$ l'ensemble des prédicats relationnels construits sur Σ et X , c'est-à-dire :

$$\Sigma(X) = \{R(x_1, \dots, x_{|R|}) \mid R \in \Sigma, x_1, \dots, x_{|R|} \in X\} .$$

Lemme 2.13 (construction de Feferman-Vaught) *Soit ϕ une formule sur la signature $\Sigma = \Sigma_1 \uplus \Sigma_2$ telle que $Vars(\phi) = X_1 \uplus X_2$ et $Rels(\phi) \subseteq \Sigma_1(X_1) \cup \Sigma_2(X_2)$. Alors il existe un ensemble fini ϕ^{X_1, X_2} de couples de formules tel que :*

$$\begin{aligned} - \text{ pour tout } (\psi_1, \psi_2) \in \phi^{X_1, X_2}, \quad & Vars(\psi_1) \subseteq X_1, \quad Rels(\psi_1) \subseteq \Sigma_1(X_1), \\ & Vars(\psi_2) \subseteq X_2, \quad Rels(\psi_2) \subseteq \Sigma_2(X_2), \end{aligned}$$

$$- \phi \Leftrightarrow_s \bigvee_{(\psi_1, \psi_2) \in \phi^{X_1, X_2}} \psi_1 \wedge \psi_2 .$$

Construction. La conclusion du lemme s'exprime comme l'équivalence avec une disjonction de conjonctions. Il est nécessaire pour pouvoir traiter le cas de la quantification universelle de disposer d'une représentation duale sous forme de conjonctions de disjonctions. Il est possible de passer d'une telle disjonction à une conjonction comme suit. Soit Θ un ensemble de couples de formules, alors on a l'équivalence suivante :

$$\bigvee_{(\phi, \psi) \in \Theta} (\phi \wedge \psi) \quad \Leftrightarrow_s \quad \bigwedge_{(\phi, \psi) \in \bar{\Theta}} (\phi \vee \psi) ,$$

$$\text{avec } \bar{\Theta} = \left\{ \left(\bigvee_{(\phi, \psi) \in \Theta_1} \phi, \bigvee_{(\phi, \psi) \in \Theta_2} \psi \right) \mid \Theta_1 \uplus \Theta_2 = \Theta \right\} .$$

Ceci se montre par récurrence sur le cardinal de Θ . Un fait important de cette construction est que si pour tout $(\phi, \psi) \in \Theta$, $FV(\phi) \subseteq X_1$ et $FV(\psi) \subseteq X_2$ (resp. $Rels(\phi) \subseteq \Sigma_1(X_1)$ et $Rels(\psi) \subseteq \Sigma_2(X_2)$) alors il en est de même pour tout $(\phi, \psi) \in \bar{\Theta}$. Bien entendu, par dualité, une opération similaire qui à Θ associe $\underline{\Theta}$ permet de passer d'une conjonction de disjonctions à une disjonction de conjonctions :

$$\bigwedge_{(\phi, \psi) \in \Theta} (\phi \vee \psi) \quad \Leftrightarrow_s \quad \bigvee_{(\phi, \psi) \in \underline{\Theta}} (\phi \wedge \psi) .$$

L'ensemble ϕ^{X_1, X_2} est alors défini récursivement comme suit :

$$\begin{aligned}
(\phi \vee \psi)^{X_1, X_2} &= \phi^{X_1, X_2} \cup \psi^{X_1, X_2} \\
(\phi \wedge \psi)^{X_1, X_2} &= \overline{\phi^{X_1, X_2} \cup \psi^{X_1, X_2}} \\
(\exists x. \phi)^{X_1, X_2} &= \begin{cases} \{(\exists x. \phi_1, \phi_2) \mid (\phi_1, \phi_2) \in \phi^{X_1, X_2}\} & \text{si } x \in X_1 \\ \{(\phi_1, \exists x. \phi_2) \mid (\phi_1, \phi_2) \in \phi^{X_1, X_2}\} & \text{si } x \in X_2 \end{cases} \\
(\forall x. \phi)^{X_1, X_2} &= \begin{cases} \{(\forall x. \phi_1, \phi_2) \mid (\phi_1, \phi_2) \in \overline{\phi^{X_1, X_2}}\} & \text{si } x \in X_1 \\ \{(\phi_1, \forall x. \phi_2) \mid (\phi_1, \phi_2) \in \overline{\phi^{X_1, X_2}}\} & \text{si } x \in X_2 \end{cases} \\
R(x_1, \dots, x_{|R|}) &= \begin{cases} \{(R(x_1, \dots, x_{|R|}), \text{vrai})\} & \text{si } R \in \Sigma_1 \\ \{(\text{vrai}, R(x_1, \dots, x_{|R|}))\} & \text{si } R \in \Sigma_2 \end{cases} \\
\text{faux}^{X_1, X_2} &= \emptyset \\
\text{vrai}^{X_1, X_2} &= \{(\text{vrai}, \text{vrai})\}
\end{aligned}$$

La validité de cette construction pour \vee , \exists , **vrai** et **faux** est évidente. La validité pour \wedge et \forall s'obtient par dualité. La validité pour $R(x_1, \dots, x_{|R|})$ provient de l'hypothèse sur l'ensemble $\text{Rels}(\phi)$. \square

De nombreuses variantes de cette construction sont possibles. L'important est son aspect syntaxique, et surtout le fait que les quantifications que l'on trouve dans les formules de ϕ^{X_1, X_2} sont plus «simples» que celles de ϕ . Cette notion de simplification peut être mesurée en terme d'alternance de quantifications ou de niveaux d'imbrication... En fait, il arrive que cette mesure dépende elle-même des variables. C'est par exemple le cas dans la preuve du théorème 16.

Application

La construction de Feferman-Vaught est utilisée dans la suite pour décomposer des formules faisant référence à deux structures distinctes en formules faisant référence uniquement à l'une ou l'autre de ces structures. Pour formaliser la possibilité pour une formule logique de faire référence à deux structures, on utilise une technique similaire à celle des logiques multi-sortées. Nous présentons ici une technique ad-hoc pour la suite de ce travail.

L'objectif est de modifier la logique (du premier ordre ou monadique) de manière à définir une relation de satisfaction reliant un couple de structures (au lieu d'une seule structure) à une formule. Ainsi, on modifie la logique (premier ordre, monadique ou autre) afin de définir pour deux structures \mathcal{S}_1 et \mathcal{S}_2 la satisfaction d'une formule ϕ par $(\mathcal{S}_1, \mathcal{S}_2)$, notée $(\mathcal{S}_1, \mathcal{S}_2) \models \phi$. L'idée de cette logique est que la formule ϕ fait explicitement référence à l'indice des structures et peut donc

quantifier sur un élément de \mathcal{S}_1 ou un élément de \mathcal{S}_2 . Les symboles relationnels possèdent aussi une marque permettant d'identifier si ils correspondent à une relation de \mathcal{S}_1 ou de \mathcal{S}_2 (cela est nécessaire pour traiter le cas des relations d'arité zéro).

On définit donc tout d'abord une signature utilisée par les prédicats de cette logique. Soient Σ_1 et Σ_2 deux signatures, notons $\Sigma_1 \sqcup \Sigma_2$ la signature définie comme suit :

$$\Sigma_1 \sqcup \Sigma_2 = \{R^1 \mid R \in \Sigma_1\} \uplus \{R^2 \mid R \in \Sigma_2\}, \quad \text{avec } |R^i| = |R|.$$

Les formules sont alors modifiées de la manière suivante. On suppose l'ensemble des variables du premier ordre (resp. monadique) partagé en deux : $\mathcal{V} = \mathcal{V}^1 \oplus \mathcal{V}^2$ (resp. $\mathcal{V}_m = \mathcal{V}_m^1 \oplus \mathcal{V}_m^2$) de telle sorte qu'il existe pour $i \in [2]$ une bijection de \mathcal{V} sur \mathcal{V}^i (resp. de \mathcal{V}_m sur \mathcal{V}_m^i). Pour x une variables de \mathcal{V} (resp. $X \in \mathcal{V}_m$), $x^i \in \mathcal{V}^i$ (resp. $X^i \in \mathcal{V}_m^i$) est l'image de x (resp. de X) par la bijection correspondante.

Un prédicat de la forme $R^i(x_1, \dots, x_{|R^i|})$ n'est syntaxiquement valide que si toutes les variables x_j appartiennent à \mathcal{V}^i (resp. $x \in X$ n'est syntaxiquement valide que si il existe $i \in [2]$ tel que $x \in \mathcal{V}^i$ et $X \in \mathcal{V}_m^i$).

La relation de satisfaction relie alors une paire de structures à un formule. Soient donc \mathcal{S}_1 une Σ_1 -structure et \mathcal{S}_2 une Σ_2 -structure. Soit \mathcal{X} un ensemble de variables, une valuation de \mathcal{X} sur $(\mathcal{S}_1, \mathcal{S}_2)$ est une application qui à chaque variable telle que $x \in \mathcal{X} \cap \mathcal{V}^i$ pour $i \in [2]$ associe un élément de l'univers de \mathcal{S}_i (resp. à toute variable de $X \in \mathcal{X} \cap \mathcal{V}_m^i$ associe une partie de l'univers de \mathcal{S}_i).

Soit ϕ une formule logique et pour γ une valuation de $FV(\phi)$ sur $(\mathcal{S}_1, \mathcal{S}_2)$, on définit la relation de satisfaction $\gamma, (\mathcal{S}_1, \mathcal{S}_2) \models \phi$ par récurrence sur ϕ comme précédemment.

Lemme 2.14 *Pour Σ_1 et Σ_2 deux alphabets relationnels et toute formule ϕ (monadique ou du premier ordre) sur $\Sigma_1 \sqcup \Sigma_2$, il existe un ensemble fini ϕ^* de couples de formules tel que :*

- pour $(\psi_1, \psi_2) \in \phi^*$, ψ_i est une formule sur la signature Σ_i ,
- pour toute Σ_1 -structure \mathcal{S}_1 et toute Σ_2 -structure \mathcal{S}_2 , $(\mathcal{S}_1, \mathcal{S}_2) \models \phi$ si et seulement si il existe $(\psi_1, \psi_2) \in \phi^*$ tel que $\mathcal{S}_1 \models \psi_1$ et $\mathcal{S}_2 \models \psi_2$.

Preuve. Par application du lemme 2.13. □

2.3 Opérations sur les structures

Jusqu'à présent, toutes les structures rencontrées étaient définies de manière interne, c'est-à-dire en explicitant l'univers et l'interprétation des relations. Dans

la pratique, il est plus agréable de construire les structures par application successive à une structure simple de transformations dont les propriétés sont connues.

Ce paragraphe a pour objectif de présenter de telles transformations et d'en donner les principales caractéristiques. Nous portons une attention particulière à établir les liens reliant ces différentes opérations, et ce dans l'objectif d'éviter les redondances et les définitions équivalentes dans la suite de ce travail.

Notons enfin que toutes ces transformations partagent une propriété fondamentale, celle d'être définies à *isomorphisme près* (sauf en ce qui concerne la modification finie). Ainsi, les utiliser permet de s'abstraire du codage des éléments de l'univers.

2.3.1 Modification finie

La modification finie de structure ne sert pas tant à décrire des structures qu'à préciser les limites d'une famille de structures.

La *modification finie* d'une structure consiste à appliquer une combinaison finie des transformations suivantes :

- ajout d'un élément à l'univers (sans changer les interprétations des symboles),
- ajout d'un hyperarc entre des éléments fixés de l'univers,
- élimination d'un élément de l'univers et, par conséquent, de tous les hyperarcs dans lequel il est impliqué,
- élimination d'un hyperarc.

Lorsque seules les deux premières transformations sont utilisées, on parle d'*ajout fini* à la structure.

2.3.2 Interprétation logique

L'interprétation est une méthode classique en logique pour transformer une structure en une autre, la transformation étant décrite par des formules logiques. L'interprétation logique se définit de manière variée suivant la logique choisie. Une *interprétation* est un triplet $\mathcal{I} = (\delta, \Sigma, (\phi_R)_{R \in \Sigma})$ où $\delta(x)$ est une formule logique d'unique variable libre du premier ordre x , Σ est une signature, et pour tout symbole $R \in \Sigma$, $\phi_R(x_1, \dots, x_{|R|})$ est une formule logique de variables libres du premier ordre $x_1, \dots, x_{|R|}$. Soit \mathcal{S} une structure, l'interprétation \mathcal{I} appliquée à \mathcal{S} est la structure d'univers $\{u \in \mathcal{U}_{\mathcal{S}} \mid \mathcal{S} \models \delta(u)\}$, et l'interprétation d'un symbole relationnel $R \in \Sigma$ est

$$\{(u_1, \dots, u_{|R|}) \mid \forall i \in [|R|]. \mathcal{S} \models \delta(u_i), \mathcal{S} \models \phi_R(u_1, \dots, u_{|R|})\} .$$

La formule δ est appelée *restriction* car elle sélectionne les éléments qui doivent être conservés dans l'univers. Lorsque $\delta = \text{vrai}$, tous les éléments de l'univers sont

conservés par l'interprétation, il s'agit d'une interprétation *sans restriction*. Suivant la logique choisie pour l'interprétation, on parle d'interprétation booléenne, booléenne positive, au premier ordre ou monadique (ou en fait toute autre variante construite sur la logique du premier ordre).

Propriété 2.15 *La composition de deux interprétations (booléennes/booléennes positives/existentielles/au premier ordre/monadique) est une interprétation dans la même logique.*

Propriété 2.16 *L'interprétation (existentielle/au premier ordre/monadique) préserve la décidabilité de la théorie correspondante.*

L'interprétation logique est une opération tellement utile pour la suite de ce travail qu'il convient de lui donner une notation spéciale. Considérons une syntaxe logique close par disjonction (*i.e.* si ϕ et ψ sont des formules bien formées, alors $\phi \vee \psi$ l'est aussi). Soient R_1, \dots, R_n des symboles (éventuellement plusieurs fois le même), et ψ_1, \dots, ψ_n des formules logiques dans la logique fixée telles que $\psi_i(x_1, \dots, x_{|R_i|})$ a pour variables libres les $x_1, \dots, x_{|R_i|}$, alors, la notation :

$$\left| \begin{array}{l} R_1(x_1, \dots, x_{|R_1|}) \quad : \quad \psi_1(x_1, \dots, x_{|R_1|}) \\ \vdots \\ R_n(x_1, \dots, x_{|R_n|}) \quad : \quad \psi_n(x_1, \dots, x_{|R_n|}) \end{array} \right|$$

représente l'interprétation logique :

$$(\text{vrai}, \{R_i \mid i \in [n]\}, \phi), \quad \text{avec } \phi_R = \bigvee_{\substack{i \in [n] \\ R_i = R}} \psi_i(x_1, \dots, x_{|R|}) .$$

On étend cette notation pour avoir la possibilité d'exprimer une restriction.

$$\left| \begin{array}{l} x \in \mathcal{U} \quad \quad \quad : \quad \delta(x) \\ R_1(x_1, \dots, x_{|R_1|}) \quad : \quad \psi_1(x_1, \dots, x_{|R_1|}) \\ \vdots \\ R_n(x_1, \dots, x_{|R_n|}) \quad : \quad \psi_n(x_1, \dots, x_{|R_n|}) \end{array} \right|$$

représente l'interprétation logique :

$$(\delta, \{R_i \mid i \in [n]\}, \phi), \quad \text{avec } \phi_R = \bigwedge_{j \in [|R|]} \delta(x_j) \wedge \bigvee_{\substack{i \in [n] \\ R_i = R}} \psi_i(x_1, \dots, x_{|R|}) .$$

Un cas particulier d'interprétation est le renommage. Soit Σ une signature et K une relation binaire entre Σ_1 et Σ_2 reliant uniquement des symboles de même arité, alors le *renommage* des symboles par K est l'interprétation :

$$\left| R(x_1, \dots, x_{|R|}) \quad : \quad R'(x_1, \dots, x_{|R|}), \quad (R', R) \in K \right| .$$

Cette opération extrêmement simple permet de changer quand nécessaire la signature d'une structure afin, par exemple, d'éviter des confusions dues au choix de nommage des symboles.

Plusieurs autres transformations de graphes [Cau96, Cau02, Urv02] sont des cas particuliers d'interprétations :

la substitution rationnelle inverse : pour chaque lettre a de l'alphabet du graphe résultat, on se fixe un langage rationnel L_a sur l'alphabet du graphe d'origine. Le graphe résultat a alors le même ensemble de sommets que le graphe originel¹, et il y a un arc étiqueté par a entre deux sommets s'il y a un chemin étiqueté par un mot de L_a entre ces deux sommets dans le graphe originel. On utilise pour cela la formule `accessibilite`[L_a] définie dans l'exemple 2.8. Il s'agit donc d'une interprétation monadique sans restriction.

la substitution finie inverse : il s'agit d'un cas particulier de la substitution rationnelle inverse : les langages associés à chaque lettre doivent être finis au lieu de rationnels. Il s'agit d'une interprétation sans restriction en logique du premier ordre.

le morphisme inverse : le morphisme inverse est un cas encore plus restreint : chaque langage ne contient au plus qu'un mot. Il s'agit toujours d'une interprétation sans restriction en logique du premier ordre. Le morphisme inverse possède la propriété supplémentaire de préserver le déterminisme d'un graphe.

l'élimination des sommets isolés : cette transformation élimine les sommets de degré nul d'un graphe. C'est une interprétation avec uniquement de la restriction en logique du premier ordre.

la restriction par accessibilité : cette transformation ne conserve d'un graphe enraciné que les sommets accessibles à partir de sa racine. Il s'agit d'une interprétation en logique monadique ou monadique.

la restriction rationnelle : elle est paramétrée par un langage rationnel L . Cette transformation ne conserve que les sommets accessibles à partir de la racine par un chemin étiqueté par un mot de L .

le marquage rationnel : il est paramétré par un langage rationnel L et un symbole $\#$. Appliqué à un graphe enraciné, il laisse ce dernier inchangé si ce n'est qu'à les chaque sommet accessible à partir de la racine par un chemin étiqueté par un mot de L , est attachée une boucle étiquetée par $\#$.

Toutes ces transformations sont contrôlées par des chemins entre sommets. Il est courant d'en augmenter la puissance en autorisant dans les langages de prendre en sens inverse les arcs. On utilise pour cela la lettre barrée. Ainsi \bar{a} représente

1. Dans les travaux originaux de Caucal les sommets sont implicitement restreints aux extrémités des arcs.

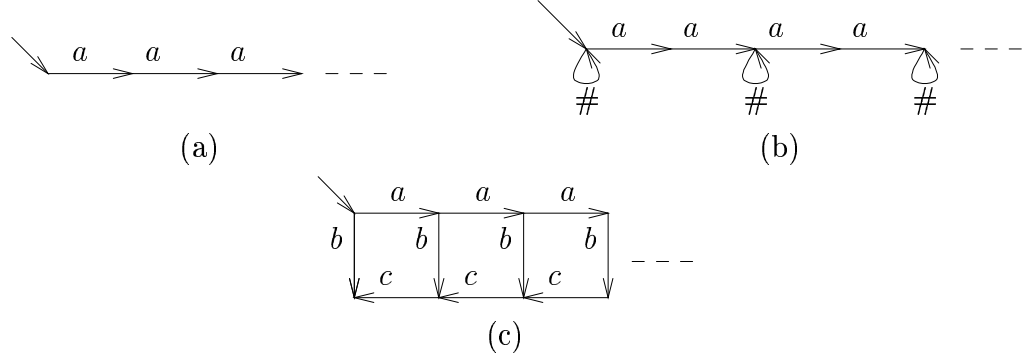


FIG. 2.6 – Demi-droite des entiers enracinée (a), marquée (c), et échelle enracinée (b).

implicitement la relation \xrightarrow{a} inversée. On parle alors de substitution rationnelle inverse (resp. substitution finie inverse, resp. restriction rationnelle, resp. marquage rationnel) *avec retour*.

Il peut paraître *a priori* étonnant qu’une telle profusion de transformations d’expressivité relativement identique ait vue le jour. La raison est d’une part historique, mais de plus, ces interprétations possèdent des propriétés non vérifiées. dans le cas général : elles sont continues (au sens des ordres partiels complets présentés au paragraphe 3.1 et appliqués aux structures page 52), et en ce qui concerne les substitutions inverses, elles correspondent à des transformations naturelles de langages sur les traces [Urv02].

Exemple 2.17 L’échelle enracinée (figure 2.6(c)) s’obtient à partir de la demi-droite des entiers (figure 2.6(a)) en appliquant tout d’abord un marquage rationnel : tous les sommets atteignables à partir de la racine par un chemin étiqueté par un mot de $(aa)^*$ sont marqués d’une boucle $\#$ (figure 2.6(b)). Pour obtenir l’échelle, on applique alors une substitution finie inverse avec retour. On met un arc étiqueté par a entre un sommet marqué et son deuxième successeur sur la demi-droite des entiers, soit $L_a = \#aa$. On met un arc étiqueté par b entre un sommet marqué et son successeur, soit $L_b = \#a$. On met un arc étiqueté par c entre un sommet non marqué et son deuxième prédécesseur, soit $L_c = \bar{a}\#\bar{a}$. Notons que la substitution inverse ne permet pas de tester l’absence d’une marque. Pour être sûr que la substitution a lieu au bon endroit, il est nécessaire d’aller chercher une marque un peu plus loin, puis de revenir (ce que fait le mot $\bar{a}\#\bar{a}$).

2.3.3 Union disjointe

Soient deux structures \mathcal{S}_1 et \mathcal{S}_2 , l’union disjointe de \mathcal{S}_1 et \mathcal{S}_2 , notée $\mathcal{S}_1 \oplus \mathcal{S}_2$, est la structure d’univers $\{1\} \times \mathcal{S}_1 \cup \{2\} \times \mathcal{S}_2$ et dont l’interprétation d’un symbole

R est :

$$R^{\mathcal{S}_1 \oplus \mathcal{S}_2} = \begin{aligned} & \{((1, u_1), \dots, (1, u_{|R|})) \mid R^{\mathcal{S}_1}(u_1, \dots, u_{|R|})\} \\ \cup & \{((2, u_1), \dots, (2, u_{|R|})) \mid R^{\mathcal{S}_2}(u_1, \dots, u_{|R|})\} . \end{aligned}$$

L'*union disjointe explicite* ajoute de l'information à cette construction afin de pouvoir déterminer l'origine des relations. Elle est définie comme suit. Soit Σ_1 et Σ_2 deux signatures, l'union disjointe explicite d'une Σ_1 -structure avec une Σ_2 -structure produit une structure de signature $\Sigma' = \{R^1 \mid R \in \Sigma_1\} \uplus \{R^2 \mid R \in \Sigma_2\}$. Soient \mathcal{S}_1 une Σ_1 -structure et \mathcal{S}_2 une Σ_2 -structure, l'union disjointe explicite de \mathcal{S}_1 et \mathcal{S}_2 est la Σ' -structure $\mathcal{S}_1 \oplus_e \mathcal{S}_2$ d'univers $\{1\} \times \mathcal{S}_1 \cup \{2\} \times \mathcal{S}_2$ et dont l'interprétation d'un symbole $R^i \in \Sigma'$ pour $i \in [2]$ est :

$$(R^i)^{\mathcal{S}_1 \oplus_e \mathcal{S}_2} = \{((i, u_1), \dots, (i, u_{|R^i|})) \mid R^{\mathcal{S}_i}(u_1, \dots, u_{|R^i|})\} .$$

Le lemme suivant énonce comment passer d'une de ces formes d'union disjointe à l'autre au moyen de recoloriages.

Propriété 2.18 *Étant données deux signatures Σ_1 et Σ_2 , il existe trois renommages \mathcal{I} , \mathcal{I}_1 et \mathcal{I}_2 tels que pour toute Σ_1 -structure \mathcal{S}_1 et toute Σ_2 -structure \mathcal{S}_2 , on a :*

$$\begin{aligned} \mathcal{S}_1 \oplus \mathcal{S}_2 &= \mathcal{I}(\mathcal{S}_1 \oplus_e \mathcal{S}_2) \\ \text{et} \quad \mathcal{S}_1 \oplus_e \mathcal{S}_2 &= \mathcal{I}_1(\mathcal{S}_1) \oplus \mathcal{I}_2(\mathcal{S}_2) . \end{aligned}$$

Preuve. Très naturellement \mathcal{I} renomme chaque symbole R^i en R et pour $i \in [2]$, \mathcal{I}_i renomme chaque symbole R en R^i . \square

Propriété 2.19 *L'union disjointe (resp. l'union disjointe explicite) de deux structures possédant une théorie au premier ordre (resp. monadique) décidable a une théorie au premier ordre (resp. monadique) décidable.*

Preuve. Par application du lemme 2.13. \square

2.3.4 Produits

L'union disjointe de deux structures produit une structure dont le nouvel univers est l'union disjointe des univers des deux structures. Il n'y a en fait pas beaucoup de choix possibles pour de telles opérations. C'est un peu différent dans le cas du produit. Une fois adopté le principe de produire une structure dont l'univers est le produit des univers des structures arguments, il reste des choix

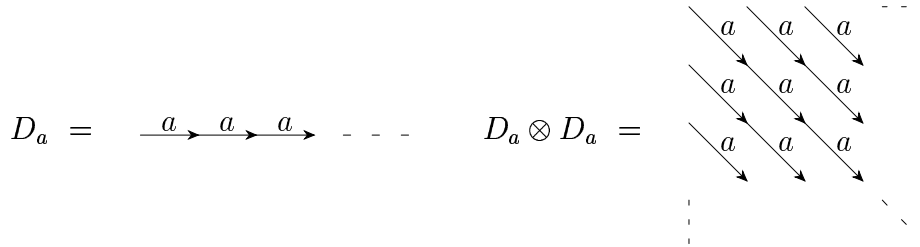


FIG. 2.7 – *Produit synchrone de la demi-droite des entiers avec elle-même*

possibles pour définir les interprétations des relations. Ainsi, nous considérons trois types de produits dans ce travail : le produit synchrone, le produit généralisé qui en est une variante, et le produit asynchrone².

Soient \mathcal{S}_1 et \mathcal{S}_2 deux structures sur la signature Σ , le *produit synchrone* (aussi appelé produit cartésien) de \mathcal{S}_1 et \mathcal{S}_2 est la structure $\mathcal{S}_1 \otimes \mathcal{S}_2$ définie par :

$$\begin{aligned} \mathcal{U}_{\mathcal{S}_1 \otimes \mathcal{S}_2} &= \mathcal{U}_{\mathcal{S}_1} \times \mathcal{U}_{\mathcal{S}_2} , \\ \text{pour } R \in \Sigma, \quad R^{\mathcal{S}_1 \otimes \mathcal{S}_2} &= \{ ((u_1, v_1), \dots, (u_{|R|}, v_{|R|})) \mid \begin{aligned} &R^{\mathcal{S}_1}(u_1, \dots, u_{|R|}) \\ &\wedge R^{\mathcal{S}_2}(v_1, \dots, v_{|R|}) \} . \end{aligned} \end{aligned}$$

Exemple 2.20 *La figure 2.7 montre le résultat du produit de la demi-droite des entiers avec elle-même. Le résultat est une union disjointe infinie de demi-droites des entiers.*

Propriété 2.21 *Soient \mathcal{A}_1 et \mathcal{A}_2 deux automates, alors $\mathcal{A}_1 \otimes \mathcal{A}_2$ est un automate tel que :*

$$\mathcal{L}(\mathcal{A}_1 \otimes \mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) .$$

Théorème 3 (Feferman et Vaught [FV59]) *Le produit synchrone préserve la décidabilité de la théorie au premier ordre.*

Il existe des graphes dont la théorie au second ordre monadique est décidable et dont le produit synchrone possède une accessibilité indécidable. Pour le montrer, on peut par exemple réduire l'indécidabilité de la vacuité de l'intersection de deux langages algébriques à une question d'accessibilité sur un graphe produit. Pour cela, on admet momentanément que pour tout langage algébrique, il existe effectivement un automate possédant une théorie monadique décidable et dont il est le langage accepté (voir théorèmes 12 et 13 page 92). La vacuité de l'intersection de deux langages algébriques étant indécidable, cela signifie que la vacuité

² En général, sous la terminologie de produit sans autres précisions, on entend le produit synchrone ou le produit généralisé.

du langage accepté par l'automate produit est indécidable (d'après la propriété 2.21). Pourtant, ceci revient à déterminer s'il existe un chemin entre le sommet initial et un sommet final, c'est à dire une question d'accessibilité.

En revanche, le produit synchrone avec une structure finie préserve la décidabilité de la théorie monadique. Ce résultat est présenté à titre d'exemple (exemple 2.26) pour illustrer les transductions monadiques.

Le produit généralisé permet une utilisation plus aisée du produit synchrone. Il se définit comme suit. Étant données deux structures \mathcal{S}_1 et \mathcal{S}_2 de signatures respectives Σ_1 et Σ_2 , le *produit généralisé* de \mathcal{S}_1 et \mathcal{S}_2 est une structure $\mathcal{S}_1 \otimes_g \mathcal{S}_2$ sur la signature $\Sigma' = \{R^1 \mid R \in \Sigma_1\} \uplus \{R^2 \mid R \in \Sigma_2\}$ avec pour tout $R^i \in \Sigma'$, $|R^i| = |R|$. La structure $\mathcal{S}_1 \otimes_g \mathcal{S}_2$ est définie comme suit:

$$\mathcal{U}_{\mathcal{S}_1 \otimes_g \mathcal{S}_2} = \mathcal{U}_{\mathcal{S}_1} \times \mathcal{U}_{\mathcal{S}_2} ,$$

et pour $R^i \in \Sigma'$, $(R^i)^{\mathcal{S}_1 \otimes_g \mathcal{S}_2} = \{((u_1^1, u_1^2), \dots, (u_{|R|}^1, u_{|R|}^2)) \mid R^{\mathcal{S}_i}(u_1^i, \dots, u_{|R|}^i)\} .$

En fait, produit généralisé et produit synchrone sont des opérations d'expressivité identique si l'on dispose d'interprétations booléennes positives comme le montre la propriété 2.22.

Propriété 2.22 *Pour toute signature Σ , il existe trois interprétations booléennes positives \mathcal{I} , \mathcal{I}_1 et \mathcal{I}_2 telles que pour toutes Σ -structures \mathcal{S}_1 et \mathcal{S}_2 on a*

$$\mathcal{S}_1 \otimes_g \mathcal{S}_2 = \mathcal{I}_1(\mathcal{S}_1) \otimes \mathcal{I}_2(\mathcal{S}_2)$$

et $\mathcal{S}_1 \otimes \mathcal{S}_2 = \mathcal{I}(\mathcal{S}_1 \otimes_g \mathcal{S}_2) .$

Preuve. L'interprétation \mathcal{I}_i pour $i \in [2]$ renomme $R \in \Sigma$ en R^i et attribue comme interprétation de R^j pour $j \in [2] - \{i\}$ la relation complète d'arité $|R|$. L'interprétation \mathcal{I} donne comme interprétation d'un symbole $R \in \Sigma$ l'intersection des interprétations de R^1 et R^2 . \square

Le troisième type de produit que nous considérons est le produit asynchrone. Étant données deux structures \mathcal{S}_1 et \mathcal{S}_2 , le *produit asynchrone* de \mathcal{S}_1 et \mathcal{S}_2 est la structure $\mathcal{S}_1 \square \mathcal{S}_2$ définie par :

$$\mathcal{U}_{\mathcal{S}_1 \square \mathcal{S}_2} = \mathcal{U}_{\mathcal{S}_1} \times \mathcal{U}_{\mathcal{S}_2} ,$$

pour $R \in \Sigma$, $R^{\mathcal{S}_1 \square \mathcal{S}_2} = \{((u_1, v), \dots, (u_{|R|}, v)) \mid R^{\mathcal{S}_1}(u_1, \dots, u_{|R|}), v \in \mathcal{U}_{\mathcal{S}_2}\} \cup \{((u, v_1), \dots, (u, v_{|R|})) \mid u \in \mathcal{U}_{\mathcal{S}_1}, R^{\mathcal{S}_2}(u_1, \dots, u_{|R|})\} .$

Exemple 2.23 *Le produit asynchrone de deux demi-droites des entiers permet d'obtenir le quart de grille infinie (cf figure 2.8). Sachant que la demi-droite des entiers a une théorie monadique décidable (théorème 1 de Büchi), et en appliquant la résultat d'indécidabilité de la théorie monadique du quart de grille infini*

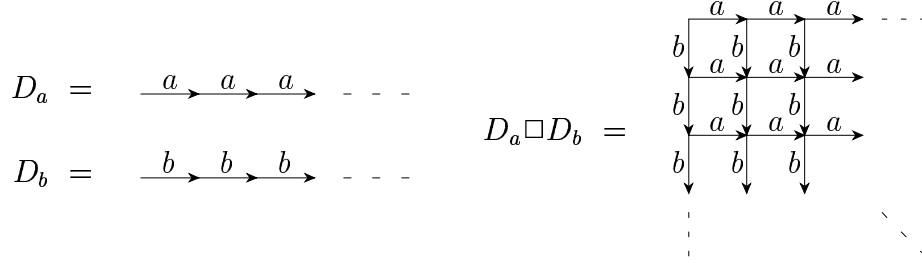


FIG. 2.8 – Le quart de grille infinie comme produit asynchrone de demi-droites des entiers

(propriété 2.9), on en déduit que le produit asynchrone ne préserve pas la décidabilité de la théorie monadique. En revanche nous verrons que cette opération préserve la décidabilité des propriétés d'accessibilité.

Produits synchrones et asynchrones ont des propriétés bien distinctes. Pourtant, ils sont inter-convertibles au moyen d'interprétations logiques comme le montre la propriété ci-dessous.

Propriété 2.24 Soit Σ un alphabet relationnel, alors

1. il existe un renommage \mathcal{I}_1 et une interprétation existentielle \mathcal{I} tels que pour toutes structures \mathcal{S}_1 et \mathcal{S}_2 de signature Σ ,

$$\mathcal{S}_1 \otimes \mathcal{S}_2 = \mathcal{I}(\mathcal{I}_1(\mathcal{S}_1) \square \mathcal{S}_2) ,$$

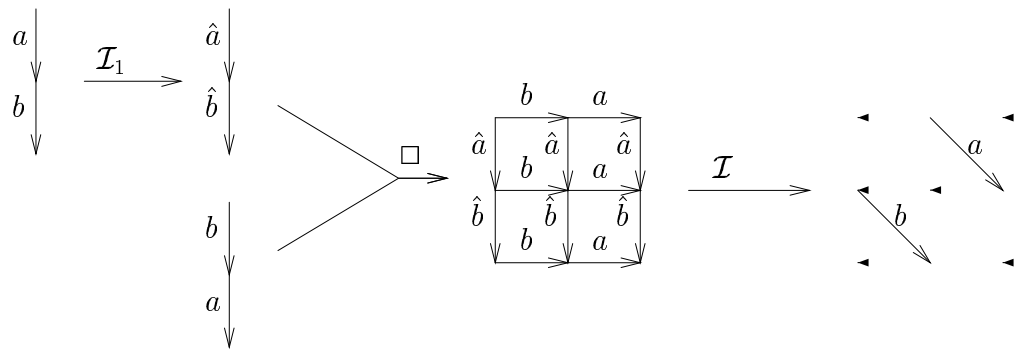
2. il existe deux interprétations booléennes positives \mathcal{I}_1 et \mathcal{I}_2 et un renommage \mathcal{I} tels que pour toutes structures \mathcal{S}_1 et \mathcal{S}_2 de signature Σ :

$$\mathcal{S}_1 \square \mathcal{S}_2 = \mathcal{I}(\mathcal{I}_1(\mathcal{S}_1) \otimes \mathcal{I}_2(\mathcal{S}_2)) .$$

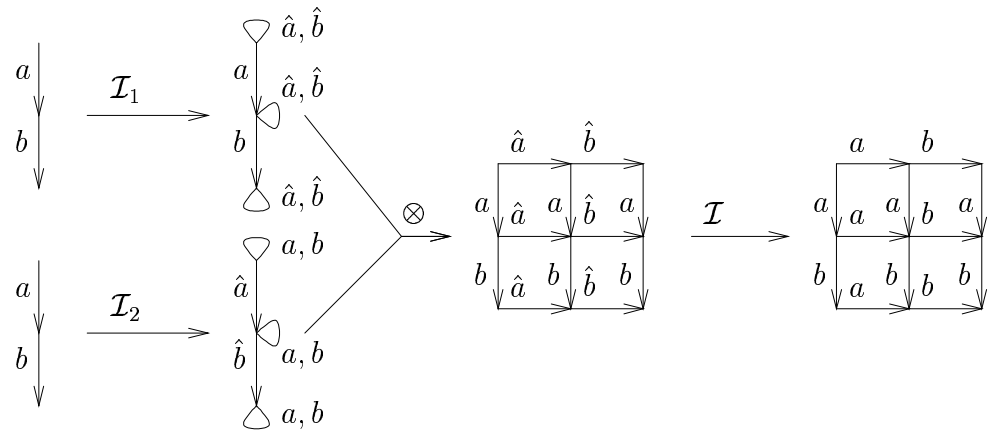
Preuve. Soit $\hat{\Sigma}$ l'ensemble des symboles \hat{R} d'arité $|R|$ pour R un symbole de Σ . La figure 2.9 illustre les constructions de cette propriété.

Résultat 1. L'interprétation \mathcal{I}_1 renomme chaque symbole R en un symbole \hat{R} . Cela revient à rendre disjointes les signatures des deux structures. L'interprétation existentielle est alors la suivante :

$$\mathcal{I} = \left| \begin{array}{l} R(x_1, \dots, x_{|R|}) \quad : \quad \exists y_{1,1} \dots \exists y_{1,|R|} \dots \exists y_{|R|,1} \dots \exists y_{|R|,|R|} \cdot \\ \bigwedge_{i \in [|R|]} R(y_{i,1}, \dots, y_{i,|R|}) \\ \wedge \bigwedge_{j \in [|R|]} \hat{R}(y_{1,j}, \dots, y_{|R|,j}) \\ \wedge \bigwedge_{i \in [|R|]} x_i = y_{i,i} \end{array} \right|$$



(1) du produit asynchrone vers le produit synchrone.



(2) du produit synchrone vers le produit asynchrone.

FIG. 2.9 – Illustration des constructions de la propriété 2.24.

La formule logique de cette interprétation devine une matrice d'éléments de l'univers $y_{1,1}, \dots, y_{n,n}$ dont la diagonale est x_1, \dots, x_n . Un hyperarc R relie x_1, \dots, x_n si à chaque colonne de la matrice y correspond un hyperarc R de la première structure et à chaque ligne de y correspond un hyperarc R de la seconde structure (c'est à dire un hyperarc \hat{R}).

Résultat 2.

$$\begin{aligned} \text{Soient } \mathcal{I}_1 = & \left| \begin{array}{ll} R(x_1, \dots, x_{|R|}) & : R(x_1, \dots, x_{|R|}), & R \in \Sigma \\ \hat{R}(x_1, \dots, x_{|R|}) & : x_1 = x_2 \wedge \dots \wedge x_1 = x_{|R|}, & R \in \Sigma \end{array} \right|, \\ \mathcal{I}_2 = & \left| \begin{array}{ll} R(x_1, \dots, x_{|R|}) & : x_1 = x_2 \wedge \dots \wedge x_1 = x_{|R|}, & R \in \Sigma \\ \hat{R}(x_1, \dots, x_{|R|}) & : R(x_1, \dots, x_{|R|}), & R \in \Sigma \end{array} \right|, \\ \text{et } \mathcal{I} = & \left| \begin{array}{ll} R(x_1, \dots, x_{|R|}) & : R(x_1, \dots, x_{|R|}) \vee \hat{R}(x_1, \dots, x_{|R|}), & R \in \Sigma \end{array} \right|. \end{aligned}$$

Les interprétations \mathcal{I}_1 et \mathcal{I}_2 remplissent deux tâches : elles renomment les symboles de \mathcal{S}_1 et \mathcal{S}_2 afin que les signatures soient disjointes, et elles ajoutent de nouveaux symboles d'interprétation «identité». Le produit synchrone produit alors la structure souhaitée si ce n'est que les symboles de Σ sont utilisés sur une composante et les symboles de $\hat{\Sigma}$ sur l'autre. Le renommage \mathcal{I} élimine ce dernier point. \square

2.3.5 Transduction monadique

Dans les travaux de Courcelle [Cou94], une opération de base est la *transduction monadiquement définissable sans paramètre*. Dans le présent travail, le terme de *transduction monadique* est employé³. Le principe de la transduction monadique est d'appliquer à la structure successivement une duplication et une interprétation. Bien entendu, cette opération préserve la décidabilité de la théorie monadique.

Théorème 4 (Courcelle) *Si une structure \mathcal{S} a une théorie monadique décidable et si T est une transduction monadique, alors $T(\mathcal{S})$ a une théorie monadique décidable.*

Une autre propriété importante de ces transductions est la suivante.

Propriété 2.25 (Courcelle) *La composition de deux transductions monadiques est une transduction monadique.*

Formellement, étant donnés deux alphabets Σ et Σ' de symboles relationnels, une *transduction monadique* des structures de signature Σ dans les structures de signature Σ' est un quadruplet

$$T = (\delta, N, (\phi_n)_{n \in N}, (\psi_{R, n_1, \dots, n_{|R|}})_{R \in \Sigma', n \in N^{|R|}})$$

3. Cette terminologie ne doit pas être confondue avec les transductions MS-compatibles définies aussi dans les travaux de Courcelle.

où :

- δ est une formule monadique close sur la signature Σ ,
- N est un ensemble fini,
- $\phi_n(x)$ pour $n \in N$ est une formule monadique sur la signature Σ d'unique variable libre du premier ordre x ,
- pour R un symbole de Σ' et $n \in N^{|R|}$, $\psi_{R,n_1,\dots,n_{|R|}}(x_1,\dots,x_{|R|})$ est une formule monadique sur la signature Σ de variables libres $x_1,\dots,x_{|R|}$.

Soit \mathcal{S} une structure sur la signature Σ , la structure $T(\mathcal{S})$ sur la signature Σ' est définie par :

- si $\mathcal{S} \models \delta$, alors $T(\mathcal{S})$ n'est pas défini,
- sinon, l'univers de $T(\mathcal{S})$ est :

$$\mathcal{U}_{T(\mathcal{S})} = \{(n, u) \in N \times \mathcal{U}_{\mathcal{S}} \mid \mathcal{S} \models \phi_n(u)\},$$

et l'interprétation d'un symbole $R \in \Sigma'$ est :

$$R^{T(\mathcal{S})} = \{((n_1, u_1), \dots, (n_{|R|}, u_{|R|})) \mid \mathcal{S} \models \psi_{R,n_1,\dots,n_{|R|}}(u_1, \dots, u_{|R|})\}.$$

Exemple 2.26 *Considérons une structure finie \mathcal{S}_0 . Il s'agit de produire une transduction monadique effectuant le produit synchronisé d'une structure quelconque \mathcal{S} par \mathcal{S}_0 . Cela s'obtient en appliquant la transduction monadique $T = \{\text{vrai}, \mathcal{U}_{\mathcal{S}_0}, \phi, \psi\}$ avec*

$$\begin{aligned} \phi_u &= \text{vrai} , \\ \text{et } \psi_{R,u_1,\dots,u_{|R|}}(x_1,\dots,x_{|R|}) &= \begin{cases} R(x_1, \dots, x_{|R|}) & \text{si } R^{\mathcal{S}_0}(u_1, \dots, u_{|R|}) \\ \text{faux} & \text{sinon.} \end{cases} \end{aligned}$$

2.3.6 Dépliage

Le dépliage est une opération qui transforme les graphes enracinés en arbres. Si l'on voit un graphe comme une machine dont chaque transition est représentée par un arc et l'état initial est la racine, le dépliage fournit une représentation des exécutions possibles de cette machine (en fait, le graphe déplié est bisimilaire au graphe original).

Étant donné un graphe enraciné \mathcal{G} , son dépliage $Depliage(\mathcal{G})$ est un arbre tel que :

- les nœuds sont les chemins d'origine la racine dans \mathcal{G} ,
- le nœud racine $racine^{Depliage(\mathcal{G})}$ est le chemin de longueur nulle $racine^{\mathcal{G}}$,
- pour toute étiquette a

$$\xrightarrow{a}_{Depliage(\mathcal{G})} = \left\{ (\pi, \pi a s) \mid \pi \in \mathcal{U}_{Depliage(\mathcal{G})}, \text{dest}(\pi) \xrightarrow{a}_{\mathcal{G}} s \right\}.$$

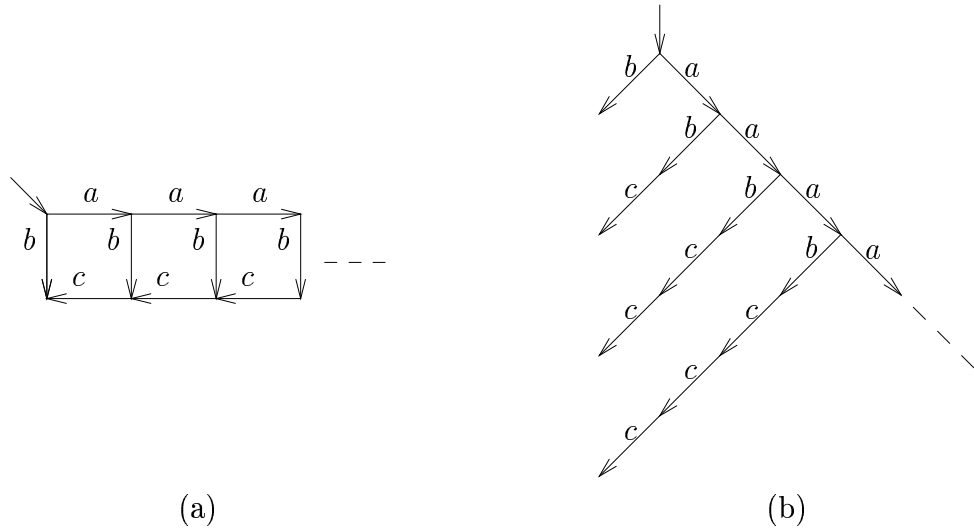


FIG. 2.10 – L 'échelle enracinée (a) et son dépliage (b).

Pour \mathcal{G} un graphe et v un sommet, $Depliage(\mathcal{G}, v)$ représente le graphe \mathcal{G} déplié après avoir substitué v à la racine.

Exemple 2.27 *L'échelle déjà rencontrée précédemment peut être dépliée. Le résultat est présenté à la figure 2.10.*

Théorème 5 (Courcelle et Walukiewicz[CW98]) *Le dépliage préserve la décidabilité de la théorie monadique.*

La démonstration de ce résultat dans le cas général dépasse le cadre de ce mémoire. Une version pour graphes déterministes a été donnée par Courcelle [Cou95]. Une autre démonstration de ce cas restreint est donnée au théorème 10 dans le cadre du μ -calcul.

2.3.7 Structure arborescente

La structure arborescente peut être vue comme une extension de la notion de dépliage : sous les bonnes hypothèses, et combinées avec d'autres transformations, ces deux opérations sont équivalentes (voir les propriétés 2.29 et 2.30). Il s'agit aussi d'une construction qui a l'avantage d'avoir un sens non seulement sur les graphes, mais aussi sur les structures dans toute leur généralité.

Étant donnée une structure \mathcal{S} et une (nouvelle) étiquette $\#$, la structure arborescente de \mathcal{S} , $\mathcal{S}^\#$ est définie par :

- l'univers est l'ensemble des suites non vides d'éléments de $\mathcal{U}_{\mathcal{S}}$, *c.a.d.* en empruntant la notation des langages formels, $\mathcal{U}_{\mathcal{S}}^+$,

- la relation $\xrightarrow{\#}$ duplique le dernier élément de chaque séquence. Son interprétation est

$$\#^{\mathcal{S}^{\#}} = \{(Uu, Uuu) \mid U \in \mathcal{U}_S^*, u \in \mathcal{U}_S\} ,$$

- la relation R pour tout symbole R se comporte comme dans la structure originale, mais n'affecte que le dernier élément de la séquence: son interprétation est :

$$R^{\mathcal{S}^{\#}} = \{(Uu_1, \dots, Uu_{|R|}) \mid U \in \mathcal{U}_S^*, R^{\mathcal{S}}(u_1, \dots, u_{|R|})\} .$$

Quand la structure arborescente est appliquée à un graphe, le résultat est aussi un graphe. Par convention, la structure arborescente issue d'un graphe enraciné est aussi un graphe enraciné, et sa racine est le mot d'une lettre *racine*.

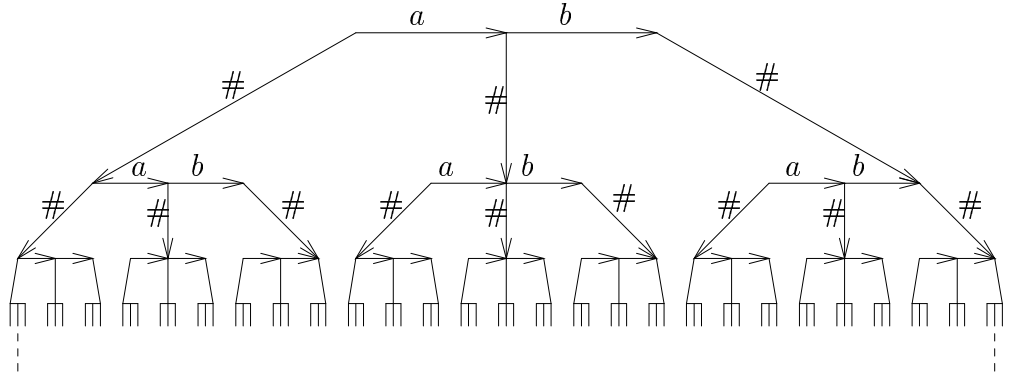


FIG. 2.11 – La structure arborescente obtenue à partir de $\xrightarrow{a} \xrightarrow{b}$.

Exemple 2.28 La figure 2.11 donne un exemple de structure arborescente d'un graphe à trois sommets.

L'une des propriétés fondamentales qui justifie son introduction est donnée par le théorème 6.

Théorème 6 (Muchnik-Walukiewicz [Wal96]) La structure arborescente préserve la décidabilité de la théorie monadique.

Historiquement, une première notion de structure arborescente a été introduite par Shelah [She75]. Elle se distinguait de la construction présente en ce qu'elle ne possédait pas la relation $\#$ mais à la place une relation binaire *fil*s dont l'interprétation est $\{(Uu, Uuv) \mid U \in \mathcal{U}_S^*, u, v \in \mathcal{U}_S\}$. La preuve de préservation de la décidabilité de la théorie monadique de cette construction est attribuée à Stupp dans [Tho90]. Semenov [Sem84] mentionne quant à lui un résultat plus fort pour une construction possédant un prédicat unaire supplémentaire *clone* dont

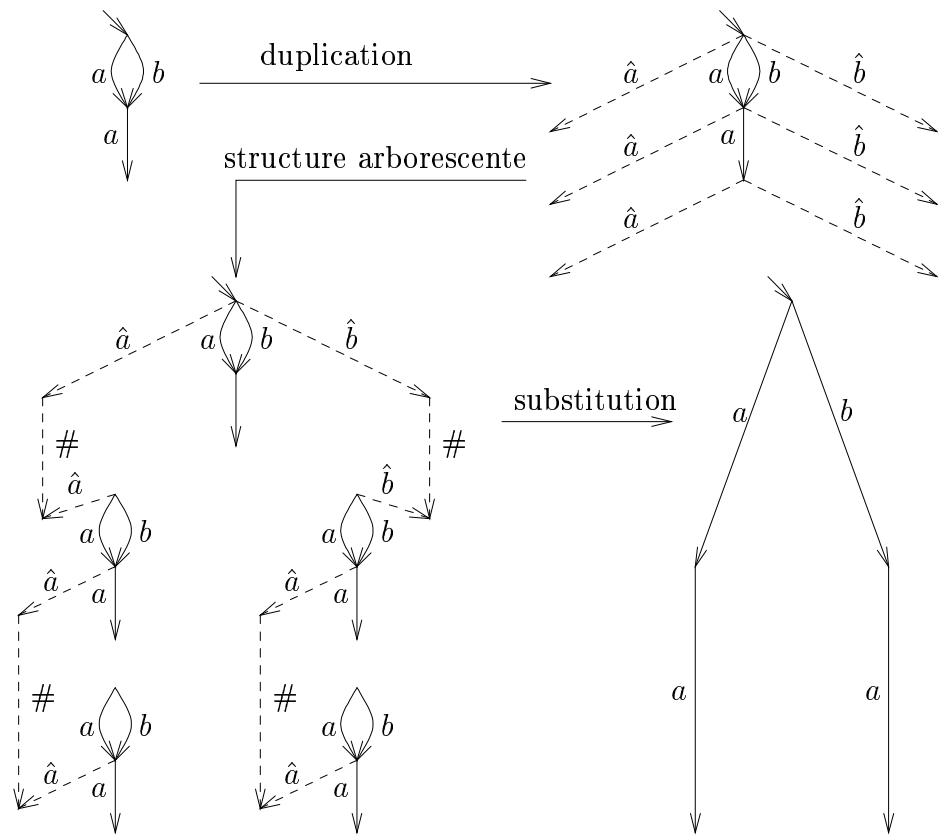


FIG. 2.12 – Exemple d'obtention d'un dépliage par une structure arborescente. Seule une partie du graphe obtenu après construction de la structure arborescente est représentée.

l'interprétation est $\{Uuu \mid U \in \mathcal{U}_S^*, u \in \mathcal{U}_S\}$. Il attribue ce résultat à Muchnik. La construction présentée dans ce travail s'obtient à partir de cette dernière par une simple interprétation booléenne positive. Une preuve publiée de ce résultat a été proposée par Walukiewicz [Wal96].

Les deux propriétés suivantes, montrent le lien qui relie cette transformation au dépliage de graphes.

Propriété 2.29 *Le dépliage d'un graphe s'obtient par applications successives d'une duplication, d'une construction de structure arborescente, d'une substitution finie inverse avec retour et d'une restriction aux sommets accessibles.*

Construction. Soit A l'ensemble des étiquettes du graphe. Le principe est d'appliquer la substitution $L_a = \#a$. Cela ne suffit pas car il y a confusion quand deux arcs étiquetés différemment ont même origine et même destination. Pour résoudre ce problème on effectue en premier une \hat{A} -duplication où \hat{A} est l'ensemble $\{\hat{a} \mid a \in A\}$ d'étiquettes fraîches. Suit la construction de structure arborescente puis une substitution inverse avec retour définie par le langage $L_a = \hat{a}\#\bar{\hat{a}}a$. La figure 2.12 illustre cette construction. \square

Propriété 2.30 *Sur un arbre, la structure arborescente s'obtient par applications successives d'une substitution finie inverse avec retour, d'un dépliage, d'une restriction rationnelle et d'une substitution finie inverse.*

Construction. De nouveau \hat{A} représente l'alphabet $\{\hat{a} \mid a \in A\}$. La première étape de la transformation consiste à ajouter des boucles étiquetées par $\#$ sur chaque sommet et à ajouter pour chaque arc étiqueté par a , un autre arc de mêmes extrémités mais d'orientation inverse et étiqueté par \hat{a} . Pour cela, on applique la substitution inverse avec retour définie par $L_{\#} = \varepsilon$, $L_a = a$ et $L_{\hat{a}} = \bar{a}$. La figure 2.13 présente cette construction.

La deuxième étape consiste à déplier le graphe obtenu. Ensuite, il s'agit d'appliquer la restriction rationnelle définie par le langage

$$L'(\#L')^* , \quad \text{avec } L' = A^* + \hat{A}^* + \sum_{\substack{a,b \in A \\ a \neq b}} \hat{A}^* \hat{a} b A^* .$$

La dernière opération consiste à retourner les arcs étiquetés par une lettre de \hat{A} . Elle s'effectue en appliquant la substitution finie inverse avec retour définie pour tout $a \in A$ par $L_a = a + \bar{\hat{a}}$ et par $L_{\#} = \#$. \square

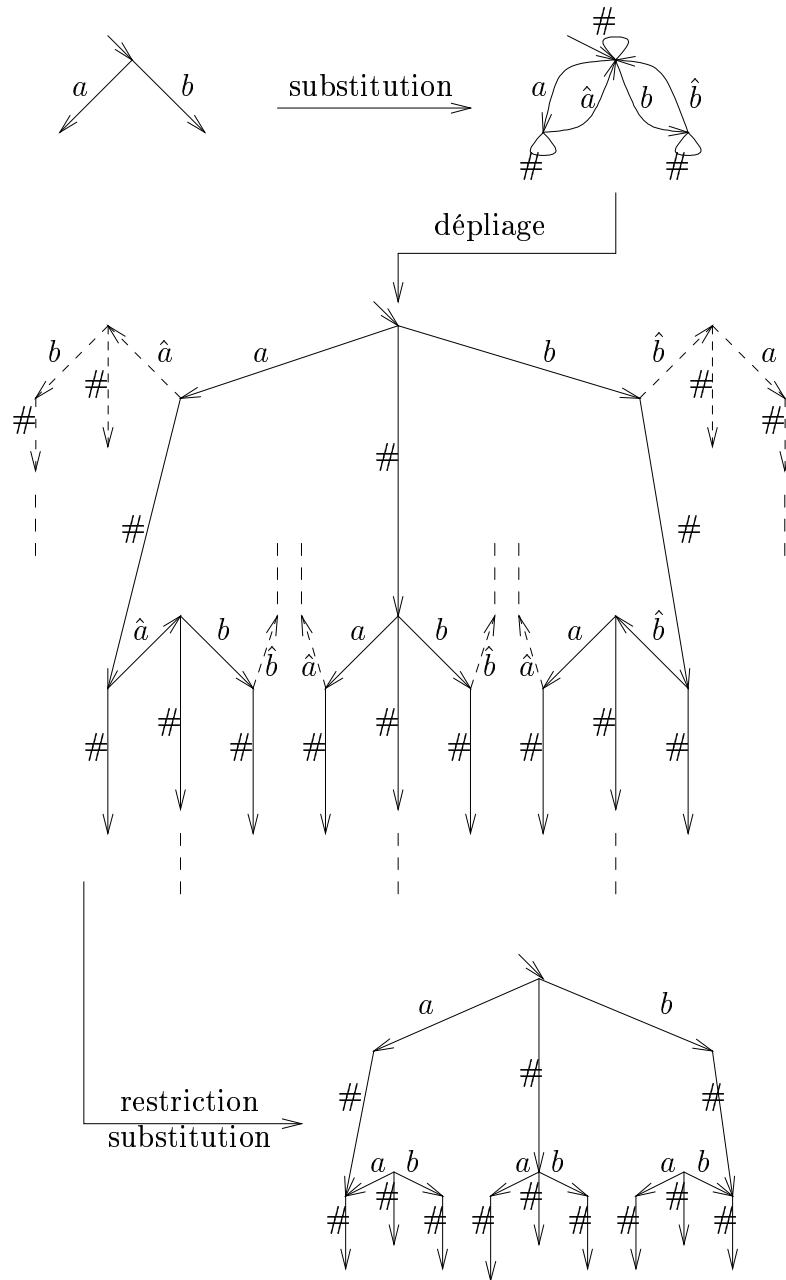


FIG. 2.13 – Exemple de structure arborescente d'arbre obtenue par dépliage.



2.4 Propriétés structurelles

Cette partie décrit des «mesures» applicables aux graphes infinis. Elles permettent d'exprimer des contraintes structurelles sur ces graphes. En particulier, si deux graphes n'ont pas la même «mesure», alors ils ne sont pas isomorphes.

Nous présentons successivement trois propriétés de ce type et les relations qu'elles entretiennent. Cette présentation n'est aucunement exhaustive.

2.4.1 La largeur arborescente

La largeur arborescente associe un entier à tout graphe (ou éventuellement ∞ pour les graphes infinis). La définition classique (pour les graphes finis) correspond à l'existence d'une certaine forme de décomposition arborescente du graphe [RS83, RS86, RS90, Die00]. Les travaux de R. Diestel donnent une nouvelle description (à base de jeu) plus simple de cette notion. Nous présentons ici cette version et son adaptation au cas des graphes infinis proposée par Christof Löding.

Le *jeu du voleur et des policiers* fini se joue sur un *graphe fini* non étiqueté, non orienté, par deux joueurs nommés *le voleur* et *la police*. Le voleur () se trouve à chaque instant du jeu sur un sommet du graphe. La police possède quant à elle k policiers () disposés eux aussi sur les sommets du graphe. Le jeu se déroule alors comme suit.

1. La police place ses k policiers sur les sommets du graphe.
2. Le voleur choisit un sommet initial où il se place.
3. La police choisit un de ses policiers, l'enlève du graphe, et choisit un sommet du graphe sur lequel se trouvera ce policier au tour du jeu suivant.
4. Le voleur se déplace autant qu'il veut en empruntant les arcs du graphe et en évitant les policiers.
5. Le policier momentanément retiré du graphe est placé sur sa destination.
6. Le jeu reprend à l'étape 3.

La police gagne si elle arrive à placer un de ses policiers sur le même sommet que le voleur. Le voleur gagne s'il ne se fait jamais attraper, le jeu ne s'arrête pas.

Un graphe fini \mathcal{G} est de *largeur arborescente* k si $k + 1$ est le nombre minimum de policiers permettant à la police de systématiquement attraper le voleur (*c.a.d.* en terme de jeu, que la police possède une stratégie gagnante). On la note $LA(\mathcal{G})$.

Exemple 2.31 *La figure 2.14 présente trois situations de jeux.*

Le premier jeu (a) se déroule sur un arbre avec deux policiers. Dans une telle configuration les policiers sont sûrs de gagner. Pour cela, à chaque tour de jeu, l'un des policiers reste immobile et isole ainsi le voleur dans un sous-arbre. L'autre policier se déplace alors jusqu'au sommet adjacent à celui de son collègue en direction du voleur (comme indiqué en pointillé). Cette tactique diminue

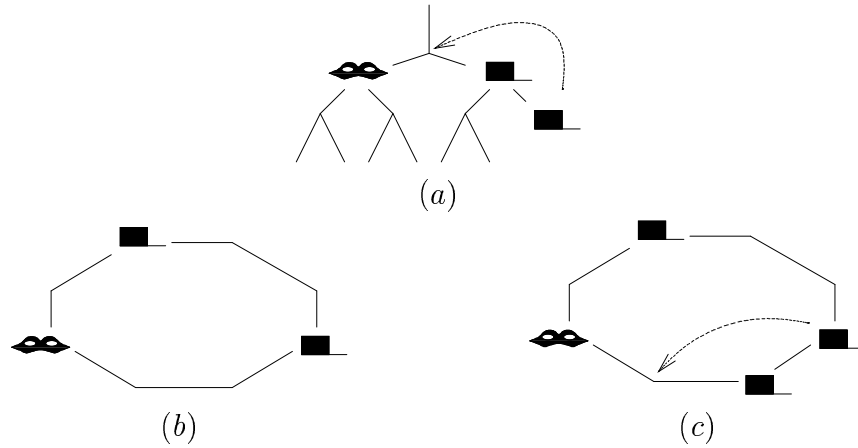


FIG. 2.14 – Jeu du voleur et des policiers sur un arbre (a), et une boucle (b et c).

l'espace dont dispose le voleur. Les deux policiers inversent alors leur rôle et recommencent ainsi jusqu'à acculer le voleur dans une feuille de l'arbre. Ainsi, la largeur arborescente d'un arbre est 1 (sauf dans le cas dégénéré d'un arbre ne contenant aucun arc, auquel cas la largeur arborescente est 0).

Le deuxième jeu (b) se déroule toujours avec deux policiers, mais sur un graphe réduit à un cycle. Cette fois-ci le voleur peut garantir sa liberté. Pour cela, il attend qu'un policier décide de se déplacer, et il se rend à un endroit quelconque où aucun des policiers ne se trouvera au tour de jeu suivant. Il peut toujours le faire car, en présence d'un seul policier, l'autre se déplaçant, tous les sommets du cycle lui sont accessibles.

En revanche, si un troisième policier arrive en renfort (c), ce dernier n'a qu'à se placer sur un sommet du cycle, et ses deux confrères, en adoptant une stratégie comparable à celle de l'arbre poussent le voleur vers lui. La police gagne.

Ainsi, ces deux derniers cas montrent que la largeur arborescente d'un cycle (d'au moins trois sommets) est 2.

Remarque 2.32 Si les policiers disposent d'une stratégie gagnante, il peuvent toujours l'utiliser avec succès après que l'on a enlevé des arcs ou des sommets du graphe. On en déduit que la largeur arborescente est croissante vis à vis de l'inclusion.

Un cas classique utile de graphe dont la largeur arborescente est connue est celui des grilles. Appelons $k \times k$ -grille étiquetée par a le graphe de sommets $[k] \times [k]$ et tel que l'interprétation de \xrightarrow{a} soit

$$\begin{aligned} & \{((n, m), (n + 1, m)) \mid n \in [k - 1], m \in [k]\} \\ & \cup \{((n, m), (n, m + 1)) \mid n \in [k], m \in [k - 1]\} . \end{aligned}$$

Propriété 2.33 *Soit $k \geq 2$ un entier, la $k \times k$ -grille est de largeur arborescente k .*

La définition de la largeur arborescente s'étend aux graphes infinis en adaptant le jeu. Deux définitions équivalentes peuvent être données (sur une idée originale de Christof Löding) :

- au début du jeu, le voleur choisit un sous-graphe fini sur lequel se déroule le jeu fini.
- le jeu se déroule sur le graphe infini en suivant les règles du jeu fini, mais, cette fois-ci, pour gagner, le voleur doit passer une infinité de fois par un certain sommet (sa planque).

Un graphe est de *largeur arborescente bornée* si il est de largeur arborescente k pour un k fini.

De ces définitions se déduisent les résultats de Thomas et Thomassen.

Propriété 2.34 (Thomassen [Tho88, Tho89]) *Soit \mathcal{G} un graphe et k un entier, alors $LA(\mathcal{G}) \leq k$ ssi pour tout sous-graphe fini \mathcal{H} de \mathcal{G} on a $LA(\mathcal{H}) \leq k$.*

Remarque 2.35 *La grille infinie n'est pas de largeur arborescente bornée, car, quel que soit le nombre de policiers, le voleur peut choisir de jouer sur une grille suffisamment grande pour gagner.*

2.4.2 Graphe uniformément creux

La notion de graphe uniformément creux permet de mesurer la densité d'arcs présents dans le graphe.

Un graphe \mathcal{G} est *uniformément k -creux* (*k -sparse*) pour k un entier positif si, pour tout sous-graphe fini $H \subseteq \mathcal{G}$ possédant n sommets et m arcs, on a $m \leq k.n$. On dit qu'un graphe est uniformément creux si il existe un entier k tel qu'il soit uniformément k -creux.

Courcelle montre que sous cette hypothèse et pour les graphes finis, une formule du second ordre gardée peut être traduite en une formule équivalente en logique monadique [Cou03].

Un cas simple de tels graphes est celui des graphes de degré borné.

Propriété 2.36 *Les graphes de degré borné sont uniformément creux.*

En particulier, la grille infinie est de degré borné, et donc est creuse. Pourtant, ce graphe n'est pas de largeur arborescente bornée. Néanmoins, ceux-ci sont aussi uniformément creux.

Propriété 2.37 *Les graphes de largeur arborescente bornée sont uniformément creux.*

Preuve. Prouvons tout d'abord le résultat pour un graphe tel que deux arcs distincts (d'étiquettes différentes) ne relient pas les mêmes sommets et qu'il n'y a pas de boucles (*).

Par contraposée. Supposons un graphe non uniformément creux, et considérons un jeu à k policiers se déroulant sur ce graphe. Comme le graphe n'est pas uniformément creux, il existe un sous-graphe fini contenant n sommets et strictement plus de $k.n$ arcs. Sur ce graphe, on enlève tous les sommets de degré inférieur à k . Cette transformation laisse le ratio arcs/sommets strictement supérieur à k . On itère le processus jusqu'à ce qu'il n'y ait plus de sommet de degré inférieur à k . Le graphe obtenu est non vide et le voleur choisit de jouer dessus. Il dispose alors d'une stratégie évidente : quand un policier décide d'aller à l'emplacement du voleur, celui-ci dispose d'au moins k arcs pour s'enfuir (grâce à (*)), et au plus $k - 1$ des sommets qu'il peut atteindre de la sorte sont gardés par un policier. Un au moins est libre et le voleur s'y rend. Le voleur gagne.

Considérons un graphe possédant n sommets, m arcs et p étiquettes. En lui enlevant toutes ses boucles (au plus $p.n$), puis en ne conservant qu'un arc entre toute paire de sommets reliés (un arc est gardé sur au plus $2.p$ arcs), on obtient un graphe satisfaisant (*), possédant n sommets et au moins $(m - pn)/(2.p)$ arcs. Le ratio arcs/sommets ainsi obtenu tend vers l'infini quand m/n tend vers l'infini. Dans un graphe non uniformément creux, il est donc possible de trouver des sous-graphes satisfaisant (*) et ayant un ratio arcs/sommets aussi grand que voulu. \square

2.4.3 Sous-graphes bipartis complets

La présence de sous-graphes bipartis complets permet une autre mesure des graphes.

Soit k un entier naturel, le *graphe biparti complet* étiqueté par a de taille k est le graphe de sommets $V = [2] \times [k]$ possédant un arc étiqueté par a entre $(1, n)$ et $(2, m)$ pour tout $n \in [k]$ et tout $m \in [k]$.

La première propriété nous indique que l'hypothèse de ne pas contenir de graphe biparti complet aussi grand que possible est moins restrictif pour un graphe que d'être creux.

Propriété 2.38 *Si un graphe est k -creux, alors il existe l tel qu'il ne contient pas de graphe biparti complet de taille l .*

Par contre, ces deux notions ne sont pas équivalentes comme le montre la propriété suivante.

Propriété 2.39 *Il existe un graphe non-creux ne possédant pas de sous-graphe biparti complet de taille 2.*

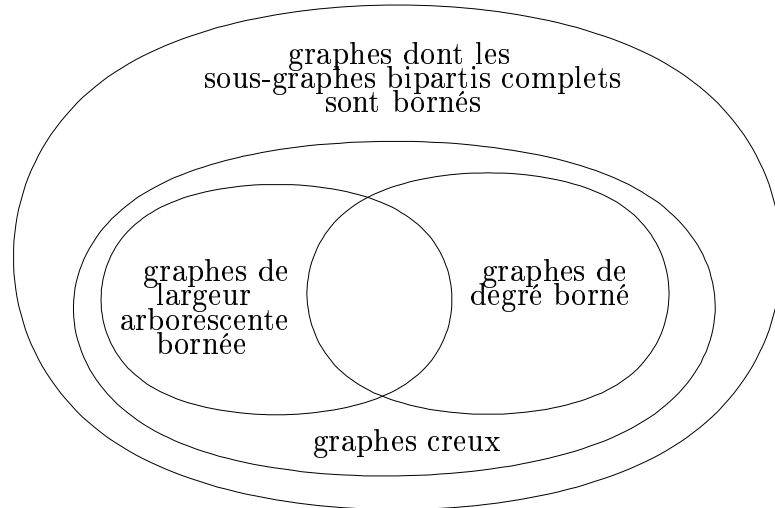


FIG. 2.15 – Propriétés structurelles de graphes et inclusions mutuelles.

Preuve. Considérons le graphe \mathcal{H} de sommets \mathbb{N} et dont les arcs étiquetés par a sont définis par

$$\xrightarrow[a]{\mathcal{H}} = \{(n, n + 2^k) \mid n \in \mathbb{N}, k \in \mathbb{N}\} .$$

Ce graphe n'est pas creux car pour k fixé, le graphe induit par les sommets de $[0, 2^k - 1]$ contient $(k - 1)2^k + 1$ arcs pour 2^k sommets et le rapport de ces deux quantités n'est pas borné.

Il s'agit de montrer qu'il n'existe pas de sous-graphe de \mathcal{H} de la forme

$$\mathcal{M} = \begin{array}{ccc} m & \longrightarrow & p \\ & \searrow & \nearrow \\ n & \longrightarrow & q \end{array}$$

On a par définition $p = m + 2^k = n + 2^{k'}$ et $q = m + 2^l = n + 2^{l'}$ pour k, l, k' et l' entiers naturels. Supposons sans perte de généralité que $m < n$. Alors, le calcul de $n - m$ nous donne que $2^{k'}(2^{k-k'} - 1) = 2^{l'}(2^{l-l'} - 1)$ avec $k', k - k', l'$ et $l - l'$ positifs (car $m < n$). On en déduit que $k' = l'$ et $k = l$ et donc $p = q$. Le motif \mathcal{M} n'est donc pas un sous-graphe de \mathcal{H} . \square

Ces différents résultats nous donnent donc le tableau de la figure 2.15.

Chapitre 3

Systèmes d'équations

La notion de système d'équations apparaît très tôt en informatique comme un outil puissant de description d'objets au même titre que les automates. Ainsi Mezei et Wright présentent-ils dès le milieu des années 70 les notions « *d'équationnalité* » et de « *reconnaissabilité* » comme propices à une généralisation aux algèbres quelconques des travaux sur les mots et sur les termes [MW67]. Ces premiers travaux sur les systèmes d'équations s'intéressaient à la représentation d'ensembles d'objets finis (les langages de mots finis ou de termes finis par exemple, ou plus généralement d'éléments d'une algèbre de type fini), et le processus de génération consistait à saturer ces ensembles par un nombre fini d'équations.

Concernant le travail présent, il convient d'utiliser une théorie similaire pour la description de structures infinies. On utilise pour cela la théorie des ordres partiels complets (en fait des ω -ordres partiels complets). En effet, la notion d'ordre partiel complet permet de définir une forme de topologie sur des familles d'objets et d'attribuer un sens aux notions de limites et de continuité. Le résultat fondamental dû à Tarski [Tar55] nous enseigne que, dans ce formalisme, toutes les applications continues admettent un unique plus petit point fixe. C'est cette propriété qui permet de donner un sens aux systèmes d'équations. Dans le cas particulier étudié par Mezei et Wright, la notion de limite coïncide avec la terminaison du processus de saturation.

On étudie alors successivement deux outils permettant de travailler sur les systèmes d'équations. Le premier de ces outils est le μ -calcul. Il s'agit d'une théorie qui dépasse largement le cadre de ce mémoire et seule une version restreinte en est présentée. De notre point de vue, le μ -calcul est avant tout un formalisme puissant et concis permettant de décrire des ensembles de termes infinis.

Le second outil permet de transformer les termes. Il s'agit des transducteurs descendants déterministes avec anticipation μ -définissable. Comme il n'y a pas d'ambiguïté dans le présent contexte, le terme de transducteur déterministe est utilisé. Là encore, les transducteurs déterministes peuvent être exécutés directe-

ment sur les systèmes d'équations: ils correspondent aux transductions monadiques de systèmes d'équations.

Le reste de ce paragraphe est construit comme suit. La partie 3.1 présente la notion d'ordre partiel complet et d'algèbre continue. La partie 3.2 introduit les systèmes d'équations et fournit les premières propriétés les concernant. La partie 3.3 est consacrée à la notion de μ -définissabilité. La partie 3.4 présente quant à elle les transducteurs déterministes.

3.1 Ordres partiels complets et algèbres

L'outil que nous avons choisi ici pour définir la notion de limite est celui des ordres partiels complets. L'utilisation de catégories ω -complètes est une autre possibilité de formalisme que nous n'avons pas adoptée ici.

Ce paragraphe commence par la présentation de la notion d'ordre partiel complet. Puis quelques exemples utiles sont donnés. Enfin ces ensembles ordonnés sont munis d'opérateurs pour former des algèbres continues.

3.1.1 Définitions

Étant donné un ensemble E ordonné, on appelle *chaîne ascendante* (on dit parfois simplement *chaîne*) toute suite croissante, indicée par \mathbb{N} , d'éléments de E . Un *majorant* d'une chaîne ascendante est un élément de E supérieur à tous les éléments de la chaîne. La *borne supérieure* d'une chaîne ascendante u (si elle existe) est le plus petit majorant des éléments de la chaîne. Elle est aussi appelée *limite* de la chaîne ascendante.

Étant donné un ensemble E ordonné par \sqsubseteq , le couple (E, \sqsubseteq) est un *ordre partiel complet* (cpo pour "complete partial order" en anglais) si les deux conditions suivantes sont satisfaites :

- il existe dans E un plus petit élément pour \sqsubseteq (généralement noté \perp),
- toute chaîne ascendante u admet une borne supérieure dans E (généralement notée $\sqcup u$ ou encore $\sqcup_{n \in \mathbb{N}} u_n$).

Afin de rendre explicite le plus petit élément et la borne supérieure, on fait aussi référence au cpo $(E, \sqsubseteq, \perp, \sqcup)$.

Exemple 3.1 Les entiers naturels \mathbb{N} munis de l'ordre usuel \leq ne forment pas un cpo. En effet, la chaîne $(n)_{n \in \mathbb{N}}$ n'est majorée par aucun entier, et donc n'admet pas de borne supérieure.

Exemple 3.2 Les entiers naturels augmentés d'un symbole infini $\mathbb{N} \uplus \{\infty\}$ et munis de l'ordre usuel \leq étendus pour tout n par $n \leq \infty$ est un cpo. Son plus

petit élément est 0, et étant donné une chaîne ascendante u ,

- si u est majorée par un entier, alors u est constante à partir d'un certain rang et cette constante est sa borne supérieure,
- si u n'est pas bornée par un entier naturel, alors sa borne supérieure est ∞ .

Les propriétés 3.3, 3.4, 3.5, 3.6 et 3.7 montrent comment les opérations ensemblistes usuelles peuvent être utilisées pour construire des cpos.

Propriété 3.3 (cpo fini) *Tout ensemble fini équipé d'un ordre partiel et possédant un plus petit élément est un cpo.*

Propriété 3.4 (cpo pointé) *Pour E un ensemble, l'ensemble $E_{\perp} = E \uplus \{\perp\}$ ordonné par $u \sqsubseteq u'$ si $u = \perp$ ou $u = u'$ est un cpo appelé cpo pointé sur E .*

Propriété 3.5 (cpo des parties) *Les parties d'un ensemble, ordonnées par l'inclusion, forment un cpo dont le plus petit élément est l'ensemble vide et la borne supérieure est l'union ensembliste.*

Propriété 3.6 (cpo produit) *Étant donnés deux cpo $(E_1, \sqsubseteq_1, \perp_1, \sqcup_1)$ et $(E_2, \sqsubseteq_2, \perp_2, \sqcup_2)$, l'ensemble produit $E_1 \times E_2$ est un cpo quand il est ordonné par $\sqsubseteq_{E_1 \times E_2}$ défini comme suit :*

$$(x_1, x_2) \sqsubseteq_{E_1 \times E_2} (y_1, y_2) \quad \text{si} \quad x_1 \sqsubseteq_1 y_1 \quad \text{et} \quad x_2 \sqsubseteq_2 y_2 .$$

La borne supérieure d'une chaîne ascendante $((u_n, v_n)_{n \in \mathbb{N}})$ est $(\sqcup_1 u, \sqcup_2 v)$ et le plus petit élément est (\perp_1, \perp_2) .

Propriété 3.7 (cpo des applications) *Étant donné un ensemble D et un cpo $(E, \sqsubseteq, \perp, \sqcup)$, l'ensemble des applications de D dans E est un cpo quand il est ordonné par \sqsubseteq' définie par :*

$$f \sqsubseteq' g \quad \text{si pour tout } x \in D, \quad f(x) \sqsubseteq g(x) .$$

Le plus petit élément est l'application constante égale à \perp . La limite d'une chaîne ascendante f est alors $\sqcup' f$ définie par :

$$\text{pour tout } x \text{ de } D, \quad (\sqcup' f)(x) = \bigsqcup_{n \in \mathbb{N}} (f_n(x)) .$$

Remarque 3.8 (cpo des fonctions) *Soient E et F deux ensembles. L'ensemble des applications de E dans F_{\perp} est un cpo (propriétés 3.4 et 3.7). Cet ensemble d'applications coïncide avec l'ensemble des fonctions de E dans F , l'élément \perp s'interprétant comme indéfini. Le cpo ainsi défini correspond à l'ordre de «prolongement» entre fonctions, c.a.d $f \sqsubseteq g$ si le domaine de f est contenu dans le domaine de g et si sur le domaine de f les deux fonctions coïncident.*

Étant donnés deux cpo $(E, \sqsubseteq, \perp, \sqcup)$ et $(E', \sqsubseteq', \perp', \sqcup')$, une application f de E dans E' est *continue* si elle commute avec la limite, c'est à dire que, pour toute chaîne ascendante u de E , $(f(u_n))_{n \in \mathbb{N}}$ est une chaîne ascendante (notée $f(u)$), de borne supérieure

$$\sqcup' f(u) = f(\sqcup u) .$$

Remarque 3.9 (continuité et croissance) *Toute application continue est croissante (c'est à dire $f(x) \sqsubseteq' f(y)$ pour tout $x \sqsubseteq y$) mais la réciproque est fautive dans le cas général¹.*

Considérons l'ensemble $E = \mathbb{N} \cup \{0', 1'\}$ ordonné totalement par $0 \sqsubseteq 1 \sqsubseteq \dots \sqsubseteq 0' \sqsubseteq 1'$. On vérifie aisément qu'il s'agit d'un cpo. Considérons maintenant l'application f de E dans E définie par :

$$\text{pour tout } n \text{ de } \mathbb{N}, f(n) = n, \quad \text{et } f(0') = f(1') = 1' .$$

Cette application est croissante sans être continue :

$$f\left(\bigsqcup_{n \in \mathbb{N}} n\right) = f(0') = 1' \neq 0' = \bigsqcup_{n \in \mathbb{N}} n = \bigsqcup_{n \in \mathbb{N}} f(n) .$$

Le résultat fondamental qui justifie l'introduction des applications continues est le suivant.

Théorème 7 (Knaster-Tarski [Tar55]) *Toute application continue d'un cpo $(E, \sqsubseteq, \perp, \sqcup)$ dans lui-même admet un plus petit point fixe $\text{fix}(f)$ défini par :*

$$\text{fix}(f) = \min\{x \mid f(x) = x\} = \min\{x \mid f(x) \sqsubseteq x\} = \bigsqcup_{n \in \mathbb{N}} f^n(\perp) .$$

Preuve. Remarquons tout d'abord que $f^n(\perp)$ est une chaîne ascendante : il suffit pour cela de montrer par récurrence sur n que $f^n(\perp) \sqsubseteq f^{n+1}(\perp)$ en utilisant la croissance de f .

Montrons maintenant qu'il s'agit d'un point fixe :

$$\begin{aligned} f\left(\bigsqcup_{n \in \mathbb{N}} f^n(\perp)\right) &= \bigsqcup_{n \in \mathbb{N}} f\left(f^n(\perp)\right) && \text{(par continuité)} \\ &= \bigsqcup_{n \in \mathbb{N}} f^{n+1}(\perp) . \end{aligned}$$

Il reste à montrer que tout point fixe lui est supérieur. Soit c tel que $f(c) = c$. On montre par récurrence et croissance de f que pour tout n , $f^n(\perp) \sqsubseteq c$. Par passage à la limite, on en déduit la propriété souhaitée. \square

Les propriétés 3.10, 3.11, 3.12, et 3.13 montrent que les constructions usuelles engendrent des fonctions continues.

Propriété 3.10 *Toute application constante est continue.*

1. Elle reste vraie quand l'ensemble d'arrivée est fini.

Propriété 3.11 *La composition de deux applications continues est une application continue.*

Propriété 3.12 (continuité dans le cpo produit)

Étant donné des cpos (E, \sqsubseteq) , (E_1, \sqsubseteq_1) et (E_2, \sqsubseteq_2) ,

- *la projection sur la première composante π_1 de $E_1 \times E_2$ dans E_1 définie par $\pi_1(x, y) = x$ est continue,*
- *la projection sur la seconde composante π_2 de $E_1 \times E_2$ dans E_2 définie par $\pi_2(x, y) = y$ est continue,*
- *pour deux applications continues f et g respectivement de E dans E_1 et de E dans E_2 , l'application $f \times g$ de E dans $E_1 \times E_2$ définie par $(f \times g)(x) = (f(x), g(x))$ est continue.*

Propriété 3.13 (continuité dans le cpo des applications)

Considérons un ensemble D et deux cpos (E, \sqsubseteq) , (E', \sqsubseteq') alors,

- *pour tout x de D , l'application «eval _{d} » de E^D dans E qui à f associe $f(x)$ est continue,*
- *étant donnée pour tout d de D une application f_d de E dans E' , l'application de E dans E'^D qui à x associe l'application qui à d associe $f_d(x)$ est continue².*

La combinaison de toutes ces propriétés permet de donner un sens à la continuité des applications de plusieurs variables. En fait, toutes les combinaisons «naturelles» de telles applications sont aussi continues.

Une définition naturelle d'un morphisme précise que celui-ci est continu et préserve le plus petit élément. Un tel morphisme préserve la notion de point fixe.

Propriété 3.14 (les morphismes de cpos préservent les points fixes) *Soient deux cpos E et F et η une application continue de E dans F satisfaisant $\eta(\perp) = \perp$. Pour toutes applications continues f et g respectivement de E dans E et de F dans F , si $\eta \circ f = g \circ \eta$ alors $\eta(\text{fix}(f)) = \text{fix}(g)$.*

Propriété 3.15 (continuité à la limite) *Soient D et E des cpos, et f une chaîne ascendante d'applications continues de D dans E , alors la borne supérieure de f est une application continue de D dans E .*

Donnons pour finir une propriété permettant de construire des cpos à partir de fonctions continues.

2. Une manière plus intuitive de dire la même chose consiste à revenir à la définition de la continuité et à utiliser la notation d'abstraction λ : pour toute chaîne ascendante u , $\lambda d.f_d(u)$ est une chaîne ascendante et $\sqcup(\lambda d.f_d(u)) = \lambda d.f_d(\sqcup u)$. On peut ainsi dire que la borne supérieure «commute» avec le λ .

Propriété 3.16 (cpo des points fixes) Soit (E, \sqsubseteq) un cpo et f une application continue de E dans E , alors l'ensemble des points fixes de f munis de l'ordre \sqsubseteq est un cpo. La borne supérieure de ce cpo coïncide avec celle de E restreinte aux points fixes de f , mais attention, le plus petit élément est le plus petit point fixe de f et n'est pas en général le plus petit élément de (E, \sqsubseteq) .

Si de plus f satisfait pour tout $x \in E$, $x \sqsubseteq f(x)$, alors, pour tout élément $x \in E$, il existe un plus petit point fixe $f^*(x)$ de f qui lui soit supérieur, et l'application f^* est continue.

Une application de ce résultat concerne les relations binaires sur un ensemble F . Ces relations forment un cpo pour l'inclusion (les parties du produit $F \times F$). Considérons maintenant l'application des relations binaires dans les relations binaires qui à R associe $R \cup RR \cup R^{-1}$ (où $RR(x, z)$ si $R(x, y)$ et $R(y, z)$ pour un certain $z \in F$ et $R^{-1}(x, y)$ si $R(y, x)$). Elle est continue, et donc l'ensemble de ses points fixes forme un cpo (propriété 3.16). Or, les points fixes de cette application sont les relations d'équivalences. Enfin, la propriété 3.16 nous enseigne que l'application associant à chaque relation sa clôture réflexive symétrique et transitive est continue. Bien entendu, un raisonnement similaire fonctionne pour toute autre forme de clôture.

3.1.2 Quelques ordres partiels complets

Nous avons déjà présenté au chapitre précédent la théorie des cpos. Il s'agit ici, de manière plus pragmatique, de l'instancier au travers des constructions dont nous aurons l'usage dans le reste de ce mémoire.

Cpo des langages

Un *langage de termes* est un ensemble de termes finis. De même, un *langage de mots* est un ensemble de mots finis. L'ensemble des langages de termes (resp. de mots) ordonné par l'inclusion forme un cpo d'après la propriété 3.5. Ils correspondent au contexte des travaux de Mezei et Wright. Le plus petit élément de ces cpos est le langage vide. La borne supérieure coïncide avec l'union ensembliste.

On dispose alors des fonctions continues suivantes :

- l'union est une application continue,
- pour f une application n -aire des termes dans les termes (resp. des mots dans les mots), alors son extension aux langages est continue, *c.a.d.*

$$f(L_1, \dots, L_n) = \{f(t_1, \dots, t_n) \mid t_1 \in L_1, \dots, t_n \in L_n\} .$$

Cpo des structures et des graphes

Les structures relationnelles sont munies de l'ordre de l'inclusion : étant données deux structures \mathcal{S} et \mathcal{S}' , $\mathcal{S} \subseteq \mathcal{S}'$ si $\mathcal{U}_{\mathcal{S}} \subseteq \mathcal{U}_{\mathcal{S}'}$ et pour tout symbole relationnel

$R, R^S \subseteq R^{S'}$. Les structures³ munies de l'ordre partiel \subseteq forment un cpo (il s'agit de nouveau d'une variante de la propriété 3.5). Le plus petit élément de ce cpo est la structure vide qui a pour univers \emptyset et pour interprétation de toute relation \emptyset .

Le cpo des graphes n'est qu'une instantiation du cpo des structures aux graphes. La même notation \subseteq est utilisée.

Une variante du cpo des structures est le *cpo des structures avec quotient*. Il s'agit en fait du même cpo, si ce n'est que l'on utilise la relation \sim et que celle-ci doit-être une congruence (voir d'écriture des structures avec quotient). Il s'agit bien d'un cpo d'après la propriété 3.16. La relation d'ordre est \subseteq , le plus petit élément est la structure vide et la borne supérieure est l'opération \cup sur les structures.

Remarquons que l'application qui a une structure avec quotient associe sa structure quotient n'est pas continue.

Cpo des termes infinis

Un autre cpo fondamental dans ce travail est celui des termes infinis. Considérons un ensemble de symboles pointés \mathcal{F}_\perp . Rappelons qu'il existe une bijection entre les termes et leur langage de branches (propriété 2.5). Cette bijection reste valide si l'on restreint le langage aux branches sans le symbole \perp . C'est cette caractérisation qui permet d'exprimer l'ordre sur les cpos de la manière la plus compacte.

Ainsi, considérons sur les termes infinis de $\mathcal{T}^\omega(\mathcal{F}_\perp)$, la relation d'ordre \sqsubseteq définie par :

$$t \sqsubseteq t' \quad \text{si} \quad \mathcal{LB}(t) \cap (\mathcal{F} + \mathbb{N})^* \subseteq \mathcal{LB}(t').$$

Cette relation s'interprète comme “ t est obtenu en remplaçant dans t' certains sous-termes (éventuellement une infinité) par \perp ”. Par exemple, $\perp \sqsubseteq f(\perp, \perp) \sqsubseteq f(\perp, a(b(\perp))) \sqsubseteq f(c, a(b(c)))$. Les termes de $\mathcal{T}^\omega(\mathcal{F}_\perp)$ ordonnés par cette relation forment un cpo :

- le plus petit élément est \perp ,
- la borne supérieure s'obtient en effectuant l'union des langages de branches et en reconstruisant un terme à partir de ce langage.

Une propriété de ce cpo est que tout terme peut être décrit comme la limite d'une suite de termes finis.

Plus précisément, on construit par récurrence sur n pour tout \mathcal{F}_\perp -terme t (éventuellement infini) le terme fini $t \downarrow_n$ qui est identique à t si ce n'est que tous

3. Pour être rigoureux, les structures ne forment pas à strictement parler un ensemble et donc un cpo. Pour cela, il faudrait se fixer un univers global. Nous n'entrons pas ici dans ces détails qui obscurcissent le propos et les notations sans pour autant correspondre à une véritable difficulté technique.

les sous-termes situés à la profondeur n sont «élagués» et remplacés par \perp :

$$t \downarrow_n = \begin{cases} \perp & \text{si } n = 0, \\ f(t_1 \downarrow_{n-1}, \dots, t_{|f|} \downarrow_{n-1}) & \text{si } n > 0 \text{ et } t = f(t_1, \dots, t_{|f|}). \end{cases}$$

On a alors la propriété suivante.

Propriété 3.17 *Pour tout terme t de $\mathcal{T}^\omega(\mathcal{F}_\perp)$,*

$$\bigsqcup_{n \in \mathbb{N}} t \downarrow_n = t.$$

Ce résultat permet ainsi de décrire des applications sur les termes infinis par prolongement d'applications sur les termes finis.

Propriété 3.18 *Soit f une application croissante de $\mathcal{T}(\mathcal{F}_\perp)$ dans un cpo E . Alors, il existe une unique application continue de $\mathcal{T}^\omega(\mathcal{F}_\perp)$ dans E qui coïncide avec f sur $\mathcal{T}(\mathcal{F}_\perp)$.*

3.1.3 Algèbres et algèbres continues

Étant donné un alphabet gradué \mathcal{F} , une \mathcal{F} -algèbre est un ensemble \mathbf{A} muni d'une application $\text{sem}[\mathbf{A}]$ qui associe à chaque symbole f de \mathcal{F} une application de $\mathbf{A}^{|f|}$ dans \mathbf{A} . Les symboles de \mathcal{F} sont appelés des *opérateurs* et $\text{sem}[\mathbf{A}]$ est l'application de *sémantique*. Pour cette raison, on parle aussi de domaines sémantiques au lieu d'algèbres. Les opérateurs d'arité 0 sont appelés *opérateurs constants*. L'intérêt de cette définition est qu'il est possible de considérer plusieurs \mathcal{F} -algèbres sur des ensembles différents mais partageant la même dénomination des opérateurs. Le plus souvent, le contexte permet de déterminer l'application de sémantique sans ambiguïté. Dans un tel cas, pour un symbole f et $e_1, \dots, e_{|f|}$ des éléments de \mathcal{F} , on notera simplement $f(e_1, \dots, e_{|f|})$ au lieu de $\text{sem}[\mathbf{A}](f)(e_1, \dots, e_{|f|})$.

Un *morphisme d'algèbre* d'une \mathcal{F} -algèbres \mathbf{A} dans une \mathcal{F} -algèbre \mathbf{A}' est une application ϕ de \mathbf{A} dans \mathbf{A}' telle que pour tout $f \in \mathcal{F}$ et $e_1, \dots, e_{|f|} \in \mathbf{A}$,

$$\phi(f(e_1, \dots, e_{|f|})) = f(\phi(e_1), \dots, \phi(e_{|f|})).$$

Une \mathcal{F} -algèbre \mathbf{A} est *libre* si pour tous symboles f et f' de \mathcal{F} et tous éléments $e_1, \dots, e_{|f|}, e'_1, \dots, e'_{|f'|}$ de \mathbf{A} ,

$$\text{si } f(e_1, \dots, e_{|f|}) = f'(e'_1, \dots, e'_{|f'|}) \quad \text{alors } f = f' \text{ et pour tout } i, \quad e_i = e'_i.$$

Propriété 3.19 *Pour \mathcal{F} un alphabet gradué, l'ensemble des termes $\mathcal{T}^\omega(\mathcal{F})$ (resp. des termes finis $\mathcal{T}(\mathcal{F})$) est une algèbre libre.*

Pour un alphabet gradué \mathcal{F} et une \mathcal{F} -algèbre \mathbf{A} , il existe un et un seul morphisme d'algèbre de $\mathcal{T}(\mathcal{F})$ dans \mathbf{A} . Il est appelé *morphisme canonique* et l'image d'un terme t par ce morphisme est notée $\llbracket t \rrbracket_{\mathbf{A}}$. Ce morphisme associe à une expression (*c.a.d.* un terme) sa *valeur* dans l'algèbre. On note $\llbracket t \rrbracket$ à la place de $\llbracket t \rrbracket_{\mathbf{A}}$ quand il n'y a pas d'ambiguïté sur l'algèbre concernée.

Une \mathcal{F} -algèbre continue \mathbf{A} est une \mathcal{F} -algèbre munie d'un ordre partiel complet telle que pour tout opérateur $f \in \mathcal{F}$ l'application $\text{sem}[\mathbf{A}](f)$ est continue. Un *morphisme d'algèbres continues* est un morphisme continu d'algèbres continues qui préserve le plus petit élément. Bien entendu, la composition de deux morphismes d'algèbres continues est encore un morphisme d'algèbres continues. L'ensemble des termes de $\mathcal{T}^\omega(\mathcal{F}_\perp)$ forme une \mathcal{F} -algèbre continue.

Étant donnée une \mathcal{F} -algèbre continue \mathbf{A} , il existe un unique morphisme d'algèbres continues de $\mathcal{T}^\omega(\mathcal{F}_\perp)$ dans \mathbf{A} . Ce morphisme coïncide avec le morphisme canonique sur les termes finis. Il est ensuite prolongé par continuité de manière unique grâce à la propriété 3.18. Naturellement, comme pour les algèbres, ce morphisme est appelé *morphisme canonique* et l'image d'un terme t par ce morphisme est sa *valeur*, notée $\llbracket t \rrbracket_{\mathbf{A}}$. On utilise de même la version abrégée $\llbracket t \rrbracket$ lorsqu'il n'y a pas d'ambiguïté sur l'algèbre continue correspondante.

Exemple 3.20 *Considérons l'alphabet gradué $\mathcal{F}_N = \{0 : 0, 1 : 0, \infty : 0, + : 2, * : 2\}$. On utilise les notations infixes $t + t'$ et $t * t'$ au lieu de $+(t, t')$ et $*(t, t')$ (ainsi que les priorités habituelles sur ces opérateurs).*

Soit donc N la \mathcal{F}_N -algèbre de domaine $\mathbb{N} \cup \{\infty\}$ dont la sémantique est donnée par

$$\begin{aligned} \text{sem}[N](0) &= 0, & \text{sem}[N](1) &= 1, & \text{sem}[N](\infty) &= \infty, \\ \text{sem}[N](+)(n, n') &= \begin{cases} n + n' & \text{si } n \in \mathbb{N} \text{ et } n' \in \mathbb{N}, \\ \infty & \text{sinon,} \end{cases} \\ \text{sem}[N](*)(n, n') &= \begin{cases} nn' & \text{si } n \in \mathbb{N} \text{ et } n' \in \mathbb{N}, \\ 0 & \text{si } n = 0 \text{ ou } n' = 0, \\ \infty & \text{sinon.} \end{cases} \end{aligned}$$

*Comme vu précédemment, $\mathbb{N} \cup \{\infty\}$ muni de l'ordre usuel \leq forme un cpo. Pour cet ordre, les sémantiques des opérateurs sont continues (en particulier, on a posé $\infty * 0 = 0 * \infty = 0$ afin d'assurer cette continuité). L'algèbre N est donc une algèbre continue.*

3.2 Systèmes d'équations

3.2.1 Une première définition

- Un \mathcal{F} -système d'équations est un triplet (X, x_0, Eq) où
- X est un ensemble (éventuellement infini) de variables, identifiable à de nouveaux symboles d'arité 0 (X et \mathcal{F} sont disjoints),
 - $x_0 \in X$ est la variable résultat,
 - Eq est une application qui à chaque variable x associe un terme de $\mathcal{T}(\mathcal{F} \uplus X)$; on utilise la syntaxe $x = Eq(x)$ qui est appelée *équation*.

Il est important de noter qu'aucune contrainte de finitude n'est imposée.

Pour attribuer une unique solution à un \mathcal{F} -système d'équations, on l'évalue dans une \mathcal{F} -algèbre continue. Considérons donc un \mathcal{F} -système d'équations $S = (X, x_0, Eq)$ et une \mathcal{F} -algèbre continue \mathbf{A} . On appelle *valuation* du système une application ρ de X dans \mathbf{A} . L'ensemble des valuations est muni d'une structure de cpo d'après la propriété 3.7. Pour ρ une valuation, $\mathbf{A}\{\rho\}$ dénote la $\mathcal{F} \uplus X$ -algèbre dont la sémantique coïncide avec celle de \mathbf{A} sur \mathcal{F} , et est égale à ρ sur X . Soit alors $F[\mathbf{A}, S]$ l'application des valuations dans elles-mêmes définie par :

$$F[\mathbf{A}, S](\rho) = \rho' \quad \text{définie pour tout } x \text{ de } X \quad \text{par } \rho'(x) = \llbracket Eq(x) \rrbracket_{\mathbf{A}\{\rho\}} .$$

L'application $F[\mathbf{A}, S]$ s'interprète comme une étape de propagation des calculs dans le système. Les propriétés 3.11, 3.12 et 3.13 montrent alors la continuité de $F[\mathbf{A}, S]$. Cette application admet donc un unique plus petit point fixe (théorème 7) et il est par conséquent valide de définir la *solution* de S dans \mathbf{A} comme la valeur de la solution de $F[\mathbf{A}, S]$ à la racine :

$$Solution[\mathbf{A}](S) = (\text{fix}(F[\mathbf{A}, S]))(x_0) .$$

Bien souvent, il n'est pas ambigu de ne pas préciser l'algèbre continue \mathbf{A} . On note alors simplement $Solution(S)$ au lieu de $Solution[\mathbf{A}](S)$.

La propriété suivante montre que la solution d'un \mathcal{F} -système d'équations est préservée par morphisme d'algèbres continues.

Propriété 3.21 *Soient deux \mathcal{F} -algèbres continues \mathbf{A} et \mathbf{A}' , et ϕ un morphisme d'algèbres continues de \mathbf{A} dans \mathbf{A}' . Pour tout \mathcal{F} -système d'équations S , on a*

$$\phi(Solution[\mathbf{A}](S)) = Solution[\mathbf{A}'](S) .$$

Preuve. Conséquence directe de la propriété 3.14. □

Le corollaire suivant montre qu'il est suffisant de résoudre les systèmes d'équations dans les termes infinis puis d'envoyer la solution par morphisme canonique dans l'algèbre continue voulue.

Corollaire 3.22 *Pour tout alphabet gradué \mathcal{F} , toute \mathcal{F} -algèbre continue \mathbf{A} et tout \mathcal{F} -système d'équations S ,*

$$\text{Solution}[\mathbf{A}](S) = \llbracket \text{Solution}[\mathcal{T}^\omega(\mathcal{F})](S) \rrbracket_{\mathbf{A}} .$$

Remarque 3.23 *Dans certains contextes, il est utile de pouvoir utiliser plusieurs algèbres continues et des systèmes d'équations mettant en jeu des variables dont la valuation appartient à l'une ou l'autre de ces algèbres. Ce besoin coïncide avec une vision communément admise de la notion de type : chaque valeur possède un type qui définit les différentes opérations dans lesquelles elle est susceptible d'être utilisée.*

Le modèle de systèmes d'équations décrit jusqu'à présent ne possède pas cette capacité. Il est pourtant possible de la simuler en utilisant une algèbre contenant l'union disjointe de toutes les algèbres utilisées dans le système d'équations. Plus précisément, une algèbre continue sortée est décrite par la donnée :

- d'une famille de cpos $(E_i)_{i \in I}$,
- un alphabet \mathcal{F} ,
- d'une application de sorte ρ de \mathcal{F} dans $I^* \times I$,
- et une application sémantique ρ qui à chaque symbole f de \mathcal{F} de type $\rho(f) = (\tau_1 \dots \tau_n, \tau)$ associe une application de $E_{\tau_1} \times \dots \times E_{\tau_n}$ dans E_τ .

Le reste des définitions et des propriétés n'est pas modifié par cette extension.

3.2.2 Systèmes d'équations comme graphes

Pour représenter un système d'équations comme un graphe, on se restreint aux *systèmes d'équations normalisés*, c'est-à-dire ceux dont les équations sont de la forme $x = f(x_1, \dots, x_{|f|})$ où f est un symbole et $x_1, \dots, x_{|f|}$ sont des variables du système d'équations.

Normaliser un système consiste à produire un système normalisé dont la solution est la même que celle du système original. Il est toujours possible de normaliser un système d'équations pourvu que l'on dispose d'une représentation de \perp comme solution d'un système d'équations. En effet, le processus de normalisation consiste à introduire des variables intermédiaires pour décomposer les termes de hauteur supérieure à deux en termes de hauteur un. Le problème intervient lorsqu'un terme de hauteur 0 apparaît dans le système comme par exemple dans l'équation $x = x$. La plus petite solution de cette équation étant \perp , il faut disposer d'une représentation normalisée de \perp pour normaliser ce système.

Pour représenter un système d'équations normalisé comme un graphe, on utilise des conventions similaires à celles mises en jeu dans la définition des termes (voir page 14). Ainsi, un graphe enraciné *représente un \mathcal{F} -système d'équations*

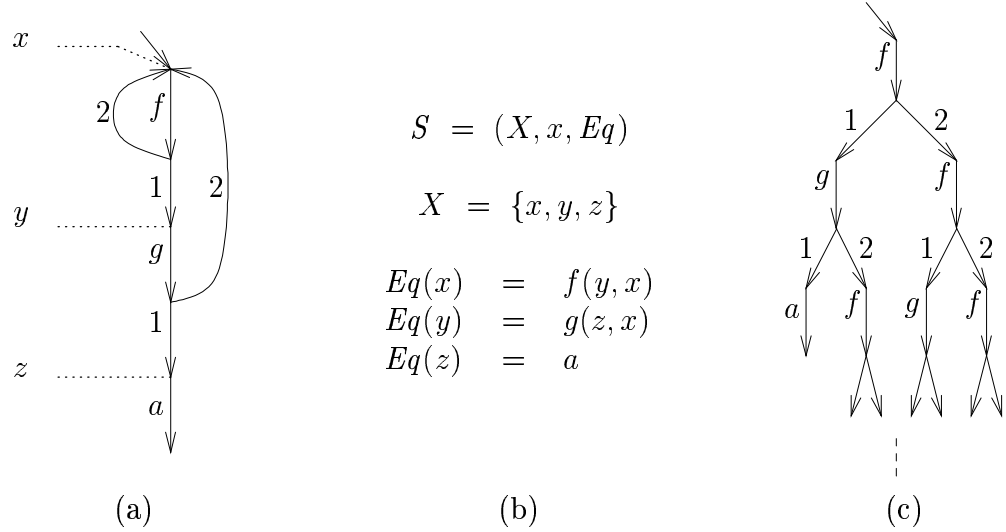


FIG. 3.1 – Un graphe enraciné (a) représentant un système d'équations (b) et son dépliage (c).

normalisé si ses étiquettes sont dans l'alphabet $\mathcal{F} \uplus \mathbb{N}$ et que l'ensemble des sommets se partage en deux ensembles V' et V'' tels que :

- le sommet racine appartient à V'
- les arcs étiquetés par un symbole de \mathcal{F} ont pour origine un sommet de V' et pour destination un sommet de V'' ,
- les arcs étiquetés par un entier ont pour origine un sommet de V'' et pour destination un sommet de V' ,
- chaque sommet de V' est origine d'un unique arc,
- chaque sommet de V'' est destination d'un unique arc,
- si un sommet de V'' est destination d'un arc étiqueté par le symbole $f \in \mathcal{F}$, alors il est origine d'exactly $|f|$ arcs étiquetés respectivement par $1, \dots, |f|$.

On associe un système d'équations à un graphe enraciné représentant un système d'équations de la manière suivante :

- l'ensemble X de variables est V' ,
- la variable résultat est la racine du graphe,
- pour $x \in V'$ un sommet, celui-ci n'est origine que d'un unique arc étiqueté par un symbole de \mathcal{F} . Soit f ce symbole et y la destination de cet arc. On a $y \in V''$ et donc y est origine d'exactly $|f|$ arcs étiquetés respectivement $1, \dots, |f|$. Soient donc $x_1, \dots, x_{|f|}$ les sommets destinations de ces arcs, on pose $Eq(x) = f(x_1, \dots, x_{|f|})$.

Bien entendu, la construction inverse peut être effectuée. Ce codage est donc une bijection entre les systèmes d'équations et les graphes représentant des systèmes d'équations. On se permet donc dorénavant d'identifier chaque système d'équations avec sa représentation sous forme d'un graphe.

Exemple 3.24 Soit $\mathcal{F} = \{f : 2, g : 1, a : 0\}$ un alphabet gradué. La figure 3.1 illustre un graphe (a) représentant un \mathcal{F} -système d'équations (b). Le dépliage du graphe est présenté en (c). Ce terme est aussi égal à $Solution[\mathcal{T}^\omega(\mathcal{F})](S)$.

Cette dernière remarque se généralise comme le montre la propriété suivante.

Propriété 3.25 Pour S un graphe enraciné représentant un \mathcal{F} -système d'équations,

$$Depliage(S) = Solution[\mathcal{T}^\omega(\mathcal{F})](S) .$$

Lorsque l'on combine cette dernière propriété avec le corollaire 3.22, on obtient le résultat ci-dessous qui identifie la solution d'un système à l'image par morphisme canonique du terme infini obtenu par dépliage du système.

Corollaire 3.26 Soit \mathcal{F} un alphabet gradué, une \mathcal{F} -algèbre continue \mathbf{A} et S un \mathcal{F} -système d'équations, alors,

$$Solution[\mathbf{A}](S) = \llbracket Depliage(S) \rrbracket_{\mathbf{A}} .$$

3.3 μ -définissabilité et μ -calcul

Nous avons vu au paragraphe précédent comment représenter certains éléments d'une algèbre continue :

1. tout d'abord, comme valeur de termes infinis,
2. puis, comme solutions de systèmes d'équations.

Nous avons aussi vu que ces deux approches coïncident et le passage du système d'équations au terme s'obtenait par dépliage.

Le propos de cette partie est de représenter des parties d'une algèbre continue, c'est-à-dire d'introduire une notion de «rationalité». Dans cette optique, nous utilisons le μ -calcul et nous définissons les ensembles μ -définissables de termes. Nous verrons que cette famille de langages satisfait de nombreuses propriétés de clôtures utiles par la suite. Nous étendrons ensuite cette notion de μ -définissabilité aux algèbres continues quelconques.

L'intérêt de cette approche est de disposer de tous les résultats classiques concernant le μ -calcul et en particulier de ses procédures de décision. En ce sens, l'un des résultats les plus importants de cette partie est le théorème 10 (page 71) qui traduit toute μ -formule exprimée sur les termes en formule monadique faisant

référence aux système d'équations. Ce résultat nous permet d'utiliser les techniques de décidabilité de la théorie monadique pour décider de l'appartenance de la solution d'un système d'équations à un ensemble μ -définissable.

Le μ -calcul est un formalisme qui a été bien étudié. Le lecteur intéressé par une présentation plus complète du domaine peut se reporter au récent ouvrage de Arnold et Niwiński [AN01]. La théorie du μ -calcul possède nombre de points communs avec celle des cpos, ne serait-ce que par la notion de point fixe qui les sous-tend toutes deux. Le parti est pris dans cet exposé de les considérer comme séparées.

3.3.1 Généralités sur les points fixes

La construction fondamentale du μ -calcul est la possibilité de calculer des points fixes, plus petits comme plus grands. Ce premier paragraphe présente quelques résultats généraux sur la notion de point fixe. Pour garantir l'existence de points fixes, nous travaillons dans un treillis complet. Un ensemble ordonné (E, \leq) est un *treillis complet* si :

- toute partie X de E admet une *borne inférieure*, c'est à dire un élément e de E noté $\bigwedge X$ tel que
 1. pour tout $a \in X$, $e \leq a$,
 2. soit $e' \in E$ tel que pour tout $a \in X$, $e' \leq a$, alors $e' \leq e$,
- toute partie X de E admet une *borne supérieure*, c'est à dire un élément e de E noté $\bigvee X$ tel que
 1. pour tout $a \in X$, $a \leq e$,
 2. soit $e' \in E$ tel que pour tout $a \in X$, $a \leq e'$, alors $e \leq e'$.

Remarque 3.27 *En fait, de l'existence d'une borne inférieure peut se déduire de l'existence d'une borne supérieure, et réciproquement. Ceci se montre en utilisant l'équivalence suivante:*

$$\text{pour tout } X \subseteq E, \quad \bigwedge X = \bigvee \{x \in E \mid \forall y \in X. x \leq y\} .$$

Une conséquence directe de la définition est que tout treillis complet possède un plus petit et un plus grand élément (respectivement la borne inférieure et la borne supérieure de E). On note $\mathbf{0}$ le plus petit élément et $\mathbf{1}$ le plus grand.

Pour la suite, deux treillis nous intéressent.

Le treillis des parties : Étant donné un ensemble A , l'ensemble des parties de A , ordonné par l'inclusion est un treillis complet dont le plus petit élément est \emptyset , le plus grand élément est E , la borne inférieure coïncide avec l'intersection ensembliste, et la borne supérieure avec l'union ensembliste.

Le treillis produit : Pour des treillis complets $(E_1, \leq_1), \dots, (E_n, \leq_n)$, $E_1 \times \dots \times E_n$ est ordonné par $(e_1, \dots, e_n) \leq (e'_1, \dots, e'_n)$ si pour tout i , $e_i \leq_i e'_i$. Il s'agit d'un treillis complet. Le plus petit (resp. grand) élément est le vecteur formé des plus petits (resp. grands) éléments des E_i . La borne inférieure (resp. supérieure) s'obtient en calculant la borne inférieure (resp. supérieure) composante par composante.

Pour A_1, \dots, A_n des ensembles, les ensembles $E = 2^{A_1 \uplus \dots \uplus A_n}$ et $F = 2^{A_1} \times \dots \times 2^{A_n}$ sont isomorphes. L'ensemble E est muni d'une structure de treillis complet en tant qu'ensemble des parties de $A_1 \uplus \dots \uplus A_n$. L'ensemble F est muni d'une structure de treillis complet en tant que produit des 2^{A_i} , chacun étant un treillis complet. En fait, les deux treillis sont isomorphes, c'est-à-dire que les ordres ainsi construits coïncident. Cette remarque permet d'assimiler tous les treillis complets que nous considérerons par la suite à des treillis des parties.

Une application f d'un treillis complet E dans un treillis complet F est *croissante* si pour tous $e, e' \in E$, si $e \leq e'$ alors $f(e) \leq f(e')$. Un point fixe d'une application d'un treillis complet E dans lui-même est un élément e de E tel que $f(e) = e$. Le théorème de Knaster-Tarski permet d'associer un unique plus petit point fixe et un unique plus grand point fixe à toute application croissante.

Théorème 8 (Knaster-Tarski) *Étant donnée une application croissante f d'un treillis complet E dans lui-même, on a*

1. $\bigwedge \{e \mid e = f(e)\} = \bigwedge \{e \mid f(e) \leq e\}$ est un point fixe de f (le plus petit),
2. $\bigvee \{e \mid e = f(e)\} = \bigvee \{e \mid e \leq f(e)\}$ est un point fixe de f (le plus grand).

Preuve. Montrons ce résultat pour le plus petit point fixe. Soit $X = \{e \in E \mid f(e) \leq e\}$. On a (a) $f(\bigwedge X) \leq \bigwedge \{f(e) \mid f(e) \leq e\} \leq \bigwedge X$ (par croissance de f et définition de X). On en déduit que $\bigwedge X \in X$. Par croissance de f , pour tout $e \in X$, $f(e) \in X$, et donc en particulier $f(\bigwedge X) \in X$. Il s'ensuit $\bigwedge X \leq f(\bigwedge X)$. Avec (a), on obtient $f(\bigwedge X) = \bigwedge X$. Il est dès lors évident que $\bigwedge \{e \in E \mid e = f(e)\} = \bigwedge X$.

Le plus grand point fixe se déduit symétriquement. □

Une application f d'un treillis complet E dans un treillis complet F est *additive* si pour toute partie A de E , $f(\bigvee A) = \bigvee f(A)$. De même, une application est *multiplicative* si $f(\bigwedge A) = \bigwedge f(A)$. Les notions d'additivité et de multiplicativité servent de notion de «continuité» dans notre cadre : les applications additives (resp. multiplicatives) commutent avec la borne supérieure (resp. inférieure). On s'aperçoit aisément que toute application additive (resp. multiplicative) est croissante et que les applications additives (resp. multiplicatives) sont closes par composition.

La propriété 3.28 nous permet de préciser ce que sont les applications à la fois additives et multiplicatives dans le treillis des parties.

Propriété 3.28 *Soit f une application de 2^A dans 2^B pour deux ensembles A et B . Alors, f est additive et multiplicative si et seulement si il existe une fonction η de B dans A et une partie B_0 de B telle que pour tout $X \subseteq A$, $f(X) = \eta^{-1}(X) \cup B_0$.*

Preuve. \Rightarrow Supposons tout d'abord $f(\emptyset) = \emptyset$. On a alors par multiplicativité, pour $a \neq b$ de E , $\emptyset = f(\{a\} \cap \{b\}) = f(\{a\}) \cap f(\{b\})$. Donc, pour tout x de B , il existe au plus un a tel que $x \in f(\{a\})$. On pose dans ce cas $\eta(x) = a$. Dans les autres cas, $\eta(x)$ n'est pas défini. Par conséquent, pour tout $a \in A$, $\eta^{-1}(\{a\}) = f(\{a\})$. On obtient alors pour tout $X \subseteq A$, $f(X) = \eta^{-1}(X)$ par additivité en remplaçant X par $\cup\{\{a\} \mid a \in X\}$.

Dans le cas général, on pose $B_0 = f(\emptyset)$ et pour tout $X \subseteq A$, $f'(X) = f(X) - B_0$. Par croissance de f , pour tout $X \subseteq A$, $B_0 \subseteq f(X)$, et donc $f'(X) \cup B_0 = (f(X) - B_0) \cup B_0 = f(X)$. L'application f' est additive et multiplicative, et d'après le cas précédent, il existe η telle que pour tout X , $f'(X) = \eta^{-1}(X)$. On a alors $f(X) = \eta^{-1}(X) \cup B_0$.

\Leftarrow évident. □

Remarque 3.29 *Il existe des applications additives et non multiplicatives (ou vice-versa). Considérons par exemple l'application f de $2^{\{a,b\}}$ dans $2^{\{c\}}$ définie par $f(\emptyset) = \emptyset$ et $f(\{a\}) = f(\{b\}) = f(\{a,b\}) = \{c\}$. L'application f est additive. En revanche $f(\{a\} \cap \{b\}) \neq f(\{a\}) \cap f(\{b\})$, et par conséquent f n'est pas multiplicative.*

Sous l'hypothèse d'additivité ou de multiplicativité, il est possible d'envoyer les points fixes de fonctions croissantes d'un treillis vers un autre.

Lemme 3.30 (lemme de transfert) *Considérons E et F deux treillis complets et les applications*

- f croissante de E dans E ,
- g croissante de F dans F ,
- et ρ de E dans F ,

telles que $\rho \circ f = g \circ \rho$ $\left(\begin{array}{ccc} & & f \\ & E & \longrightarrow & E \\ \rho \downarrow & & & \rho \downarrow \\ & F & \longrightarrow & F \end{array} \right)$.

1. Si $\rho(\mathbf{0}) = \mathbf{0}$ et ρ est additive alors pour x le plus petit point fixe de f et y le plus petit point fixe de g , on a $\rho(x) = y$,
2. si $\rho(\mathbf{1}) = \mathbf{1}$ et ρ est multiplicative alors pour x le plus grand point fixe de f et y le plus grand point fixe de g , on a $\rho(x) = y$.

Preuve. Vérifions le cas 1. On a $g(\rho(x)) = \rho(f(x)) = \rho(x)$, donc (a) $\rho(x)$ est un point fixe de g .

Considérons $X = \{e \mid \rho(e) \leq y\}$. L'ensemble X est non vide puisque $\rho(\mathbb{0}) = \mathbb{0} \leq y$. Pour tout $e \in X$, $\rho(f(e)) = g(\rho(e)) \leq g(y) = y$ donc (b) $f(e) \in X$. On a $\rho(\vee X) = \vee \rho(X) \leq y$ (additivité de ρ , définition de X et non vacuité de X), donc (c) $\vee X \in X$. (b) et (c) donnent $f(\vee X) \in X$ et donc $f(\vee X) \leq \vee X$. D'après Knaster-Tarski, on a alors $x \leq \vee X$, et donc $\rho(x) \leq y$.

Comme $\rho(x)$ est un point fixe de g et que y est le plus petit point fixe de g , on a $\rho(x) = y$.

Le cas 2 s'obtient par symétrie du cas précédent. \square

3.3.2 Syntaxe et sémantique du μ -calcul

Le μ -calcul introduit une syntaxe pour manipuler les notions de points fixes présentées au paragraphe précédent. L'opérateur μ calcule le plus petit point fixe et l'opérateur ν le plus grand.

Formellement, il convient de nouveau de se fixer un alphabet infini dénombrable \mathcal{V}_μ de *variables du μ -calcul*. Pour un alphabet gradué \mathcal{F} , les formules du μ -calcul (ou μ -formules) sur \mathcal{F} sont définies récursivement comme suit :

- pour $x \in \mathcal{V}_\mu$ une variable, x est une μ -formule,
- pour f un symbole de \mathcal{F} et $\phi_1, \dots, \phi_{|f|}$ des μ -formules, $f(\phi_1, \dots, \phi_{|f|})$ est une μ -formule,
- pour ϕ une μ -formule et $x \in \mathcal{V}_\mu$ une variable, $\mu x.\phi$ est une μ -formule,
- pour ϕ une μ -formule et $x \in \mathcal{V}_\mu$ une variable, $\nu x.\phi$ est une μ -formule,
- pour ϕ et ψ deux formules, $\phi \cap \psi$ et $\phi \cup \psi$ sont des μ -formules.

On définit l'ensemble $FV(\phi)$ des variables libres d'une formule du μ -calcul de façon usuelle. Une formule sans variables libres est appelée *formule close*. Les conventions habituelles concernant le renommage implicite des variables liées sont adoptées.

Pour ϕ et ψ deux μ -formules et x une variable libre de ϕ telles que les variables libres de ψ n'apparaissent pas dans $FV(\phi) - \{x\}$, on note $\phi\{\psi/x\}$ la formule ϕ dans laquelle toutes les occurrences libres de la variable x ont été remplacées syntaxiquement par ψ . Si l'on fait l'hypothèse d'une α -conversion implicite, alors il suffit de supposer que $(FV(\phi) - \{x\}) \cap FV(\psi) = \emptyset$. Plus généralement, on note $\phi\{\psi_i/x_i\}_{i \in [n]}$ pour signifier qu'aux n variables x_1, \dots, x_n ont été respectivement (et simultanément) substituées les μ -formules ψ_1, \dots, ψ_n .

Le domaine de calcul du μ -calcul⁴ est l'ensemble des parties de $\mathcal{T}^\omega(\mathcal{F})$. Pour un ensemble X de μ -variables, une valuation de X est une application γ qui

4. Pour le μ -calcul présenté dans ce travail.

à toute variable $x \in X$ associe $\gamma(x) \subseteq \mathcal{T}^\omega(\mathcal{F})$. On note $\gamma, x \rightarrow E$ pour x une variable et E une partie de $\mathcal{T}^\omega(\mathcal{F})$ la valuation qui coïncide avec γ sur le domaine de γ , et qui à x associe E .

Etant donnée une *valuation* γ de l'ensemble X de μ -variables, la valeur d'une formule ϕ de variables libres X dans le contexte γ est une partie de $\mathcal{T}^\omega(\mathcal{F})$ notée $\phi[\gamma]$ et définie récursivement par :

- $x[\gamma] = \gamma(x)$,
- $f(\phi_1, \dots, \phi_{|f|})[\gamma] = \{f(t_1, \dots, t_{|f|}) \mid t_1 \in \phi_1[\gamma], \dots, t_{|f|} \in \phi_{|f|}[\gamma]\}$,
- $(\mu x.\phi)[\gamma] = \bigcap \{A \subseteq \mathcal{T}^\omega(\mathcal{F}) \mid \phi[\gamma, x \rightarrow A] = A\}$,
- $(\nu x.\phi)[\gamma] = \bigcup \{A \subseteq \mathcal{T}^\omega(\mathcal{F}) \mid \phi[\gamma, x \rightarrow A] = A\}$,
- $(\phi \cap \psi)[\gamma] = \phi[\gamma] \cap \psi[\gamma]$,
- $(\phi \cup \psi)[\gamma] = \phi[\gamma] \cup \psi[\gamma]$.

On a $\mu x.x[] = \emptyset$ et $\nu x.x = \mathcal{T}^\omega(\mathcal{F})$. Ainsi, $\mathbb{1}$ est une abbréviation pour $\nu x.x$, et $\mathbb{0}$ en est une pour $\mu x.x$. Pour $\phi(x_1, \dots, x_n)$ une μ -formule de variables libres x_1, \dots, x_n , et A_1, \dots, A_n des parties de $\mathcal{T}^\omega(\mathcal{F})$, on note $\phi(A_1, \dots, A_n)$ pour $\phi(x_1, \dots, x_n)[x_i \rightarrow A_i \mid i \in [n]]$.

Une remarque fondamentale pour la suite est que toutes les constructions du μ -calcul sont croissantes.

Lemme 3.31 *Pour $\phi(x_1, \dots, x_n)$ une μ -formule de variables libres x_1, \dots, x_n , l'application qui à (A_1, \dots, A_n) associe $\phi(A_1, \dots, A_n)$ est croissante, c.a.d. pour tout $A'_1 \subseteq A_1, \dots, A'_n \subseteq A_n$, $\phi(A'_1, \dots, A'_n) \subseteq \phi(A_1, \dots, A_n)$.*

Preuve. Par récurrence sur la structure de ϕ . □

En combinant ce résultat au théorème 8 de Knaster-Tarski, il s'ensuit que μ et ν calculent effectivement des points fixes (ce que formalise le corollaire qui suit).

Corollaire 3.32 *Étant donné une formule ϕ de variables libres x, y_1, \dots, y_n et $A_1, \dots, A_n \subseteq \mathcal{T}(\mathcal{F})$,*

si $A = (\mu x.\phi)[y_1 \rightarrow A_1, \dots, y_n \rightarrow A_n]$ ou $A = (\nu x.\phi)[y_1 \rightarrow A_1, \dots, y_n \rightarrow A_n]$, alors $\phi[x \rightarrow A, y_1 \rightarrow A_1, \dots, y_n \rightarrow A_n] = A$.

Le lemme de substitution montre qu'effectuer des substitutions à l'intérieur de μ -formules est consistant avec la valeur de ces formules.

Lemme 3.33 (de substitution) *Soient ϕ et ψ deux μ -formules d'ensemble X de variables libres, x une variable et γ une valuation de X , on a*

$$\phi\{\psi/x\}[\gamma] = \phi[\gamma, x \rightarrow \psi[\gamma]] .$$

Preuve. Par récurrence sur la structure de ϕ . □

3.3.3 Décomposition du μ -calcul

Il s'agit ici de montrer comment décomposer une formule logique en sous-formules. Il s'agit d'une variante de la clôture de Fischer-Ladner utilisée pour la logique PDL [FL79] puis étendue pour le μ -calcul [Koz83].

Soit ϕ_0 une μ -formule close, la clôture de ϕ , notée $Cl(\phi_0)$ est le plus petit ensemble de formules closes défini par :

- $\phi_0 \in Cl(\phi_0)$,
- si $\phi \cap \psi \in Cl(\phi_0)$ alors $\phi \in Cl(\phi_0)$ et $\psi \in Cl(\phi_0)$,
- si $\phi \cup \psi \in Cl(\phi_0)$ alors $\phi \in Cl(\phi_0)$ et $\psi \in Cl(\phi_0)$,
- si $f(\psi_1, \dots, \psi_{|f|}) \in Cl(\phi_0)$ alors pour tout i de $[|f|]$, $\psi_i \in Cl(\phi_0)$,
- si $\mu x.\phi \in Cl(\phi_0)$ alors $\phi\{\mu x.\phi/x\} \in Cl(\phi_0)$,
- si $\nu x.\phi \in Cl(\phi_0)$ alors $\phi\{\nu x.\phi/x\} \in Cl(\phi_0)$.

Lemme 3.34 *Pour ϕ_0 une formule du μ -calcul, $Cl(\phi_0)$ est fini.*

Preuve. Classique. Voir [Koz83]. □

L'ensemble ainsi défini permet d'utiliser une formule du μ -calcul ϕ comme un automate d'arbres dont l'ensemble d'états est $Cl(\phi)$. Le lemme ci-dessous explicite les transitions de cet automate.

Lemme 3.35 *Pour toute formule close ϕ du μ -calcul sur l'alphabet gradué \mathcal{F} , il existe une application D_ϕ qui à chaque symbole $f \in \mathcal{F}$ associe une partie de $Cl(\phi)^{|f|}$ telle que :*

$$\phi[] = \left(\bigcup_{\substack{f \in \mathcal{F} \\ (\psi_1, \dots, \psi_{|f|}) \in D_\phi(f)}} f(\psi_1, \dots, \psi_{|f|}) \right) [] .$$

Preuve. Classique. Voir [Koz83]. □

3.3.4 Extension vectorielle du μ -calcul

Dans la pratique, il nuit souvent à la lisibilité de devoir traiter chaque variable du μ -calcul indépendamment. C'est en particulier vrai si l'on veut effectuer des conversions entre systèmes d'équations et μ -calcul. Dans de telles constructions,

il est plus aisé de considérer toutes les variables du système d'équations comme un vecteur sur lequel les opérateurs de point fixe peuvent agir. C'est cette possibilité que propose l'extension vectorielle du μ -calcul. La propriété 3.37 montre que cette extension n'augmente pas l'expressivité du formalisme. Cette équivalence nous permet donc d'utiliser ces constructions vectorielles comme des raccourcis de notations.

Soit $\theta \in \{\mu, \nu\}$, x_1, \dots, x_n des variables, ϕ_1, \dots, ϕ_n des μ -formules vectorielles et $i \in [n]$,

$$\theta x_i. \left\{ \begin{array}{l} x_1 = \phi_1 \\ \vdots \\ x_n = \phi_n \end{array} \right\}$$

est une μ -formule vectorielle. On utilise aussi les notations ensemblistes

$$\theta x_i. \{x_k = \phi_k \mid k \in [n]\}.$$

Les autres constructions du μ -calcul vectoriel sont celles du μ -calcul.

La sémantique de la nouvelle construction μ (resp. ν) est de calculer la plus petite (resp. la plus grande) valeur de x_i solution du système d'équations entre accolades.

Formellement, pour γ une valuation,

$$\mu x_i. \{x_k = \phi_k \mid k \in [n]\}[\gamma] = \bigwedge \{e_i \mid \forall k \in [n], e_k = \phi_k[\delta], \delta = \gamma, x_1 \rightarrow e_1, \dots, x_n \rightarrow e_n\}$$

Cela revient à effectuer le calcul de point fixe dans le treillis produit, puis de projeter le résultat sur la i -ème composante.

On a bien évidemment pour $\theta \in \{\mu, \nu\}$, toute μ -formule ϕ et toute variable x l'équivalence suivante :

$$\theta x. \{x = \phi\} \equiv \theta x. \phi \tag{3.1}$$

Pour plus de deux équations, c'est alors la transformation suivante qui s'applique.

Lemme 3.36 *Pour $\theta \in \{\mu, \nu\}$, ϕ_1, \dots, ϕ_n des μ -formules et pour i de $[n]$, on a*

$$\theta x_i. \{x_k = \phi_k \mid k \in [n]\} \equiv \theta x_i. \phi_i \{ \theta x_j. \{x_k = \phi_k \mid k \in [n], k \neq i\} / x_j \}_{j \in [n] - \{i\}} \tag{3.2}$$

On applique récursivement les équivalences 3.1 et 3.2 pour obtenir la propriété ci-dessous.

Propriété 3.37 (principe de Bekič [Bek67]) *Toute formule du μ -calcul vectoriel est effectivement équivalente à une formule du μ -calcul.*

Ainsi, d'un point de vue expressivité, il est équivalent de considérer le μ -calcul ou le μ -calcul vectoriel.

3.3.5 Termes marqués

À plusieurs reprises dans ce travail, il est nécessaire de disposer d'une représentation des formules monadiques «effective» sur les arbres déterministes. La réponse classique consiste à utiliser des automates d'arbres infinis avec condition d'acceptation de Rabin ou de parité. Dans notre cas, nous utilisons le μ -calcul qui est équivalent à ces deux formalismes. Deux aménagements sont néanmoins nécessaires.

- Le μ -calcul ne pouvant pas manipuler des ensembles de sommets, il faut disposer d'annotations sur l'arbre pour qu'une formule monadique puisse y faire référence.
- Le μ -calcul est défini pour des termes et non pour des arbres déterministes; en particulier, seuls les sommets principaux d'un terme ont une signification pour le μ -calcul. Il faut donc disposer de résultats permettant de passer des arbres déterministes aux termes et réciproquement.

Les constructions de ce paragraphe ont donc pour but de préciser ces points et les notations. Il s'agit de constructions classiques.

On prend la convention que les termes ont pour ensemble de sommets leur langage de branches (cf corollaire 2.5 page 13). Ainsi, on identifie l'ensemble des sommets principaux d'un \mathcal{F} -terme t au langage $P_t = \mathcal{LB}(t) \cap (\mathcal{FN})^*\mathcal{F}$. Pour tout $f \in \mathcal{F}$, on note P_t^f l'ensemble des sommets principaux origine d'un symbole f , c. a. d. $P_t^f = \mathcal{LB}(t)f^{-1} = \{u \mid uf \in \mathcal{LB}(t)\}$.

Étant donné un alphabet gradué \mathcal{F} et un ensemble fini M de marques, on considère l'alphabet gradué $\mathcal{F}_M = \mathcal{F} \times 2^M$ avec $|(f, E)| = |f|$ pour tout $(f, E) \in \mathcal{F}_M$. Pour $f \in \mathcal{F}$ et $\phi_1, \dots, \phi_{|f|}$ des formules monadiques, on écrit $f(\phi_1, \dots, \phi_{|f|})$ à la place de $\cup_{E \subseteq M} (f, E)(\phi_1, \dots, \phi_{|f|})$. Pour $m \in M$, on écrit m à la place de $\cup_{f \in \mathcal{F}, m \in E} (f, E)(\mathbf{1}, \dots, \mathbf{1})$ et $\neg m$ à la place de $\cup_{f \in \mathcal{F}, m \notin E} (f, E)(\mathbf{1}, \dots, \mathbf{1})$. Soit t un \mathcal{F}_M -terme, notons \bar{t} le terme obtenu de t en oubliant les marques.

Un terme sur \mathcal{F}_M peut être vu comme un \mathcal{F} -terme augmenté de marques appartenant à M sur chacun des symboles. Les marques sur un \mathcal{F}_M -terme t peuvent elles-mêmes être identifiées aux ensembles de sommets principaux de t . Ainsi, soit t un \mathcal{F}_M -terme et $m \in M$, on pose $m^{-1}(t) = \{v \mid t^v \in m[]\}$. Pour tout terme t sur \mathcal{F} , tout entier n et tous U_1, \dots, U_n ensembles de sommets principaux de t , on note (t, U_1, \dots, U_n) l'unique $\mathcal{F}_{[n]}$ -terme t' tel que $\bar{t}' = t$ et pour tout $i \in [n]$, $i^{-1}(t') = U_i$.

Lemme 3.38 *Étant donnés \mathcal{F} un alphabet gradué, n un entier naturel, on a les propriétés suivantes.*

- Pour toute formule monadique $\Phi(X_1, \dots, X_n)$, il existe une formule close du μ -calcul ϕ telle que pour tout \mathcal{F} -terme t et tous U_1, \dots, U_n ensembles de

sommets principaux de t , on a,

$$t \models \Phi(U_1, \dots, U_n) \quad \text{ssi} \quad (t, U_1, \dots, U_n) \in \phi[] .$$

- Pour toute formule close du μ -calcul ϕ , il existe une formule monadique $\Phi(X_1, \dots, X_n)$ telle que pour tout \mathcal{F} -terme t et tous U_1, \dots, U_n ensembles de sommets principaux de t , on a,

$$t \models \Phi(U_1, \dots, U_n) \quad \text{ssi} \quad (t, U_1, \dots, U_n) \in \phi[] .$$

Preuve. Il s'agit d'une variante du théorème 9 page 70. □

Ce résultat n'est pas encore suffisant pour traiter les arbres déterministes quelconques. Il faut se ramener à des ensembles de sommets principaux.

Lemme 3.39 *Étant donné un ensemble d'étiquettes A , il existe une transduction monadique T telle que pour tout graphe déterministe \mathcal{G} étiqueté par A ,*

- $T(\mathcal{G})$ est un $\{p : 2, c : 0\}$ -système d'équations,
 - il existe une injection ρ de $V_{\text{Depliage}(\mathcal{G})}$ dans $P_{\text{Depliage}(T(\mathcal{G}))}^c$ telle que pour toute formule monadique $\Phi(X_1, \dots, X_n)$ il existe une formule monadique $\Psi(X_1, \dots, X_n)$ telle que pour toutes parties U_1, \dots, U_n de $V_{\text{Depliage}(\mathcal{G})}$, on a
- $$\text{Depliage}(\mathcal{G}) \models \Phi(U_1, \dots, U_n) \quad \text{ssi} \quad \text{Depliage}(T(\mathcal{G})) \models \Psi(\rho(U_1), \dots, \rho(U_n)) .$$

Preuve. Codage long sans difficultés particulières. □

3.3.6 μ -définissabilité

Pour \mathcal{F} un alphabet gradué, les parties des termes $\mathcal{T}^\omega(\mathcal{F})$ obtenues comme valeur d'une μ -formule close sont appelées parties \mathcal{F} - μ -définissables (ou simplement μ -définissables s'il n'y a pas d'ambiguïté sur l'alphabet).

Une partie L d'une \mathcal{F} -algèbre continue \mathbf{A} est μ -définissable si l'ensemble $\{t \in \mathcal{T}^\omega(\mathcal{F}) \mid \llbracket t \rrbracket_{\mathbf{A}} \in L\}$ est μ -définissable.

Remarque 3.40 *On ne peut pas considérer que cette définition procure une représentation de l'ensemble L par l'ensemble L' . En particulier, si $a \in \mathbf{A}$ est tel que $a \notin \{\llbracket t \rrbracket \mid t \in \mathcal{T}^\omega(\mathcal{F})\}$ alors $L \cup \{a\}$ et $L - \{a\}$ sont tout autant représentés par L' .*

Cette définition procure une représentation effective de parties d'une algèbre. Si on dispose d'algorithmes permettant de décider de l'appartenance d'un terme à

un ensemble μ -définissable, alors ce test d'appartenance se transfère directement à toute algèbre continue.

Exemple 3.41 *Considérons de nouveau l'algèbre continue N présentée à l'exemple 3.20.*

L'ensemble des termes s'évaluant en une valeur non nulle est alors N - μ -définissable. Une μ -formule correspondante, notée simplement (> 0) est :

$$(> 0) = \mu x. 1 \cup \infty \cup (\mathbf{1} + x) \cup (x + \mathbf{1}) \cup (x * x) ,$$

c'est à dire le plus petit ensemble E de termes contenant $1, \infty$, tel que la somme d'un terme de E avec un terme quelconque appartient à E et le produit de deux termes de E appartient à E .

L'ensemble des termes d valeur strictement supérieure à 1 est aussi μ -définissable :

$$(> 1) = \mu x. \infty \cup ((> 0) + (> 0)) \cup (x + \mathbf{1}) \cup (\mathbf{1} + x) \cup (x * (> 0)) \cup ((> 0) * x) .$$

L'ensemble des termes s'évaluant en ∞ est aussi μ -définissable, mais la construction est plus délicate :

$$\begin{aligned} (= \infty) = \nu x. \mu y. & \quad \infty \\ & \cup (y + \mathbf{1}) \cup (\mathbf{1} + y) \cup (y * (> 0)) \cup ((> 0) * y) \\ & \cup (x + (> 0)) \cup ((> 0) + x) \cup (x * (> 1)) \cup ((> 1) * x) . \end{aligned}$$

La propriété exprimée est qu'un terme s'évalue en l'infini si,

- *il contient la constante ∞ (et que celle-ci n'est pas multipliée par 0),*
- *ou bien il existe une branche infinie sur laquelle :*

1. *toutes les opérations ont un résultat supérieur à leurs arguments, c'est à dire qu'il peut y avoir des sommes et des multiplications pourvu que l'autre argument des multiplications ne soit pas nul (y dans la formule),*
2. *il existe une infinité d'opérations dont le résultat est strictement supérieur aux arguments, c'est à dire une somme avec une valeur non nulle ou une multiplication par une valeur supérieure ou égale à 2 (x dans la formule).*

La propriété 3.42 liste quelques opérations utiles laissant les parties μ -définissables invariantes.

Propriété 3.42 *Les propriétés suivantes sont satisfaites :*

- *la famille des parties μ -définissables est closes par union, intersection et complémentation de chacune des parties;*
- *pour f un symbole d'arité n et L_1, \dots, L_n des parties μ -définissables, l'ensemble $f(L_1, \dots, L_n)$ est une partie μ -définissable;*

– pour tout symbole f d'arité n , tout entier $i \in [n]$ et toute partie μ -définissable L ,

$$(fi)^{-1}(L) = \{t_i \mid f(t_1, \dots, t_n) \in L\}$$

est une partie μ -définissable⁵.

Preuve. Toutes ces propriétés se traduisent directement dans la syntaxe du μ -calcul si ce n'est celles concernant le complémentaire et $(fi)^{-1}$. Le complémentaire (notons le \neg) s'obtient simplement en interchangeant \cup et \cap , μ et ν et en utilisant l'identité suivante :

$$\neg(f(\phi_1, \dots, \phi_{|f|})) \equiv \bigcup_{g \in \mathcal{F}, g \neq f} g(\mathbf{1}, \dots, \mathbf{1}) \cup \bigcup_{i \in [|f|]} f(\mathbf{1}, \dots, \mathbf{1}, \neg(\phi_i), \mathbf{1}, \dots, \mathbf{1}) .$$

↑
ième position

Étant donnée une μ -formule ϕ , la μ -formule $(fi)^{-1}(\phi)$ se construit par récurrence. Si les variables libres de ϕ sont les x_1, \dots, x_n , alors les variables libres de $(fi)^{-1}(\phi)$ sont les $x_1, \dots, x_n, \hat{x}_1, \dots, \hat{x}_n$. Les variables x correspondent aux mêmes valeurs que dans ϕ (l'expression est simplement copiée). Chaque variable \hat{x} , quant à elle, représente $(fi)^{-1}(x)$.

$$\begin{aligned} (fi)^{-1}(f(\phi_1, \dots, \phi_n)) &= \phi_i \\ \text{pour } g \neq f, \quad (fi)^{-1}(g(\phi_1, \dots, \phi_{|g|})) &= \emptyset \\ (fi)^{-1}(\phi \cup \psi) &= (fi)^{-1}(\phi) \cup (fi)^{-1}(\psi) \\ (fi)^{-1}(\phi \cap \psi) &= (fi)^{-1}(\phi) \cap (fi)^{-1}(\psi) \\ \text{pour } \theta \in \{\mu, \nu\}, \quad (fi)^{-1}(\theta x. \phi) &= \theta \hat{x}. \left\{ \begin{array}{l} \hat{x} = (fi)^{-1}(\phi) \\ x = \phi \end{array} \right\} \\ (fi)^{-1}(x) &= \hat{x} \end{aligned}$$

On montre alors par récurrence la propriété suivante :

$$(fi)^{-1}(\phi[\gamma]) = ((fi)^{-1}(\phi))[\gamma'] , \quad \text{avec } \begin{cases} \gamma'(x) = \gamma(x) , \\ \gamma'(\hat{x}) = (fi)^{-1}(\gamma(x)) . \end{cases}$$

Plus précisément, on utilise le lemme de transfert 3.30 pour les opérations de point fixe (l'application $(fi)^{-1}$ est additive et multiplicative d'après la propriété 3.28). \square

Citons pour finir une variante du théorème de Rabin pour le μ -calcul.

⁵. La notation $(fi)^{-1}$ est choisie pour coïncider avec l'expression correspondante sur le langage de branches : $\mathcal{LB}((fi)^{-1}(t)) = (fi)^{-1}(\mathcal{LB}(t))$.

Théorème 9 Une partie L de $\mathcal{T}^\omega(\mathcal{F})$ est μ -définissable si et seulement si elle est monadiquement définissable, c.a.d. s'il existe une formule monadique close Φ telle que pour tout terme $t \in \mathcal{T}^\omega(\mathcal{F})$,

$$t \in L \text{ si et seulement si } t \models \Phi.$$

Dans ce théorème, la partie délicate consiste à traduire la formule monadique en une μ -formule. En fait, le théorème suivant, qui est une variante du résultat de Courcelle [Cou95] montrant que le dépliage est MS-compatible⁶, présente une forme renforcée de la réciproque : à toute μ -formule correspond une formule monadique qui est satisfaite par un graphe si et seulement si la μ -formule est satisfaite sur son dépliage.

Théorème 10 Soit L une partie \mathcal{F} - μ -définissable. Il existe une formule monadique Φ telle que pour tout \mathcal{F} -système d'équations S :

$$\text{Depliage}(S) \in L \text{ si et seulement si } S \models \Phi.$$

Preuve. La construction de la formule monadique s'obtient par simple récurrence sur la structure de la μ -formule définissant L . Etant donnée une μ -formule ϕ de variables libres x_1, \dots, x_n , on construit la formule monadique $\phi^*(y)$ de variables libres du second ordre monadique $\hat{x}_1, \dots, \hat{x}_n$ et possédant une unique variable libre du premier ordre y (exprimée comme un argument), comme suit :

$$\begin{aligned} x^*(y) &= y \in \hat{x} \\ f(\phi_1, \dots, \phi_{|f|})^*(y) &= \exists z_1 \dots \exists z_{|f|}. \bigwedge_{i \in [|f|]} \phi_i^*(z_i) \wedge y = f(z_1, \dots, z_n) \\ (\phi \cap \psi)^*(y) &= \phi^*(y) \wedge \psi^*(y) \\ (\phi \cup \psi)^*(y) &= \phi^*(y) \vee \psi^*(y) \\ (\mu x. \phi)^*(y) &= \forall \hat{x}. (\forall z. \phi^*(z) \Rightarrow z \in \hat{x}) \Rightarrow y \in \hat{x} \\ (\nu x. \phi)^*(y) &= \exists \hat{x}. (\forall z. z \in \hat{x} \Rightarrow \phi^*(z)) \wedge y \in \hat{x} \end{aligned}$$

On reconnaît dans la traduction de la construction μ la propriété « y appartient à tout ensemble clos par ϕ^* » ce qui correspond à un calcul de plus petit point fixe. La construction duale est appliquée à ν . On montre alors par récurrence sur ϕ que pour toute valuation γ et tout sommet v :

$$\text{Depliage}(S, v) \in \phi[\gamma] \text{ si et seulement si } \gamma', S \models \phi^*(v) \\ \text{avec } \gamma'(\hat{x}) = \{s \in S \mid \text{Depliage}(S, s) \in \gamma(x)\}.$$

□

6. Le résultat complet est généralement attribué à Courcelle et Walukiewicz, mais la restriction aux graphes déterministes dont il est question ici lui est antérieure.

3.4 Transducteurs déterministes

Nous avons vu comment les systèmes d'équations permettaient de représenter des éléments d'une algèbre continue. Nous avons aussi présenté la notion de μ -définissabilité qui permet de représenter des parties d'une algèbre continue et de tester l'appartenance d'une solution d'un système d'équations à une telle partie. Il s'agit maintenant de fournir un formalisme permettant de transformer les systèmes d'équations : les *transducteurs déterministes* (descendant avec anticipation μ -définissable).

Historiquement, les transducteurs de termes ont été introduits dans le cadre de la théorie de la compilation au début des années 70. Ces transducteurs permettent de mettre en relation des termes finis. Une introduction à ce domaine est disponible dans [CDG⁺97]. Une famille particulièrement intéressante de tels transducteurs a été proposée par Engelfriet [Eng77], celle des transducteurs descendants déterministes avec anticipation régulière (deterministic top-down transducers with regular lookahead). Ces transducteurs possèdent de bonnes propriétés de décidabilité et de clôture. Engelfriet [Eng94] a aussi montré l'étroit lien qu'entretiennent ces transformations d'arbres finis avec les transformations de graphes finis.

C'est dans un objectif similaire que nous considérons les transducteurs dans ce mémoire. Cependant, la théorie classique n'est pas suffisante dans le cas présent car il nous faut manipuler des objets infinis à la place de familles d'objets finis. Pour cette raison, nous présentons ici une adaptation d'une partie de cette théorie aux termes infinis (cette extension a été originellement considérée dans [CL04]). L'objectif est de définir une classe de transformations de termes infinis satisfaisant les conditions informelles suivantes :

- elle doit être suffisamment expressive pour capturer les transformations utiles dans le reste de ce travail,
- la validité des transformations doit pouvoir être montrée simplement,
- ces transformations doivent pouvoir être exécutées symboliquement sur une version repliée d'un terme infini, c'est à dire sur un système d'équations.

3.4.1 Définition

Un *transducteur descendant déterministe avec anticipation μ -définissable* (ou plus simplement dans le reste de ce travail, un *transducteur déterministe*) est un quintuplet $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ où :

- \mathcal{F} est un alphabet gradué, l'alphabet d'entrée,
- \mathcal{F}' est un alphabet gradué, l'alphabet de sortie,
- Q est un ensemble fini, l'ensemble des états,

- q_{init} est un élément de Q , l'état initial,
- δ est un ensemble fini de transitions de l'une des formes suivantes :
 - $q(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x))$ (production de f)
avec $f \in \mathcal{F}'$, $q, q_1, \dots, q_{|f|} \in Q$ et x une variable⁷.
 - $q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i)$ (consommation de f)
avec $f \in \mathcal{F}$, $q, r \in Q$, $i \in [|f|]$ et $x_1, \dots, x_{|f|}$ variables.
 - $q(x \in L) \rightarrow r(x)$ (anticipation de L)
avec $q, r \in Q$, L une partie μ -définissable de $\mathcal{T}^\omega(\mathcal{F})$ et x une variable.

On appelle q -transition toute transition de δ dont l'état de tête est q . On fait aussi référence à un état q en tant qu'état de production (resp. de consommation, d'anticipation) s'il existe une q -transition de production (resp. de consommation, d'anticipation).

Déterminisme: on impose de plus les contraintes suivantes sur les transitions :

- s'il existe dans δ une q -transition de production alors il s'agit de la seule q -transition de δ .
- s'il existe dans δ une q -transition de consommation alors il s'agit de la seule⁸ q -transition de δ .
- s'il existe dans δ une q -transition d'anticipation alors toutes les q -transitions sont d'anticipation et pour deux q -transitions distinctes $q(x \in L_1) \rightarrow q_1(x)$ et $q(x \in L_2) \rightarrow q_2(x)$, on a $L_1 \cap L_2 = \emptyset$.

L'application d'un transducteur déterministe à un terme infini est définie par application des transitions de δ comme des règles de réécriture, puis passage à la limite. Il faut pourtant être prudent: on ne peut en effet construire l'image d'un terme infini par un transducteur déterministe comme la limite de l'image d'une suite de termes finis; en effet, les fonctions calculées par les transducteurs déterministes ne sont pas, en général, continues.

Considérons donc un transducteur déterministe $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$. On définit par récurrence sur n , pour tout état q et tout terme $t \in \mathcal{T}^\omega(\mathcal{F})$, le terme $\delta_n(q, t) \in \mathcal{T}(\mathcal{F}'_\perp)$ par :

- pour tout $t \in \mathcal{T}^\omega(\mathcal{F})$ et tout $q \in Q$, $\delta_0(q, t) = \perp$,
- pour tout $t \in \mathcal{T}^\omega(\mathcal{F})$, tout $q \in Q$ et tout entier n ,
 - si $q(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta$, $\delta_{n+1}(q, t) = f(\delta_n(q_1, t), \dots, \delta_n(q_{|f|}, t))$,
 - si $q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i) \in \delta$, $\delta_{n+1}(q, (f(t_1, \dots, t_{|f|}))) = \delta_n(r, t_i)$,
 - si $q(x \in L) \rightarrow r(x) \in \delta$, et $t \in L$, $\delta_{n+1}(q, t) = \delta_n(r, t)$,

7. Nous n'introduisons pas ici d'ensemble de variables: celles-ci sont présentes uniquement pour des raisons de lisibilité des règles. Elles peuvent tout aussi bien être omises dans la définition.

8. Cette contrainte peut sembler excessive, mais il est toujours possible de s'y ramener en faisant précéder la transition de consommation de transitions d'anticipations.

- si aucune règle ne s'applique ou bien si le membre droit de la règle n'est pas défini, alors $\delta_n(q, t)$ n'est pas défini.

Remarquons tout d'abord que, grâce aux hypothèses de déterminisme, deux règles ne peuvent pas s'appliquer sur un même couple état-terme. Par conséquent, si $\delta_n(q, t)$ est défini, alors il l'est de manière unique. On montre ensuite aisément par récurrence sur n que pour tout terme t et tout état q , la suite $(\delta_n(q, t))_{n \in \mathbb{N}}$ est soit croissante (pour \sqsubseteq), soit non définie à partir d'un certain rang⁹. Dans le premier cas, on pose $\mathbf{t}_q(t)$ égale à $\sqcup_{n \in \mathbb{N}} \delta_n(q, t)$, dans le second, $\mathbf{t}_q(t)$ n'est pas défini. L'image d'un terme par un transducteur déterministe $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ est alors simplement $\mathbf{t}(t) = \mathbf{t}_{q_{init}}(t)$.

Exemple 3.43 Reprenons l'exemple 3.41. Il s'agit maintenant de simplifier un terme de $\mathcal{T}^\omega(N)$ tout en en préservant sa valeur. Ce résultat s'obtient symboliquement grâce à un transducteur déterministe. Considérons le transducteur déterministe $\mathbf{t} = (N, N, \{q_0, q\}, q_0, \delta)$ avec δ défini par :

$$\begin{array}{lll}
& q_0(x) \rightarrow \infty & \text{si } \llbracket x \rrbracket_N = \infty , \\
\text{sinon} & q_0(x) \rightarrow 0 & \text{si } \llbracket x \rrbracket_N = 0 , \\
\text{sinon} & q_0(x) \rightarrow q(x) , & \\
& q(x) \rightarrow 1 & \text{si } \llbracket x \rrbracket_N = 1 , \\
\text{sinon} & q(x + x') \rightarrow q(x) & \text{si } \llbracket x' \rrbracket_N = 0 , \\
\text{sinon} & q(x + x') \rightarrow q(x') & \text{si } \llbracket x \rrbracket_N = 0 , \\
\text{sinon} & q(x + x') \rightarrow q(x) + q(x') , & \\
\text{sinon} & q(x * x') \rightarrow q(x) & \text{si } \llbracket x' \rrbracket_N = 1 , \\
\text{sinon} & q(x * x') \rightarrow q(x') & \text{si } \llbracket x \rrbracket_N = 1 , \\
\text{sinon} & q(x * x') \rightarrow q(x) * q(x') . &
\end{array}$$

Certaines libertés sont prises ici par rapport à la définition originale. En voici la liste :

- les transitions ne sont plus « atomiques » et il faut les décomposer pour obtenir des règles compatibles avec la définition. Par exemple, la règle $q(x + x') \rightarrow q(x) + q(x')$ nécessite l'usage de trois états intermédiaires q' , r et r' et l'usage des quatre règles :

$$\begin{array}{lll}
q(x) & \rightarrow & q'(x) & \text{si } x \in ((> 1) + (> 1)) \\
q'(x) & \rightarrow & r(x) + r'(x) \\
r(x + x') & \rightarrow & q(x) \\
r'(x + x') & \rightarrow & q(x') ,
\end{array}$$

9. Il est possible de voir les éléments non définis comme égaux à un élément supplémentaire *Undef* supérieur à tous les autres termes. Dans ce cadre, la suite $\delta_n(q, t)$ est alors croissante sans restriction.

- les anticipations sont placées informellement à l'extérieur des règles et la contrainte qu'elles imposent est exprimée sémantiquement. Cela ne pose pas de problème puisque l'on a vu que les ensembles de termes s'évaluant respectivement à 0, à 1 et à ∞ sont μ -définissables (voir exemple 3.41),
- les anticipations font références à des sous-termes. Cela ne pose pas de problème d'après les propriétés de clôture des parties μ -définissables (propriété 3.42).
- les anticipations ne sont plus disjointes, mais l'usage de la construction `sinon` énonce les priorités relatives des transitions.

Le transducteur déterministe ainsi défini transforme tout terme infini en un terme équivalent fini. Ce terme fini est en quelque sorte normalisé puisqu'il ne contient ni somme avec des arguments nuls ni multiplication impliquant des termes s'évaluant en 0 ou en 1.

3.4.2 Normalisation

Un problème technique subsiste: il est possible qu'un terme produit par un transducteur déterministe contienne le symbole \perp : considérer par exemple la transition $q(f(x, y)) \rightarrow q(x)$, qui ne produit jamais de symbole, mais en consomme. Il est préférable d'éviter ce phénomène comme nous le montre le lemme 3.44. Nous verrons aussi comment se débarrasser de cette situation grâce au lemme 3.47.

Lemme 3.44 *Soit $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ un transducteur déterministe. Considérons la plus grande (au sens de l'inclusion) application R de $Q \times 2^{\mathcal{T}^\omega(\mathcal{F}')}$ dans $2^{\mathcal{T}^\omega(\mathcal{F})}$ telle que pour tout état q_0 et tout ensemble $E \subseteq \mathcal{T}^\omega(\mathcal{F}')$,*

$$R(q_0, E) = \begin{cases} \bigcap_{i \in [|f|]} R(q_i, (fi)^{-1}(E)) & \text{si } q_0(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta, \\ \bigcup_{i \in [n]} (L_i \cap R(r_i, E)) & \text{si les } q_0\text{-transitions sont les} \\ & q_0(x \in L_i) \rightarrow r_i(x) \text{ pour } i \in [n], \\ f(\mathbf{1}, \dots, \mathbf{1}, \underset{\substack{\uparrow \\ \text{ième position}}}{R(r, E)}, \mathbf{1}, \dots, \mathbf{1}) & \text{si } q_0(f(x_1, \dots, x_n)) \rightarrow r(x_i). \end{cases}$$

$$\text{Alors} \quad R(q_0, E) = \mathbf{t}_{q_0}^{-1}(\{t' \mid \exists t \in E, t' \sqsubseteq t\}).$$

Preuve. Découle des définitions. □

Corollaire 3.45 *En particulier, si $\mathbf{t}_{q_0}(\mathbf{1}) \subseteq \mathcal{T}^\omega(\mathcal{F}')$ (c.a.d. ne contient pas le symbole \perp) alors pour toute partie E de $\mathcal{T}^\omega(\mathcal{F}')$, on a*

$$R(q_0, E) = \mathbf{t}_{q_0}^{-1}(E) .$$

Corollaire 3.46 *Pour tout état $q_0 \in Q$, l'ensemble $\mathbf{t}_{q_0}^{-1}(\perp)$ est μ -définissable.*

Preuve. On peut toujours supposer qu'il existe une constante $a \in \mathcal{F}'$ qui n'est jamais produite par une transition du transducteur déterministe. Dans ce cas $R(q_0, \{a\}) = \mathbf{t}_{q_0}^{-1}(\perp)$. Dans le système d'équations du lemme 3.44, si l'on remplace E par $\{a\}$, alors le cas de la production se réduit à \emptyset . Si on se restreint aux équations accessibles à partir de $R(q_0, \{a\})$, elles ne sont plus qu'en nombre fini (une pour chaque état). Ce système se traduit donc directement en une formule du μ -calcul vectoriel $\nu x[q].\{x[q] = \phi[q] \mid q \in Q\}$ avec :

$$\phi[q] = \begin{cases} \emptyset & \text{si } q(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta \\ \bigcup_{i \in [n]} (L_i \cap x[r_i]) & \text{si les } q\text{-transitions sont les} \\ & q(x \in L_i) \rightarrow r_i(x) \text{ pour } i \in [n] \\ f(\mathbf{1}, \dots, \mathbf{1}, x_i[r], \mathbf{1}, \dots, \mathbf{1}) & \text{si } q(f(x_1, \dots, x_n)) \rightarrow r(x_i) \\ & \uparrow \\ & \text{ième position} \end{cases}$$

□

Le lemme suivant permet d'éliminer \perp de l'image d'un transducteur déterministe, et de se ramener de la sorte aux hypothèses du corollaire précédent.

Lemme 3.47 *Soient \mathbf{t} un transducteur déterministe de $\mathcal{T}^\omega(\mathcal{F})$ dans $\mathcal{T}^\omega(\mathcal{F}')$ et \perp' un nouveau symbole constante. Il existe un transducteur déterministe \mathbf{t}' tel que :*

- $\mathbf{t}'(\mathbf{1}) \subseteq \mathcal{T}^\omega(\mathcal{F}' \uplus \{\perp'\})$ (c.a.d. ne contient pas le symbole \perp , mais le symbole \perp'),
- pour tout $t \in \mathcal{T}^\omega(\mathcal{F})$, $\mathbf{t}'(t) = h(\mathbf{t}(t))$ ou $h(t')$ est le terme t' dans lequel tous les symboles \perp sont remplacés par \perp' .

Preuve. Il suffit d'ajouter un nouvel état q_\perp produisant un symbole \perp' , et (en ajoutant des états intermédiaires) de faire précéder chaque transition d'une anticipation vérifiant que le reste du calcul ne va pas produire \perp , et le cas échéant effectuant un branchement vers q_\perp . Cette anticipation est bien μ -définissable d'après le corollaire 3.46. □

3.4.3 Propriétés

Lemme 3.48 *L'image inverse d'une partie μ -définissable par un transducteur déterministe est effectivement μ -définissable.*

Preuve. Considérons un transducteur déterministe $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$.

On définit pour tout état q , et toute μ -formule ϕ sur \mathcal{F}' une μ -formule $I(q, \phi)$ sur l'alphabet \mathcal{F} . Cette construction s'effectue par récurrence sur la structure de ϕ . Les variables libres sont transformées simultanément, les variables libres de $I(q, \phi)$ sont les $\{x[q] \mid x \in FV(\phi), q \in Q\}$.

L'application I est définie comme suit :

1. Si q_0 est un état de production : $q_0(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta$, on pose

$$I(q_0, g(\phi_1, \dots, \phi_{|g|})) = \emptyset \quad (3.3)$$

pour tout $g \in \mathcal{F}'$, $g \neq f$

$$I(q_0, f(\phi_1, \dots, \phi_{|f|})) = I(q_1, \phi_1) \cap \dots \cap I(q_{|f|}, \phi_{|f|}) \quad (3.4)$$

$$I(q_0, \phi \cap \psi) = I(q_0, \phi) \cap I(q_0, \psi) \quad (3.5)$$

$$I(q_0, \phi \cup \psi) = I(q_0, \phi) \cup I(q_0, \psi) \quad (3.6)$$

$$\text{pour } \theta \in \{\mu, \nu\}, \quad I(q_0, \theta x. \phi) = \theta x[q_0]. \{x[q] = I(q, \phi) \mid q \in Q\} \quad (3.7)$$

2. Si q_0 est un état de consommation ou d'anticipation, on pose,

$$I(q_0, \phi) = \nu y[q_0]. \{y[q] = F(q, \phi) \mid q \in Q\} \quad (3.8)$$

avec $F(q, \phi)$ défini par :

- (a) si q est un état de production, on pose $F(q, \phi) = I(q, \phi)$.
- (b) si q est un état de consommation, $q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i) \in \delta$, on pose

$$F(q, \phi) = f(\mathbf{1}, \dots, \mathbf{1}, \underset{\substack{\uparrow \\ i\text{-ème position}}}{y[r]}, \mathbf{1}, \dots, \mathbf{1}) \quad (3.9)$$

- (c) q est un état d'anticipation, soient $q(x \in L_i) \rightarrow q_i(x)$ pour $i \in [n]$ les q -transitions. On pose :

$$F(q, \phi) = (L_1 \cap y[q_1]) \cup \dots \cup (L_n \cap y[q_n]) \quad (3.10)$$

On montre par récurrence sur la structure de ϕ que pour tout état q_0 et toute valuation γ de $FV(\phi)$:

$$\mathbf{t}_{q_0}^{-1}(\phi[\gamma]) = I(q_0, \phi)[\gamma'] ,$$

avec pour toute variable $x \in FV(\phi)$ et tout état $q \in Q$, $\gamma'(x[q]) = \mathbf{t}_q^{-1}(\gamma(x))$.

1. Si q_0 est un état de production: $q_0(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta$, on a:

$$\begin{aligned} \text{Pour } g \neq f, \mathbf{t}_{q_0}^{-1}(g(\phi_1, \dots, \phi_{|g|})[\gamma]) &= \emptyset && \text{def.} \\ &= I(q_0, g(\phi_1, \dots, \phi_{|g|}))[\gamma'] && (3.3) \end{aligned}$$

$$\begin{aligned} \mathbf{t}_{q_0}^{-1}(f(\phi_1, \dots, \phi_{|f|})[\gamma]) &= \mathbf{t}_{q_0}^{-1}(\phi_1[\gamma]) \cap \dots \cap \mathbf{t}_{q_0}^{-1}(\phi_{|f|}[\gamma]) && \text{def.} \\ &= I(q_0, \phi_1)[\gamma'] \cap \dots \cap I(q_0, \phi_{|f|})[\gamma'] && \text{hyp. rec.} \\ &= I(q_0, f(\phi_1, \dots, \phi_{|f|}))[\gamma'] && (3.4) \end{aligned}$$

$$\begin{aligned} \mathbf{t}_{q_0}^{-1}(\phi \cap \psi) &= \mathbf{t}_{q_0}^{-1}(\phi[\gamma]) \cap \mathbf{t}_{q_0}^{-1}(\psi[\gamma]) \\ &= (q_0, \phi)[\gamma'] \cap I(q_0, \psi)[\gamma'] && \text{hyp. rec.} \\ &= I(q_0, \phi \cap \psi)[\gamma'] && (3.5) \end{aligned}$$

$$\begin{aligned} \mathbf{t}_{q_0}^{-1}(\phi \cup \psi) &= \mathbf{t}_{q_0}^{-1}(\phi[\gamma]) \cup \mathbf{t}_{q_0}^{-1}(\psi[\gamma]) \\ &= (q_0, \phi)[\gamma'] \cup I(q_0, \psi)[\gamma'] && \text{hyp. rec.} \\ &= I(q_0, \phi \cup \psi)[\gamma'] && (3.6) \end{aligned}$$

Cas de $\mu x.\phi$. Soient r_1, \dots, r_k les états du transducteur déterministe. Il s'agit d'une application directe du lemme de transfert 3.30 avec

$$\begin{aligned} g(E) &= \phi[\gamma, x \rightarrow E] \\ f(E_{r_1}, \dots, E_{r_k}) &= (I(r_1, \phi)[\gamma'], \dots, I(r_k, \phi)[\gamma']) \\ &\quad \text{avec } \begin{cases} \gamma'(z[q]) = \mathbf{t}_q^{-1}(\gamma(z)) & \text{pour } z \in FV(\phi) \\ \gamma(x[q]) = E_q \end{cases} \\ \rho(E) &= (\mathbf{t}_{r_1}^{-1}(E), \dots, \mathbf{t}_{r_k}^{-1}(E)) \end{aligned}$$

On a bien $\rho \circ g = f \circ \rho$ d'après l'hypothèse de récurrence. L'application ρ est additive d'après la propriété 3.28. Le lemme de transfert 3.30 conclut.

Cas de $\nu x.\phi$. comme pour μ (remplacer additive par multiplicative).

2. Si q_0 est un état de consommation ou d'anticipation, on utilise le corollaire 3.45. Comme il s'agit d'un calcul de point fixe, on peut substituer certaines expressions par leur solution, ainsi, R est la plus grande solution de,

$$R(q, E) = \begin{cases} \mathbf{t}_q^{-1}(E) & \text{si } q \text{ est un état de production,} \\ \bigcup_{i \in [n]} (L_i \cap R(r_i, E)) & \text{si les } q\text{-transitions sont les} \\ & q(x \in L_i) \rightarrow r_i(x) \text{ pour } i \in [n], \\ f(\mathbf{1}, \dots, \mathbf{1}, R(r, E), \mathbf{1}, \dots, \mathbf{1}) & \text{si } q(f(x_1, \dots, x_n)) \rightarrow r(x_i). \\ & \quad \uparrow \\ & \text{ième position} \end{cases}$$

Si dans ces équations, on ne garde que celles pour lesquelles $E = \phi[\gamma]$ (elles sont en nombre fini), en appliquant le corollaire 3.8 et l'hypothèse de récurrence, on obtient

$$\mathbf{t}_q^{-1}(\phi[\gamma]) = (\nu y[q].\{y[q] = F(q, \phi) \mid q \in Q\})[\gamma].$$

□

Corollaire 3.49 *Le domaine de définition d'un transducteur déterministe est μ -définissable.*

Remarque 3.50 *L'image directe ne se comporte pas aussi bien que l'image inverse. En particulier, l'image directe d'un ensemble μ -définissable n'est pas nécessairement μ -définissable. Considérons par exemple sur l'alphabet $\mathcal{F} = \{f : 2, a : 1, c : 0\}$ le transducteur déterministe $(\mathcal{F}, \mathcal{F}, \{q\}, q, \{q(a(x)) \rightarrow f(q(x), q(x)), q(c) \rightarrow c\})$. Ce transducteur déterministe transforme les termes de la forme $a^n(c)$ en arbres équilibrés sur f et c de hauteur n . Ainsi, l'image de ce transducteur est l'ensemble des termes équilibrés sur f et c , et cet ensemble n'est pas μ -définissable. En empruntant la terminologie de la réécriture et des transducteurs finis, ceci provient du fait que les règles définissant les transducteurs sont linéaires à gauche, mais pas nécessairement à droite.*

Propriété 3.51 *Les transducteurs déterministes sont clos par composition.*

Preuve. Considérons deux transducteurs déterministes $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ et $\mathbf{t}' = (\mathcal{F}', \mathcal{F}'', Q', q'_{init}, \delta')$.

Comme habituellement, il s'agit de construire une forme d'automate produit. L'ensemble des états du nouveau transducteur déterministe est donc $Q \times Q'$. A partir d'un état (q, q') , les transitions se déroulent comme suit. Si une transition de production peut être effectuée depuis l'état q' par \mathbf{t}' , alors elle est effectuée par l'automate produit. Lorsque de l'état q' peut être effectuée une anticipation de L dans \mathbf{t}' , une anticipation similaire est effectuée, mais cette fois-ci avec le langage $\mathbf{t}_q^{-1}(L)$. Lorsque q est un état de production de \mathbf{t} et q' un état de consommation de \mathbf{t}' , et que le même symbole est à la fois produit et consommé, alors les deux transitions sont effectuées simultanément : le nouveau transducteur déterministe effectue une transition à vide. Dans les autres cas, c'est la transition issue de q qui est choisie.

On a donc $\mathbf{t}'' = (\mathcal{F}, \mathcal{F}'', Q \times Q', (q_{init}, q'_{init}), \delta'')$ avec δ'' défini par :

$$\begin{aligned} \delta'' &= \{(q, q')(x) \rightarrow f((q, q'_1)(x), \dots, (q, q'_{|f|})(x)) \mid q'(x) \rightarrow f(q'_1(x), \dots, q'_{|f|}(x))\} \\ &\cup \{(q, q')(x \in \mathbf{t}_q^{-1}(L)) \rightarrow (q, r')(x) \mid q'(x \in L) \rightarrow r'(x) \in \delta'\} \\ &\cup \{(q, q')(f(x_1, \dots, x_{|f|})) \rightarrow (r, q')(x_i) \mid \begin{array}{l} q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i) \in \delta \\ \text{et } q' \text{ état de consommation} \end{array}\} \\ &\cup \{(q, q')(x \in L) \rightarrow (r, q')(x) \mid \begin{array}{l} q(x \in L) \rightarrow r(x) \in \delta \\ \text{et } q' \text{ état de consommation} \end{array}\} \\ &\cup \{(q, q')(x) \rightarrow (q_i, r')(x) \mid \begin{array}{l} q(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta \\ \text{et } q'(f(x_1, \dots, x_{|f|})) \rightarrow r'(x_i) \in \delta'\} \end{aligned}$$

On montre maintenant par récurrence sur l'entier n que pour tout \mathcal{F} -terme t , et tous états $q \in Q$ et $q' \in Q'$, il existe un entier k tel que :

1. si $\delta'_n(q', \mathbf{t}_q(t))$ n'est pas défini, alors $\delta''_k((q, q'), t)$ n'est pas défini,
2. sinon $\delta''_n((q, q'), t) \sqsubseteq \delta'_n(q', \mathbf{t}_q(t)) \sqsubseteq \delta''_k((q, q'), t)$.

□

3.4.4 Lien avec les systèmes d'équations

La propriété suivante permet de se servir des transducteurs déterministes comme d'un outil pour manipuler les systèmes d'équations.

Théorème 11 (Colcombet et Löding [CL04]) *Pour tous alphabets gradués \mathcal{F} et \mathcal{F}' et tout transducteur déterministe \mathbf{t} de $\mathcal{T}^\omega(\mathcal{F})$ dans $\mathcal{T}^\omega(\mathcal{F}')$, il existe effectivement une transduction monadique \mathbf{t}^* qui à chaque \mathcal{F} -système d'équations S associe un \mathcal{F}' -système d'équations $\mathbf{t}^*(S)$, tel que,*

$$\mathbf{t}(\text{Depliage}(S)) = \text{Depliage}(\mathbf{t}^*(S)) .$$

Preuve. Considérons un transducteur déterministe $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ dont l'image ne contient pas le symbole \perp . La construction s'effectue en deux temps. Dans le premier temps, on utilise un nouvel opérateur unaire i dont la sémantique est pour tout e , $i(e) = e$, *c.a.d.* l'identité. La deuxième étape consiste à éliminer ces opérateurs i .

Commençons par définir une transduction monadique $\mathbf{t}' T = (\delta, N, \phi, \psi)$.

Domaine de \mathbf{t}^ :* la transduction \mathbf{t}' est effectuée si $S \models \delta$ avec $\delta = \phi_D^* \{ \text{racine}/y \}$ où ϕ_D représente le domaine de \mathbf{t} sous forme d'une formule du μ -calcul, et ϕ_D^* la formule monadique équivalente obtenue par le théorème 9.

Ensemble de sommets de \mathbf{t}' : c'est $N \times V_S$ avec $N = Q \times \{0, 1\}$. On pose donc pour tout $(q, n) \in N$, $\phi_{q,n} = \text{vrai}$.

pour $f \in \mathcal{F}'$,

$$\psi_{g,(r,n),(r',n')}(x,y) = \begin{cases} x = y & \text{si } r = r', n = 0, n' = 1 \\ & \text{et } r(z) \rightarrow g(q_1(z), \dots, q_{|g|}(z)), \\ \text{faux} & \text{sinon.} \end{cases}$$

$$\psi_{i,(r,n),(r',n')}(x,y) = \begin{cases} x = y & \text{si } r = r', n = 0, n' = 1 \\ & \text{et } r \text{ est un état de production ou d'anticipation} \\ \text{faux} & \text{sinon.} \end{cases}$$

pour $k \in \mathbb{N}$,

$$\psi_{k,(r,n),(r',n')}(x,y) = \begin{cases} x = y & \text{si } r(z) \rightarrow g(q_1(z), \dots, q_{|g|}(z)), \\ & n = 1, n' = 0, k \in [|g|], r' = q_k, \\ \exists x'. x \xrightarrow{f} x' & \text{si } r(f(z_1, \dots, z_{|f|})) \rightarrow r'(z_i), \\ \quad \wedge x' \xrightarrow{i} y & n = 1, n' = 0, \\ x = y \wedge \Phi_L(x) & \text{si } r(z \in L) \rightarrow r'(z), \\ & n = 1, n' = 0, \\ \text{faux} & \text{sinon.} \end{cases}$$

FIG. 3.2 – Constructions de la transduction \mathbf{t}' du théorème 11.

Racine de \mathbf{t}' : c'est (*racine*^S, q_{init}), et on pose

$$\psi_{racine,(q,n)}(x) = \begin{cases} racine(x) & \text{si } q = q_{init}, \\ \text{faux} & \text{sinon.} \end{cases}$$

Relations de \mathbf{t}' : elles sont définies informellement par.

- si $q(x) \rightarrow f(q_1(x), \dots, q_{|f|}(x)) \in \delta$, alors pour tout sommet v de V_S , l'hyperarc $(v, q) = f((q_1, v), \dots, (q_{|f|}, v))$ est ajouté,
- si $q(x \in L) \rightarrow r(x) \in \delta$, alors pour tout sommet v de V_S tel que $S, v \models \Phi$, l'hyperarc $(q, v) = i((r, v))$ est ajouté, où Φ est la formule monadique obtenue de L par le théorème 10 page 71,
- si $q(f(x_1, \dots, x_n)) \rightarrow r(x_i) \in \delta$, alors pour tout hyperarc $v = f(v_1, \dots, v_{|f|})$ de S , l'hyperarc $(q, v) = i((r, v_i))$ est ajouté.

Formellement, cela correspond aux expressions présentées à la figure 3.2

On vérifie d'abord que le graphe obtenu par cette transduction suivie d'une restriction aux sommets accessibles est bien un système d'équations. En particulier, il y a exactement un opérateur d'après l'hypothèse de normalisation.

On montre alors par récurrence sur n que pour tout état q et tout sommet v ,

$$\delta_n(q, \text{Depliage}(S, v)) \sqsubseteq \bar{i}(\text{Depliage}(\mathbf{t}'(S), (q, v))), \quad (3.11)$$

où $\bar{i}(t)$ représente le terme t dans lequel les occurrences du symbole i ont été éliminées.

La deuxième étape consiste à éliminer les symboles i . Pour cela, on utilise la formule de logique monadique $\text{accessibilite}[(i1)^*f](x, y)$ qui teste s'il existe un chemin étiqueté par un mot du langage $(i1)^*f$ entre le sommet x et le sommet y . On applique alors l'interprétation monadique :

$$\mathcal{I} = \left| \begin{array}{ll} \text{racine}(x) & : \text{racine}(x) \\ x \xrightarrow{f} y & : \text{accessibilite}[(i1)^*f](x, y), \quad f \in \mathcal{F}' \\ x \xrightarrow{n} y & : x \xrightarrow{n} y, \quad n \in \mathbb{N} \end{array} \right|$$

On a alors la propriété pour tout \mathcal{F}' -système d'équations S

$$\text{Depliage}(\mathcal{I}(S)) = \bar{i}(\text{Depliage}(\mathbf{t}'(S))) .$$

Combiné avec l'équation (3.11), on obtient que pour tout \mathcal{F} -système d'équations,

$$\text{Depliage}((\mathcal{I} \circ \mathbf{t}')(S)) = \mathbf{t}(\text{Depliage}(S)) .$$

On pose donc $\mathbf{t}^* = \mathcal{I} \circ \mathbf{t}'$ □

Remarque 3.52 *Une certaine forme de réciproque à ce théorème est aussi vraie (cf. [CL04]) : si une transduction monadique T entre systèmes d'équations est telle que pour tout systèmes d'équations S et S' ,*

$$\text{Depliage}(S) = \text{Depliage}(S') \quad \Rightarrow \quad \text{Depliage}(T(S)) = \text{Depliage}(T(S')),$$

et si de plus T satisfait une certaine contrainte technique d'uniformité¹⁰ alors il existe un transducteur \mathbf{t} tel que $\mathbf{t} \circ \text{Depliage} = \text{Depliage} \circ T$.

10. Qui informellement stipule que seul le sommet *racine* du système d'équations résultat dépend de la position de la *racine* dans le premier système d'équations.

3.4.5 Utilisation des transducteurs déterministes

L'objectif est de disposer d'un lemme permettant d'établir des relations entre algèbres au moyen de transducteurs déterministes. On présente tout d'abord la notion de relation continue. Quelques considérations techniques sur l'arbre d'exécution d'un transducteur sont ensuite énoncées. Enfin, nous pourrions énoncer le résultat de correction (lemme 3.59).

Relations continues

Les transducteurs déterministes permettent de transformer des termes en termes. Il s'agit ici d'utiliser cet outil pour passer d'un cpo quelconque à un autre. Dans l'approche classique des transducteurs déterministes, c'est-à-dire quand ils servent à manipuler des termes finis, ceci correspond simplement à associer des invariants à chaque règle. Le principe de récurrence permet alors de conclure. L'exemple 3.53 montre (comme on peut s'y attendre) que ceci n'est plus suffisant en présence de termes infinis. Les relations continues présentées ici apportent un début de réponse à ce problème.

Exemple 3.53 *Considérons le transducteur déterministe \mathbf{t} reliant les N -termes aux N -termes suivant (pour simplifier on ne considère pas les symboles ∞ et $*$), possédant deux états q_{+1} (initial) et q_{id} :*

$$\begin{array}{ll} q_{+1}(t + t') \rightarrow q_{+1}(t) + q_{id}(t') , & q_{id}(t + t') \rightarrow q_{id}(t) + q_{id}(t') , \\ q_{+1}(0) \rightarrow 1 , & q_{id}(0) \rightarrow 0 , \\ q_{+1}(1) \rightarrow 1 + 1 , & q_{id}(1) \rightarrow 1 . \end{array}$$

On peut légitimement s'attendre à avoir la propriété $\llbracket \mathbf{t}(t) \rrbracket = \llbracket t \rrbracket + 1$ (). En effet, les invariants suivants sont vérifiés (ils correspondent syntaxiquement aux transitions associées à l'état q_{+1}) :*

$$\begin{array}{l} (n + n') + 1 = (n + 1) + n' , \\ 0 + 1 = 1 , \\ 1 + 1 = 1 + 1 . \end{array}$$

La propriété () est effectivement vérifiée pour tous les termes finis : une simple récurrence suffit. En revanche, si l'on considère le terme infini t solution de l'équation $t = t + 0$, il satisfait $\mathbf{t}(t) = t$, et donc $\llbracket t \rrbracket + 1 = 1 \neq \llbracket \mathbf{t}(t) \rrbracket$. Ainsi, (*) n'est pas satisfaite.*

Le propos de ce paragraphe est de dégager des conditions suffisantes pour assurer la correction de ce type de raisonnement. Le lemme 3.59 concrétise cette approche.

Dans le cas général, prouver qu'un transducteur déterministe \mathbf{t} effectue bien la tâche souhaitée s'exprime comme $R(\llbracket t \rrbracket, \llbracket \mathbf{t}(t) \rrbracket)$ pour tout terme t tel que $\mathbf{t}(t)$

est défini, où R est une relation fixée. En particulier, on ne peut pas se limiter aux relations de la forme $\llbracket t(t) \rrbracket = f(\llbracket t \rrbracket)$ pour une certaine fonction f : établir l'isomorphisme de graphes est un exemple de propriété dépassant ce cadre. Il s'agit donc tout d'abord de préciser quel type de relations il est légitime de considérer, et en particulier les hypothèses de continuité qui les concernent.

Soient trois cpos E_1 , E_2 et \mathcal{R} et deux applications ρ_1 et ρ_2 de respectivement \mathcal{R} dans E_1 et \mathcal{R} dans E_2 . On dit que $(\mathcal{R}, \rho_1, \rho_2)$ (ou simplement \mathcal{R} quand il n'y a pas d'ambiguïté) *relie continuellement* E_1 et E_2 si ρ_1 et ρ_2 sont continues. Pour R un élément de \mathcal{R} , e_1 et e_2 des éléments de respectivement E_1 et E_2 , on note $R(e_1, e_2) \in \mathcal{R}$ si $e_1 = \rho_1(R)$ et $e_2 = \rho_2(R)$ et qu'il n'y a pas d'ambiguïté sur ρ_1 et ρ_2 . On écrit $\mathcal{R}(e_1, e_2)$ s'il existe $R \in \mathcal{R}$ tel que $R(e_1, e_2)$.

Les propriétés de 3.54 à 3.58 énoncent les constructions qui permettent, par combinaison, d'obtenir toutes les relations continues du reste de ce travail. La première de ces propriétés concerne l'isomorphisme de structures.

Propriété 3.54 *Si E_1 et E_2 sont des cpos de structures avec quotient, alors on pose :*

$$\begin{aligned} \approx &= \{(\mathcal{S}_1, \Upsilon, \mathcal{S}_2) \mid \mathcal{S}_1 \in E_1, \mathcal{S}_2 \in E_2, \mathcal{S}_1 \approx_{\Upsilon} \mathcal{S}_2\} , \\ \rho_1(\mathcal{S}_1, \Upsilon, \mathcal{S}_2) &= \mathcal{S}_1 , \\ \rho_2(\mathcal{S}_1, \Upsilon, \mathcal{S}_2) &= \mathcal{S}_2 . \end{aligned}$$

L'ensemble \approx est ordonné par \subseteq défini comme suit :

$$(\mathcal{S}_1, \Upsilon, \mathcal{S}_2) \subseteq (\mathcal{S}'_1, \Upsilon', \mathcal{S}'_2) \quad \text{si} \quad \mathcal{S}_1 \subseteq \mathcal{S}'_1, \mathcal{S}_2 \subseteq \mathcal{S}'_2 \quad \text{et} \quad \Upsilon = \Upsilon' \cap (\mathcal{U}_{\mathcal{S}_1} \times \mathcal{U}_{\mathcal{S}_2}) .$$

Ainsi ordonné, \approx est un cpo et $(\approx, \rho_1, \rho_2)$ est une relation continue.

Propriété 3.55 (produit) *Soient E_1 , E_2 des cpos et soit une relation $\mathcal{R} \subseteq E_1 \times E_2$ telle que \mathcal{R} est clos par \sqcup , alors $(\mathcal{R}, \pi_1, \pi_2)$ où π_1 est la projection sur la première composante et π_2 sur la seconde composante, relie continuellement E_1 et E_2 .*

Corollaire 3.56 (applications continues) *Soient E_1 et E_2 des cpos et f une application continue de E_1 dans E_2 , alors (E_1, id, f) relie continuellement E_1 et E_2 .*

Propriété 3.57 (composition) *Soient E , E' et E'' trois cpos tels que E et E' sont reliés continuellement par $(\mathcal{R}, \rho_1, \rho_2)$ et E' et E'' sont reliés continuellement par $(\mathcal{R}', \rho'_1, \rho'_2)$, alors E et E'' sont reliés continuellement par $(\mathcal{R}'', \eta_1, \eta_2)$ défini par :*

$$\begin{aligned} \mathcal{R}'' &= \{(R, R') \mid R \in \mathcal{R}, R' \in \mathcal{R}', \rho_2(R) = \rho'_1(R')\} , \\ &\text{ordonné par } (R, R') \sqsubseteq (S, S') \text{ si } R \sqsubseteq S \text{ et } R' \sqsubseteq S' , \\ \eta_1(R, R') &= \rho_1(R) , \\ \eta_2(R, R') &= \rho'_2(R') . \end{aligned}$$

Propriété 3.58 (intersection) Soient $(\mathcal{R}, \rho_1, \rho_2)$ et $(\mathcal{R}', \rho'_1, \rho'_2)$ reliant continuellement E et E' . Soit

$$\begin{aligned}\mathcal{R}'' &= \{(R, R') \in \mathcal{R} \times \mathcal{R}' \mid \rho_1(R) = \rho'_1(R'), \rho_2(R) = \rho'_2(R')\}, \\ \eta_1(R, R') &= \rho_1(R), \\ \eta_2(R, R') &= \rho_2(R),\end{aligned}$$

alors $(\mathcal{R}'', \eta_1, \eta_2)$ relie continuellement E et E' .

Lemme de correction

Il nous reste à énoncer le lemme 3.59 exprimant la correction d'une transformation décrite par un transducteur déterministe. L'annexe A.1 précise ces constructions et donne les preuves manquantes.

Dans la suite, on appelle état (resp. transition) inévitable tout état (resp. toute transition) qui apparaît une infinité de fois dans chaque «branche de calcul infini» d'une exécution, et ce pour toutes les exécutions du transducteurs (cf. annexe A.1).

Soient un alphabet gradué \mathcal{F} et t^0 un terme de $\mathcal{T}^\omega(\mathcal{F})$. On note $\overleftarrow{t^0}$ l'ensemble $\{t \in \mathcal{T}_\perp^\omega(\mathcal{F}) \mid t \sqsubseteq t^0\}$. L'ensemble $\overleftarrow{t^0}$ muni de l'ordre \sqsubseteq est un cpo. Au sein de ce cpo, tout ensemble de termes admet une borne supérieure, notée \sqcup . Soit $T \subseteq \overleftarrow{t^0}$, on note $\sqcap T$ le plus grand terme t tel que pour tout $t' \in T$, $t \sqsubseteq t'$, c.a.d. $\sqcap T = \sqcup\{t \sqsubseteq t' \mid t' \in T\}$.

Lemme 3.59 (correction) Soit $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ un transducteur déterministe. Si

1. les transitions de consommation sont inévitables dans \mathbf{t} ,
2. pour tout $q \in Q$ et tout $t^0 \in \mathcal{T}^\omega(\mathcal{F})$, il existe un cpo $\mathcal{R}[q, t^0]$ qui relie continuellement $\overleftarrow{t^0}$ et $\mathcal{T}_\perp^\omega(\mathcal{F}')$; on note \perp_{q, t^0} son plus petit élément, ρ la projection sur $\overleftarrow{t^0}$ et ρ' la projection sur $\mathcal{T}_\perp^\omega(\mathcal{F}')$,
3. il existe un ensemble d'états $Q_\perp \subseteq Q$ inévitable dans \mathbf{t} tel que pour tout $q \in Q_\perp$ et tout $t^0 \in \mathcal{T}^\omega(\mathcal{F})$ on a $\rho'(\perp_{q, t^0}) = \perp$,
- a. pour toute transition $\gamma = q(x) \rightarrow g(r_1(x), \dots, r_{|g|}(x))$ de δ et tout terme $t^0 \in \mathcal{T}^\omega(\mathcal{F})$, il existe une application continue h_{γ, t^0} de $\mathcal{R}[r_1, t^0] \times \dots \times \mathcal{R}[r_{|g|}, t^0]$ dans $\mathcal{R}[q, t^0]$ telle que,

$$\begin{aligned}\text{pour } (R_1, \dots, R_{|g|}) &\in \mathcal{R}[r_1, t^0] \times \dots \times \mathcal{R}[r_{|g|}, t^0], \\ \sqcap\{\rho(R_1), \dots, \rho(R_{|g|})\} &\sqsubseteq \rho(h_{\gamma, t^0}(R_1, \dots, R_{|g|})), \\ \text{et } \rho'(h_{\gamma, t^0}(R_1, \dots, R_{|g|})) &= g(\rho'(R_1), \dots, \rho'(R_{|g|})),\end{aligned}$$

b. pour toute transition $\gamma = q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i)$ de δ et pour tout terme $t^0 = f(t_1^0, \dots, t_{|f|}^0) \in \mathcal{T}^\omega(\mathcal{F})$, il existe une application continue h_{γ, t^0} de $\mathcal{R}[r, t_i^0]$ dans $\mathcal{R}[q, t^0]$ telle que

$$\text{pour } R \in \mathcal{R}[r, t_i^0], \quad f(t_1^0, \dots, t_{i-1}^0, \rho(R), t_{i+1}^0, \dots, t_{|f|}^0) \sqsubseteq \rho(h_{\gamma, t^0}(R)), \\ \text{et } \rho'(h_{\gamma, t^0}(R)) = \rho'(R),$$

c. pour toute transition $\gamma = q(x \in L) \rightarrow r(x)$ de δ et tout terme $t^0 \in L$, il existe une application continue h_{γ, t^0} de $\mathcal{R}[r, t^0]$ dans $\mathcal{R}[q, t^0]$ telle que

$$\text{pour } R \in \mathcal{R}[q, t^0], \quad \rho(R) \sqsubseteq \rho(h_{\gamma, t^0}(R)), \\ \text{et } \rho'(h_{\gamma, t^0}(R)) = \rho'(R),$$

alors pour t^0 un \mathcal{F} -terme et $q \in Q$, si $\mathbf{t}_q(t)$ est défini alors $\mathcal{R}[q, t^0](t^0, \mathbf{t}_q(t^0))$.

Remarque 3.60 Dans la pratique on s'autorise à utiliser des transitions non atomiques. Ainsi, considérons une transition combinant consommation, production et anticipation de la forme suivante.

$$\text{si } x_1 \in L_1, \dots, x_{|f|} \in L_{|f|}, \quad q(f(x_1, \dots, x_{|f|})) \rightarrow u,$$

où u est un terme fini sur l'alphabet $\mathcal{F}' \uplus \{r(x_i):0 \mid r \in Q, i \in [|f|]\}$.

Soient $r_1(x_{\pi(1)}), \dots, r_n(x_{\pi(n)})$ les symboles de $\{r(x_i):0 \mid r \in Q, i \in [|f|]\}$ apparaissant dans u (chaque $r(x)$ apparaissant autant de fois que dans u). L'hypothèse permettant d'utiliser le lemme 3.59 s'exprime alors comme suit.

Pour tout $t^0 = f(t_1^0, \dots, t_{|f|}^0) \in \mathcal{T}^\omega(\mathcal{F})$, il existe une application continue h de $\mathcal{R}[r_1, t_{\pi(1)}^0] \times \dots \times \mathcal{R}[r_n, t_{\pi(n)}^0]$ dans $\mathcal{R}[q, t^0]$ telle que :

$$\text{pour } (R_1, \dots, R_n) \in \mathcal{R}[r_1, t_{\pi(1)}^0] \times \dots \times \mathcal{R}[r_n, t_{\pi(n)}^0], \\ \text{alors } f(\prod\{\rho(R_i) \mid \pi(i) = 1\}, \dots, \prod\{\rho(R_i) \mid \pi(i) = |f|\}) \sqsubseteq \rho(h(R_1, \dots, R_n)), \\ \text{et } \rho'(h(R_1, \dots, R_n)) = u\{\rho(R_i)/r_i(x_{\pi(i)})\},$$

où $u\{\rho(R_i)/r_i(x_{\pi(i)})\}$ représente le terme u dans lequel pour tout i , l'occurrence de $r_i(x_{\pi(i)})$ est remplacée par $\rho(R_i)$.

Cette hypothèse généralise celle de l'énoncé du lemme 3.59. Elle peut être décomposée en sous-règles correspondant à la décomposition de la transition en transitions atomiques. Le lemme 3.59 s'applique alors.

Remarque 3.61 Le lemme 3.59 permet d'établir des relations entre algèbres de termes. il peut aussi être utilisé pour mettre en relations deux algèbres quelconques. En effet, si $(\mathcal{R}, \rho_1, \rho_2)$ relie continuellement une \mathcal{F} -algèbre \mathbf{A} et une \mathcal{F}' -algèbre \mathbf{A}' , alors la relation $(\mathcal{R}', \rho'_1, \rho'_2)$ définie par

$$\mathcal{R}' = \{(t_1, R, t_2) \in \mathcal{T}_\perp^\omega(\mathcal{F}) \times \mathcal{R} \times \mathcal{T}_\perp^\omega(\mathcal{F}') \mid \rho_1(R) = \llbracket t_1 \rrbracket_{\mathbf{A}}, \rho_2(R) = \llbracket t_2 \rrbracket_{\mathbf{A}'}\} \\ \text{avec } (t_1, R, t_2) \sqsubseteq (t'_1, R', t'_2) \text{ si } t_1 \sqsubseteq t'_1, R \sqsubseteq R' \text{ et } t_2 \sqsubseteq t'_2, \\ \text{et } \rho'_1(t_1, R, t_2) = t_1, \quad \rho'_2(t_1, R, t_2) = t_2.$$

relie continuellement $\mathcal{T}_\perp^\omega(\mathcal{F})$ et $\mathcal{T}_\perp^\omega(\mathcal{F}')$. De plus, par définition $\mathcal{R}'(t_1, t_2)$ est vrai si et seulement si $\mathcal{R}(\llbracket t_1 \rrbracket_{\mathbf{A}}, \llbracket t_2 \rrbracket_{\mathbf{A}'})$ l'est. En appliquant ce principe, on peut directement utiliser le lemme de validité 3.59 pour établir des relations entre algèbres quelconques, et non plus uniquement entre algèbres de termes.

Exemple 3.62 Dans l'exemple 3.53, la construction ne fonctionne pas, ou en tout cas, ne produit pas l'effet espéré. Cela provient du fait que l'hypothèse 3 du lemme 3.59 n'est pas vérifiée : en particulier, pour aucun $n \in N$ on a $n + 1 = 0$, et il existe un cycle ne passant que par l'état q_{+1} .

Un transducteur \mathbf{t} permettant de remédier à ce problème est par exemple le suivant dont l'état initial est q_{+1} :

$$\begin{array}{ll}
\text{si } \llbracket t \rrbracket = 0, & q_{+1}(t) \rightarrow 1, \\
\text{si } \llbracket t \rrbracket = \infty, & q_{+1}(t) \rightarrow \infty, \\
\text{sinon} & q_{+1}(1) \rightarrow 1 + 1, \\
\text{sinon} & q_{+1}(t + t') \rightarrow q_{+1}(t) + q_{id}(t'), \\
& q_{id}(0) \rightarrow 0, \\
& q_{id}(1) \rightarrow 1, \\
& q_{id}(t + t') \rightarrow q_{id}(t) + q_{id}(t').
\end{array}$$

Toutes les relations continues utilisées ici sont de la forme $(\mathcal{R}, \rho, \rho')$ avec $\mathcal{R} \subseteq N \times N$, $\rho(n, n') = n$ et $\rho'(n, n') = n'$ (dans ce contexte, toute application est continue si et seulement si elle est croissante).

On peut alors point à point préciser quelles sont les hypothèse du lemme 3.59.

1. les transitions de consommations sont inévitables dans \mathbf{t} , car les seules transitions ne contenant pas de consommation se trouvent dans des feuilles de l'exécution,
2. on pose pour tout $n \in N$, $\mathcal{R}[q_{+1}, n] = \{(k, k+1) \mid 0 \leq k \leq n\}$ et $\mathcal{R}[q_{id}, n] = \{(k, k) \mid 0 \leq k \leq n\}$
3. on vérifie que $\{q_{id}\}$ est un ensemble inévitable (et en particulier aucune branche infinie d'exécution ne peut contenir une infinité d'états q_{+1} grâce aux deux premières transitions), et on a bien pour tout $n \in N$, $\mathcal{R}[q_{id}, n](0, 0)$,

a-c. les règles suivantes correspondent respectivement aux transitions de \mathbf{t} ,

$$\mathcal{R}[q_{+1}, 0](0, 1)$$

$$\mathcal{R}[q_{+1}, \infty](\infty, \infty)$$

$$\mathcal{R}[q_{+1}, 1](1, 2)$$

$$h_{q_{+1}, (n+n')} : (k, l), (k', l') \mapsto (k + k', l + l')$$

est une application continue de $\mathcal{R}[q_{+1}, n] \times \mathcal{R}[q_{id}, n']$ dans $\mathcal{R}[q_{+1}, n + n']$

$$\mathcal{R}_{id}(0, 0)$$

$$\mathcal{R}_{id}(1, 1)$$

$$h_{q_{id}, (n+n')} : (k, k'), (l, l') \mapsto (k + l, k' + l')$$

est une application continue de $\mathcal{R}[q_{id}, n] \times \mathcal{R}[q_{id}, n']$ dans $\mathcal{R}[q_{+1}, n + n']$.

Une preuve de correction pour un transducteur déterministe est donc délicate à formaliser. Dans la pratique, de nombreuses libertés sont prises. En particulier, ne pas préciser explicitement les applications continues. Ainsi, dans l'exemple précédent, on dira que si $\mathcal{R}[q_{+1}, n](k, l)$ et $\mathcal{R}[q_{id}, n'](k', l')$ alors $\mathcal{R}[q_{+1}](k + k', l + l')$ sans expliciter l'application $h_{q_{+1}, (n+n')}$. Comme déjà mentionné, ce raccourci omet la caractéristique de continuité de cette implication. Cependant, pour ne pas trop nuire à la lisibilité, cette manière de faire prévaut dans la suite de ce travail.

Chapitre 4

Graphes et structures sur les piles

Ce chapitre a pour but de présenter les structures définies sur les piles. Par structures définies sur les piles, on entend structure dont le codage le plus naturel est de considérer un ensemble de piles comme univers et des familles de relations «simples» construites par combinaison d'actions *dépiler* et *empiler*.

Cette définition, bien qu'informelle, capture un ensemble très précis de familles de structures dont les propriétés se distinguent clairement des familles de structures sur les termes (chapitres 5 et 6)

L'une des principales questions concernant ces structures est le choix d'une représentation. Trois grandes approches sont possibles :

- la représentation interne consiste à donner explicitement l'univers de la structure et à décrire séparément les relations définissant l'interprétation de chaque symbole;
- la représentation par systèmes d'équations consiste à décrire une famille de structures par une algèbre continue. Les structures de la famille sont alors exactement les solutions de systèmes d'équations sur les opérateurs de l'algèbre;
- la représentation par transformation] consiste à se donner une structure initiale et à s'autoriser un ensemble de transformations sur cette structure. La famille de structures décrite est alors l'ensemble des structures que l'on peut obtenir en appliquant à la structure initiale une ou plusieurs de ces transformations.

Chacune de ces représentations possède ses avantages et ses inconvénients. Ainsi, la représentation interne est la plus intuitive car elle correspond à la notion de machine : l'univers est l'ensemble des configurations de cette machine, et les relations sont un programme. La représentation par systèmes d'équations possède quant à elle l'avantage d'être facilement manipulable car les transformations algébriques d'équations sont un outil puissant. Enfin, la représentation par transformation procure la plupart des résultats de décidabilité sur les structures.

Chacune des familles de structures décrites dans la suite de ce chapitre possède éventuellement plusieurs représentations distinctes qui seront tour à tour étudiées. Établir les équivalences entre ces différentes représentations est l'objectif principal de ce chapitre.

Le paragraphe est décomposé comme suit. Suit la présentation des familles de graphes préfixes (4.2) puis des structures HR-équationnelles (4.3). Le paragraphe 4.4 sur les structures préfixes reconnaissables est le plus complexe car de nombreuses représentations de ces familles sont possibles.

4.1 Langages rationnels et systèmes d'équations

La théorie des langages se prête bien à la description en terme de système d'équations. Cela nous fournit des exemples plus simples que les graphes ou les structures étudiés par la suite. Nous présentons ici la hiérarchie de Chomsky sous cet angle. De nombreux ouvrages traitent de ce domaine comme [HU79].

4.1.1 Automates

Un jeu d'opérateurs simple permet de représenter fidèlement le comportement des automates non déterministes de mots. Pour un alphabet \mathbf{A} , les opérateurs $\text{Auto}[\mathbf{A}]$ (ou plus simplement Auto si aucune confusion n'en résulte) sont au nombre de trois :

- l'union de langages (notée $+$),
- le langage $\{\varepsilon\}$,
- la concaténation à gauche d'une lettre $a \in \mathbf{A}$ notée $a.$, associe à tout langage L le langage $a.L$ défini comme usuellement.

Bien entendu, le lien avec les langages rationnels est naturel.

Propriété 4.1 *Les langages rationnels sont les solutions des Auto-systèmes d'équations finis.*

4.1.2 Automates déterministes

Un automate déterministe quant à lui consomme systématiquement une lettre lors de chaque transition, et la destination de cette transition est entièrement déterminée par cette lettre. De plus, à partir de chaque état, l'automate peut accepter ou refuser le mot vide.

Ces remarques se traduisent en deux opérateurs sur les langages. Pour un alphabet $\mathbf{A} = \{a_1, \dots, a_n\}$, il y a deux opérateurs $\text{Auto}_d[\mathbf{A}]$ (ou plus simplement Auto_d)

- l'opération qui aux langages L_1, \dots, L_n associe le langage $a_1L_1 + \dots + a_nL_n$,

- l'opération qui aux langages L_1, \dots, L_n associe le langage $\varepsilon + a_1L_1 + \dots + a_nL_n$.

Par exemple, le langage rationnel $(ab)^*$ est la solution en X du Auto_d -système d'équations suivant :

$$\begin{aligned} X &= \varepsilon + aY + bZ , \\ Y &= aZ + bX , \\ Z &= aZ + bZ . \end{aligned}$$

On a alors le résultat classique suivant.

Propriété 4.2 *Les langages rationnels sont les solutions des Auto_d -systèmes d'équations finis.*

Précisons tout de même que, comme il s'agit d'automates en forme déterministe, il n'est pas possible de passer d'un Auto -système d'équation à un Auto_d -système d'équations par un transducteur déterministe (le contraire est en revanche vrai).

4.2 Les graphes préfixes

Les graphes préfixes sont la forme la plus simple de structures infinies que nous considérons dans ce travail. Ces graphes se décrivent par un ensemble rationnel de sommets et les relations arcs s'obtiennent par réécriture préfixe du mot représentant le sommet.

Définition 4.3 (graphe préfixe) *Un graphe est préfixe s'il est isomorphe au graphe de sommets V et dont les arcs étiquetés par a sont définis par*

$$\xrightarrow{a} = \{(uv, u'v) \in V^2 \mid (u, a, u') \in \Delta, v \in \Sigma^*\} \quad \text{avec}$$

- Σ un alphabet fini,
- V une partie rationnelle de Σ^* ,
- Δ une partie finie de $\Sigma^* \times A \times \Sigma^*$.

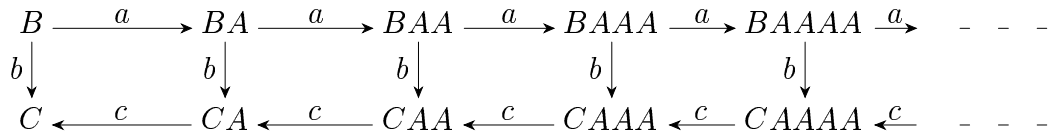


FIG. 4.1 - Un graphe préfixe : l'échelle.

Exemple 4.4 *La figure 4.1 donne un exemple simple de graphe préfixe : l'échelle. L'ensemble des étiquettes est $\{a, b, c\}$; l'alphabet est $\Sigma = \{A, B, C\}$; l'ensemble des sommets est $V = (B + C)A^*$; l'ensemble des règles de réécriture est*

$$\Delta = \{(B, a, BA), (B, b, C), (CA, c, C)\} .$$

Le premier travail s'intéressant aux graphes préfixes est dû à Müller et Schupp [MS85]. Leur formalisme est néanmoins différent pour deux raisons:

- les auteurs considèrent les graphes des transitions des automates à pile. Ainsi, l'ensemble des sommets n'est pas un ensemble de mots comme pour les graphes préfixes, mais un ensemble de configurations d'automate, c'est-à-dire de couples formés de l'état de cet automate et d'un mot de pile (sur un alphabet de pile donné);
- la seconde différence est la restriction qu'ils effectuent : ils ne considèrent comme sommets du graphe que les configurations accessibles à partir d'une configuration initiale donnée. Les graphes considérés sont donc nécessairement connexes.

En 90, Caucau [Cau90] étend cette définition originale à celle des graphes préfixes¹. Pour cela, il remarque qu'il n'est pas nécessaire de distinguer l'état de l'automate de la pile courante : on peut toujours coder cet état dans le sommet de pile. Ainsi, dans l'exemple 4.4 les lettres B et C jouent le rôle des états de l'automate alors que les mots sur A^* se comportent les mots de la pile. Il n'y a alors plus besoin que du mot de pile pour représenter une configuration et une transition de l'automate correspond à une réécriture préfixe sur ce mot.

La seconde amélioration concerne l'ensemble des sommets : il ne s'agit plus de l'ensemble des configurations accessibles à partir d'une configuration initiale, mais d'un ensemble rationnel. Il se trouve que l'ensemble des mots accessibles par réécriture préfixe à partir d'un mot donné est un ensemble rationnel (un résultat bien antérieur, dû à Büchi [Büc64]). On en déduit que les graphes préfixes contiennent les graphes considérés par Müller et Schupp. Cette extension est d'ailleurs stricte puisqu'il existe des graphes préfixes non connexes.

Cette extension du travail de Müller et Schupp se justifie pourtant aisément car elle préserve l'une des propriétés fondamentales qu'ils avaient mise en évidence.

Théorème 12 (Müller et Schupp [MS85]) *La théorie monadique des graphes préfixes est décidable.*

1. Notons que les graphes préfixes présentés dans ce travail se distinguent légèrement de la définition originale car ils peuvent contenir des sommets isolés alors que ceux-ci sont implicitement éliminés dans les travaux de Caucau.

Les propriétés suivantes donnent des méthodes de construction des graphes préfixes.

Propriété 4.5 *Les graphes préfixes contiennent*

- les graphes finis,
- les arbres réguliers de degré fini.

Les graphes préfixes sont clos par

- union disjointe,
- substitution finie inverse,
- modification finie,
- restriction rationnelle (lorsqu'ils sont enracinés),
- produit (synchrone ou asynchrone) avec un graphe fini,
- restriction à une composante connexe.

Enfin, il est important de rappeler l'une des propriétés fondamentales des graphes préfixes, c'est à dire leurs traces. En effet, les graphes préfixes étant issus des automates à pile, ils reconnaissent comme eux les langages algébriques.

Théorème 13 *Les traces des graphes préfixes entre un sommet initial et un sommet final sont les langages algébriques.*

L'inclusion des traces des graphes préfixes dans les langages algébriques est classique. La démonstration de la réciproque nécessite quant à elle une mise sous forme normale dite de Greibach de la grammaire. Nous ne présentons pas ici cette opération. Il s'agit néanmoins d'un grand classique de la théorie des langages et le lecteur intéressé peut se rapporter, entre autre, au livre de Hopcroft et Ullman [HU79], ou encore à celui de Mateescu et Salomaa [MS97].

Exemple 4.6 *Le graphe de la figure 4.1 a pour traces entre le sommet B et le sommet C le langage $\{a^n b c^n \mid n \in \mathbb{N}\}$. Ce langage est algébrique car il est la plus petite solution de l'équation $X = a.X.c + b$.*

Les graphes préfixes s'étendent naturellement aux structures en considérant des récritures faisant intervenir plus de deux mots. Nous n'explicitons pas ici cette extension; elle sera présentée de manière plus générale dans le contexte des graphes préfixes reconnaissables.

Étant donné un sommet d'un graphe préfixe, le nombre de sommets auquel il est susceptible d'être relié (par un arc avant ou retour) est borné par deux fois le nombre de règles de récriture (une fois pour les arcs avant et une fois pour les arcs retour). Les graphes préfixes sont donc de degré borné. La famille des structures HR-équationnelles lève cette restriction.

4.3 Les structures HR-équationnelles

Ce paragraphe est dédié à la présentation des structures dites HR-équationnelles, ainsi que des graphes correspondants. Nous en donnons une première approche intuitive à l'aide des grammaires de graphes (4.3.1). La définition de référence de cette famille est quant à elle donnée au paragraphe 4.3.2. Enfin, la présentation de ces graphes se termine par une étude de quelques-unes de leurs propriétés (4.3.3).

4.3.1 Première approche : grammaires de structures

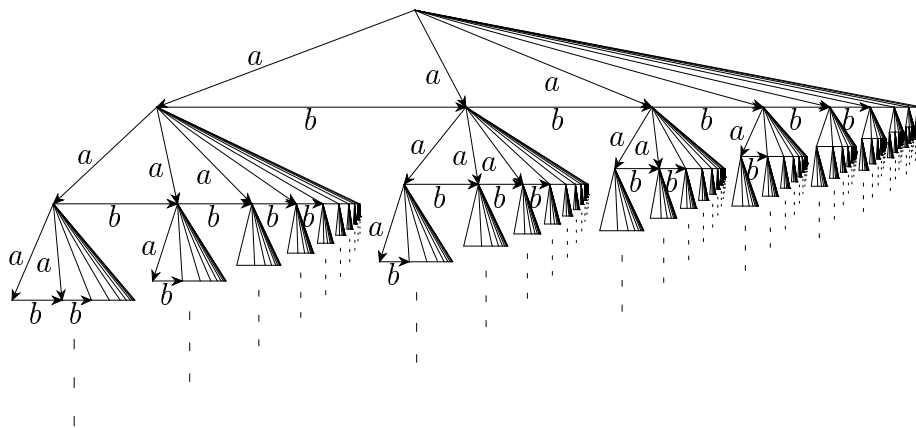


FIG. 4.2 – Graphe HR-équationnel.

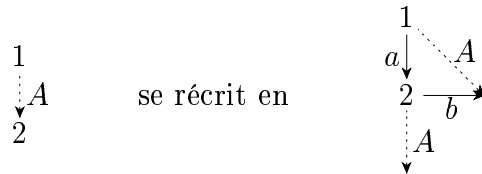
Les structures HR-équationnelles ont été proposées par Courcelle [Cou89]. Suivant les travaux, elles sont qualifiées simplement d'équationnelles, ou de régulières. Afin de lever toute ambiguïté, elles sont appelées dans le reste de ce travail structures HR-équationnelles. Cette terminologie fait référence à la définition 4.8 qui donne une caractérisation de la famille en termes de solutions de systèmes d'équations.

Ce n'est pourtant pas la première présentation de cette famille que nous donnons ici. En effet, il existe une autre définition classique de la même famille de structures : il s'agit d'une définition en termes de grammaires de structures (originellement d'hyper-graphes). Cette approche est assez délicate à formaliser mais possède l'avantage d'être intuitive. Nous la présentons au travers d'un exemple.

Il s'agit d'obtenir le graphe de la figure 4.2, c'est-à-dire l'arbre de degré infini étiqueté par a et augmenté d'une relation successeur entre frères étiquetée par b . Pour cela, on utilise un symbole relationnel intermédiaire (dit "non terminal") A d'arité 2 en plus des deux étiquettes a et b . Le graphe de la figure 4.2 est alors

obtenu comme limite de la suite de structures (H_n) définie par :

- H_0 est la structure à deux sommets en relation par A ,
- H_{n+1} se déduit de H_n par application parallèle de la règle de réécriture suivante :



c'est à dire en remplaçant chaque non terminal A de H_n par le motif correspondant (les sommets marqués par 1 et 2 sont ceux que la réécriture conserve, les autres sont insérés comme de nouveaux sommets).

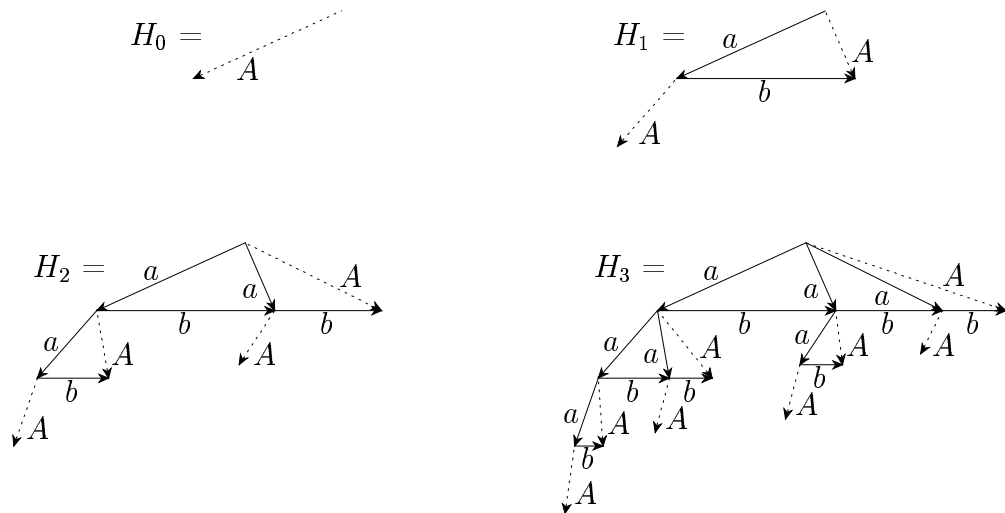


FIG. 4.3 – Structures successivement obtenues par réécriture.

La figure 4.3 présente les quatre premières structures de cette suite. Le non terminal A est représenté en pointillé comme s'il s'agissait d'un arc. Cette suite de graphes a pour limite le graphe de la figure 4.2.

L'approche présentée au travers de cet exemple se généralise sans difficultés aux structures et permet de définir la famille des structures dites HR-équationnelles. Le paragraphe suivant a pour objectif de donner une description équationnelle de cette famille de structures.

4.3.2 Représentation équationnelle

Dans le reste de ce travail, il n'est plus question de ces grammaires «de structures». Il leur est substitué les systèmes d'équations HR dont la définition est donnée ci-dessous. Cette définition nécessite de pouvoir fusionner des éléments de l'univers dans le système d'équations. Pour cela, on travaille sur le cpo des structures colorées avec quotient.

Pour C un ensemble fini de symboles unaires appelés *couleurs* et Σ un ensemble disjoint de symboles relationnels, une Σ -*structure colorée* par C est une structure dont la signature est $\Sigma \cup C$, et telle que les interprétations des $c \in C$ forment une partition de l'univers : une couleur c , et une seule, est attachée à chaque élément u de l'univers ; on dit que u a la couleur c . Un graphe coloré par C est similairement une structure possédant la signature d'un graphe augmentée de C . Lorsque l'on représente ces graphes colorés, les couleurs sont placées directement à l'emplacement du sommet.

L'intention est de se servir de ces couleurs pour identifier un ou plusieurs sommets dans les opérations de construction des structures.

Dans le cas des structures HR-équationnelles, on considère des structures colorées avec quotient.

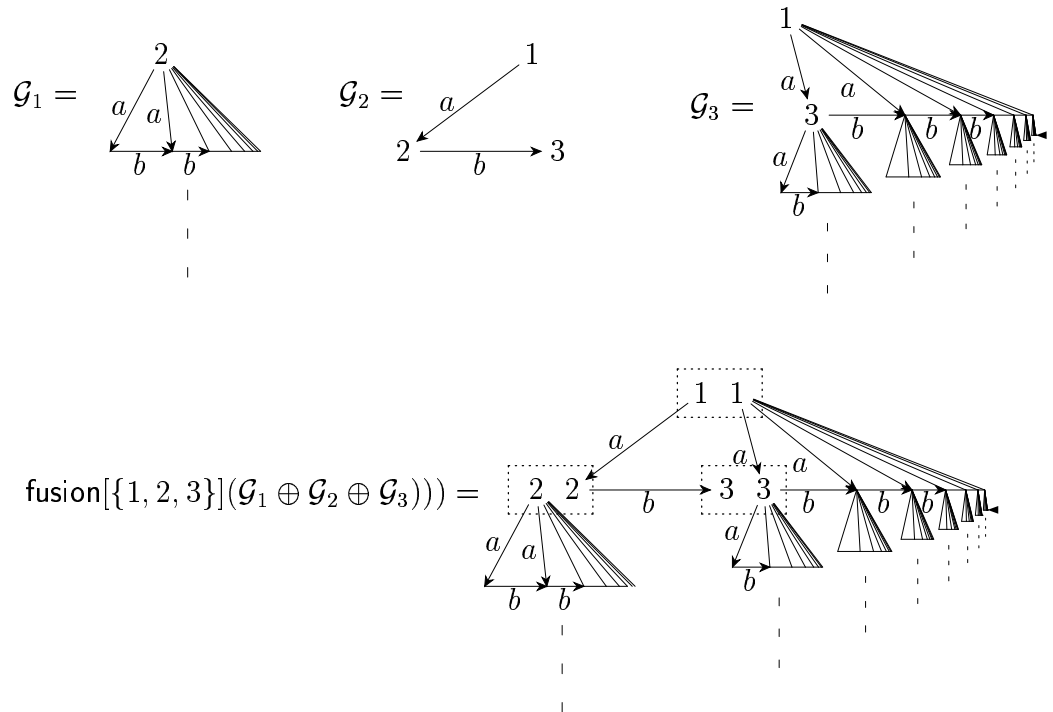


FIG. 4.4 – Union disjointe puis fusion (couleur 0 non représentée).

Les opérateurs $\text{HR}[\Sigma, C]$ (ou plus simplement HR) opèrent sur le cpo des Σ -structures colorées par C . Ce sont les suivants.

- Les structures finies colorées par C avec quotient, que l'on note simplement \mathcal{S} pour \mathcal{S} une structure finie.
- L'union disjointe \oplus avec sa sémantique habituelle.
- Le recoloriage par α , noté $\text{recol}[\alpha]$, où α est une application de C dans C . La sémantique du recoloriage laisse invariante la structure, si ce n'est que les éléments de l'univers de couleur c ont pour nouvelle couleur $\alpha(c)$:

$$\text{recol}[\alpha](\mathcal{S}) = \left[\begin{array}{lll} \alpha(c)(x) & : & c(x), & c \in C \\ R(x_1, \dots, x_{|R|}) & : & R(x_1, \dots, x_{|R|}) & R \in \Sigma \\ x \sim y & : & x \sim y & \end{array} \right] (\mathcal{S}) .$$

On utilise parfois la notation $\text{recol}[c_1 \rightarrow c_2]$, où c_1 et c_2 sont des couleurs, pour signifier $\text{recol}[\alpha]$, α étant défini par $\alpha(c) = c$ pour toute couleur $c \neq c_1$, et $\alpha(c_1) = c_2$.

- La fusion de couleur c_0 , notée $\text{fusion}[c_0]$, étend la relation d'équivalence de telle sorte que tous les sommets de couleur c_0 soient dans la même classe :

$$\text{fusion}[c_0](\mathcal{S}) = \left[\begin{array}{lll} c(x) & : & c(x), & c \in C \\ R(x_1, \dots, x_{|R|}) & : & R(x_1, \dots, x_{|R|}), & R \in \Sigma \\ x \sim y & : & x \sim y \vee (c_0(x) \wedge c_0(y)) & \end{array} \right] (\mathcal{S})$$

Pour $E = \{c_1, \dots, c_n\}$ un ensemble de couleur, $\text{fusion}[E]$ est un raccourci pour $\text{fusion}[c_1] \circ \dots \circ \text{fusion}[c_n]$.

Exemple 4.7 Soit $C = \{0, 1, 2, 3\}$. La figure 4.4 présente trois graphes \mathcal{G}_1 , \mathcal{G}_2 et \mathcal{G}_3 et le résultat du calcul $\mathcal{G} = \text{fusion}[\{1, 2, 3\}](\mathcal{G}_1 \oplus \mathcal{G}_2 \oplus \mathcal{G}_3)$. Les rectangles pointillés montrent comment les sommets sont fusionnés par les opérateurs fusion . Les graphes \mathcal{G}_1 et \mathcal{G}_3 s'obtiennent par recoloriage de \mathcal{G} . On obtient donc le système d'équations :

$$X = \text{fusion}[\{1, 2, 3\}](X_1 \oplus \mathcal{G}_2 \oplus X_3) ,$$

$$X_1 = \text{recol} \left[\begin{array}{l} 0 \rightarrow 0 \\ 1 \rightarrow 2 \\ 2 \rightarrow 0 \\ 3 \rightarrow 0 \end{array} \right] (X) , \quad X_3 = \text{recol} \left[\begin{array}{l} 0 \rightarrow 0 \\ 1 \rightarrow 1 \\ 2 \rightarrow 3 \\ 3 \rightarrow 0 \end{array} \right] (X) .$$

La plus petite solution de ce système d'équations est le graphe \mathcal{G} .

Définition 4.8 Les structures HR-équationnelles (resp. graphes équationnels) sont les structures (resp. graphes) solutions des HR-systèmes finis d'équations et élimination des couleurs.

Remarquons que dans cet énoncé, il n'est pas nécessaire de quotienter la structure par \sim car la notion d'isomorphisme traite la relation \sim de manière spécifique.

Telle qu'elle a été présentée ci-dessus, l'opération fusion semble compliquée, car elle fait intervenir un calcul de clôture réflexive et symétrique. En fait, cette opération peut être effectuée par une interprétation existentielle positive. Le lemme 4.9 donne une formalisation précise de ce fait.

Lemme 4.9 *Pour \mathcal{S} une Σ -structure colorée, $R \in \Sigma$ un symbole, $x_1, \dots, x_{|R|}$ des variables et γ une valuation de $x_1, \dots, x_{|R|}$ sur \mathcal{S} , on a*

$$\begin{aligned} \gamma, \text{fusion}[c_0](\mathcal{S}) \models R(x_1, \dots, x_{|R|}) \\ \text{ssi } \gamma, \mathcal{S} \models \bigvee_{E \subseteq [|R|]} \phi_{R,E} \\ \text{avec } \phi_{R,E} = \exists (z_i)_{i \in E}. R(z_1^E, \dots, z_{|R|}^E) \wedge \bigwedge_{i \in E} (c_0(x_i) \wedge c_0(z_i^E)) \\ \text{avec pour } i \in [|R|], z_i^E = \begin{cases} \text{variable fraîche} & \text{si } i \in E, \\ x_i & \text{sinon.} \end{cases} \end{aligned}$$

Preuve. Soit \mathcal{S} une Σ -structure colorée. Soit

$$\mathcal{S}' = \left\{ \begin{array}{lll} c(x) & : & c(x), & c \in C \\ R(x_1, \dots, x_{|R|}) & : & R(x_1, \dots, x_{|R|}), & R \in \Sigma \\ x \sim y & : & x \sim y \vee (c_0(x) \wedge c_0(y)) \end{array} \right\} (\mathcal{S})$$

Telle quelle, cette structure n'est pas bien formée car il n'y a aucune garantie a priori que la relation $\sim^{\mathcal{S}'}$ soit une équivalence et encore moins une congruence.

Montrons tout d'abord que $\sim^{\mathcal{S}'}$ est une relation d'équivalence. Symétrie et réflexivité sont héritées de la relation $\sim^{\mathcal{S}}$ et de la structure de la troisième règle. Soient u, v et w tels que $u \sim^{\mathcal{S}'} v$ et $v \sim^{\mathcal{S}'} w$. Deux cas sont à considérer, si $c_0^{\mathcal{S}}(v)$ alors $c_0^{\mathcal{S}}(u)$ et $c_0^{\mathcal{S}}(w)$ (propriété de $\sim^{\mathcal{S}}$). D'où par définition de $\sim^{\mathcal{S}'}$, $u \sim^{\mathcal{S}'} w$. Si pour un certain $c \neq c_0$, $c^{\mathcal{S}}(v)$, alors $u \sim^{\mathcal{S}} v$ et $v \sim^{\mathcal{S}} w$ (définition de $c^{\mathcal{S}'}$). Par transitivité de $\sim^{\mathcal{S}}$, $u \sim^{\mathcal{S}} w$ et donc $u \sim^{\mathcal{S}'} w$. Ainsi $\sim^{\mathcal{S}'}$ est une relation d'équivalence. Par conséquence, $\sim^{\mathcal{S}'} = \sim^{\overline{\mathcal{S}'}}$.

Supposons $R^{\overline{\mathcal{S}'}}(u_1, \dots, u_{|R|})$ avec pour tout i , $u_i = \gamma(x_i)$. Par définition de $\overline{\mathcal{S}'}$, il existe $v_1, \dots, v_{|R|} \in \mathcal{U}_{\mathcal{S}}$ tels que pour tout $i \in [|R|]$, $u_i \sim^{\mathcal{S}'} v_i$ et $R^{\mathcal{S}}(v_1, \dots, v_{|R|})$. Soit $E = \{i \mid \mathcal{S} \models c_0(u_i)\}$. Si $i \in E$ alors $\mathcal{S} \models c_0(u_i)$ et donc par définition de $\sim^{\mathcal{S}'}$, $\mathcal{S} \models c_0(v_i)$. Pour $i \notin E$, par définition de $\sim^{\mathcal{S}'}$, on a $u_i \sim^{\mathcal{S}} v_i$. On peut donc remplacer v_i par u_i pour tout $i \notin E$ dans $R^{\mathcal{S}}(u_1, \dots, u_{|R|})$. Ainsi, $\gamma, \mathcal{S} \models \phi_{R,E}$.

On montre similairement l'autre implication. \square

4.3.3 Propriétés des graphes HR-équationnels

Énonçons le résultat fondamental de décidabilité associé aux structures HR-équationnelles.

Théorème 14 (Courcelle [Cou89]) *Les solutions des HR-systèmes d'équations de théorie monadique décidable ont une théorie au second ordre gardée décidable.*

En fait, le résultat de Courcelle est plus général.

Théorème 15 (Courcelle [Cou89]) *Pour toute formule close ϕ du second ordre gardée, les modèles de ϕ sont HR- μ -définissables.*

La famille des structures HR-équationnelles contient strictement les graphes préfixes. Il est même possible d'en donner une caractérisation exacte.

Propriété 4.10 (Caucal [Cau90]) *Les graphes HR-équationnels de degré borné sont les graphes préfixes.*

Propriété 4.11 *Les structures HR-équationnelles sont closes par :*

- modification finie,
- union disjointe,
- restriction à une composante connexe,
- marquage rationnel,
- restriction rationnelle.

Nous ne donnons pas ici de preuve directe de ces résultats. Ils peuvent être déduits des résultats sur les graphes VR-équationnels.

En revanche, il manque une propriété de clôture simple aux structures HR-équationnelles : la clôture par interprétation au premier ordre (et en particulier par substitution finie inverse pour les graphes).

La propriété suivante s'intéresse à la largeur arborescente des graphes HR-équationnels. Cette même famille de graphe était définie en utilisant un jeu différent d'opérateurs.

Propriété 4.12 (Courcelle [Cou92]) *Les graphes solutions de HR-systèmes d'équations (éventuellement infinis) sont de largeur arborescente bornée.*

Preuve. Pour C un ensemble de couleurs et \mathcal{G} un graphe coloré, notons $LA_C(\mathcal{G})$ la largeur arborescente du graphe \mathcal{G} restreint aux sommets possédant une couleur dans C . On a alors les inégalités suivantes :

1. Pour tout ensemble fini C de couleurs, $LA_C(\perp) = 0$, le cas pathologique du graphe sans sommet a pour largeur arborescente 0 par convention.
2. Pour tous graphes colorés finis \mathcal{G} et \mathcal{H} et tout ensemble fini de couleurs C , $LA_C(\mathcal{G} \oplus \mathcal{H}) \leq \max(LA_C(\mathcal{G}), LA_C(\mathcal{H}))$. Il suffit pour cela de remarquer que dans un jeu du voleur et des policiers sur le graphe $\mathcal{G} \oplus \mathcal{H}$, le voleur choisit

une position de départ qui est soit dans \mathcal{G} soit dans \mathcal{G}' , et il ne pourra plus sortir de ce graphe en empruntant les arcs. Les policiers n'ont donc qu'à appliquer une stratégie gagnante sur ce graphe pour attraper le voleur.

3. Pour tout graphe coloré \mathcal{G} fini, tout ensemble fini C de couleurs et toute application α des couleurs dans les couleurs, $LA_C(\text{recol}[\alpha](\mathcal{G})) = LA_{\alpha^{-1}(C)}(\mathcal{G})$. Les deux membres de l'égalité correspondent au même jeu : seules les couleurs ont changé, et elles n'interviennent pas dans le jeu.
4. Pour \mathcal{G} un graphe et c_0 une couleur,

$$LA_C(\text{fusion}[c_0](\mathcal{G})) \leq \begin{cases} LA_{C-\{c_0\}}(\mathcal{G}) + 1 & \text{si } c_0 \in C \\ LA_C(\mathcal{G}) & \text{sinon} \end{cases}$$

Le cas où $c_0 \notin C$ est évident (et correspond en fait à une égalité). Si $c_0 \in C$, supposons que k policiers aient une stratégie gagnante sur \mathcal{G} privé des sommets de couleur c_0 . Alors, pour obtenir une stratégie gagnante sur $\text{fusion}[c_0](\mathcal{G})$, il suffit de rajouter un $(k + 1)$ -ème policier sur le sommet de couleur c_0 et d'utiliser la stratégie gagnante à k policiers sur les autres sommets.

Notons $C(\mathcal{G})$ les couleurs effectivement utilisées dans un graphe coloré \mathcal{G} . Naturellement $|C(\mathcal{G})|$ représente le nombre de couleurs distinctes apparaissant dans \mathcal{G} . Considérons maintenant un ensemble fini J d'opérateurs HR sur les couleurs C_0 , et dont les opérateurs graphes finis sont $\mathcal{H}_1, \dots, \mathcal{H}_n$. Posons $K = \max\{LA_{C_0}(\mathcal{H}_i) - |C(\mathcal{H}_i)| \mid i \in [n]\}$. On montre alors que pour tout J_\perp -terme fini t et pour tout ensemble de couleurs $C \subseteq C_0$,

$$LA_C(\llbracket t \rrbracket) \leq K + |C(\llbracket t \rrbracket)| \quad (*)$$

Ce résultat s'obtient par récurrence sur t , en utilisant la définition de K pour les graphes finis, et les inégalités 1 à 4 pour les autres opérateurs.

Considérons maintenant un jeu sur un graphe infini valeur d'un J -terme infini t . La première étape consiste, pour le voleur, à sélectionner un sous-graphe fini H sur lequel le jeu se déroule. Comme le graphe est décrit comme limite d'une suite croissante de graphes finis $(\mathcal{G}_n)_{n \in \mathbb{N}}$ (correspondante aux valeurs des J_\perp -termes finis de limite t), il existe un indice i tel que $H \subseteq \mathcal{G}_i$. Ce graphe est valeur d'un J_\perp -terme fini, donc, d'après $(*)$ $LA_{C_0}(H) \leq LA_{C_0}(\mathcal{G}_i) \leq K + |C(\mathcal{G}_i)| \leq K + |C_0|$.

Donc, la largeur arborescente du graphe $\llbracket t \rrbracket$ est inférieure à $K + |C_0|$. \square

Les opérateurs VR du paragraphe suivant se distinguent des opérateurs HR entre autre par le fait qu'ils ne possèdent plus cette propriété de largeur arborescente bornée.

4.4 Graphes préfixes reconnaissables et extensions

Les graphes préfixes reconnaissables sont une extension des graphes du paragraphe précédent. Bien des descriptions équivalentes de ces graphes sont connues. Elles se décomposent en représentations internes, par transformations et équationnelles.

Représentation interne. La première description de cette famille de graphes est due à Caucau : ce sont les graphes dont l'ensemble des sommets est rationnel, et les relations arcs sont définies par des règles de réécriture de mots décrites par des ensembles reconnaissables [Cau96].

Représentation par transformations. Caucau montre aussi que les mêmes graphes peuvent être déduits de l'arbre binaire complet par applications d'un marquage rationnel, d'une substitution rationnelle et d'une restriction rationnelle [Cau96]. Ces transformations sont des cas particuliers d'interprétations monadiques, dont se déduit la décidabilité de la théorie monadique de ces graphes. En fait, les graphes préfixes reconnaissables sont exactement ceux obtenables par interprétation monadique de l'arbre binaire complet [Bar97].

Représentation équationnelle. Barthelman donne une représentation équationnelle des graphes préfixes reconnaissables ; ce sont les solutions des systèmes d'équations finis sur un ensemble adapté d'opérateurs, appelés opérateurs VR [Bar97].

En fait, une autre direction d'extension est aussi possible. Il s'agit d'autoriser des systèmes d'équations infinis dans la représentation équationnelle. Cette extension peut être traduite dans les deux autres types de représentations. On obtient alors pour toute famille acceptable de systèmes d'équations Λ , l'équivalence des trois familles de graphes suivantes.

Représentation interne. Les graphes dont les sommets sont formés d'un ensemble de mots reconnus par un automate déterministe $\mathcal{A} \in \Lambda$, et dont les relations arcs sont représentées par des règles de réécriture suffixe décrites par des ensembles reconnaissables.

Représentation par transformations. Les interprétations monadiques du dépliage des graphes de Λ .

Représentation équationnelle. Les solutions de VR-systèmes d'équations de Λ . Cette triple équivalence coïncide avec la précédente quand Λ est l'ensemble des graphes finis. En fait, une seconde représentation équationnelle est établie dans ce chapitre, celle avec les graphes solutions de systèmes d'équations sur les opérateurs VR augmentés d'un opérateur de fusion (hérité des opérateurs HR).

Tous ces résultats s'adaptent relativement aisément aux structures. Ces extensions seront présentées tout au long du chapitre.

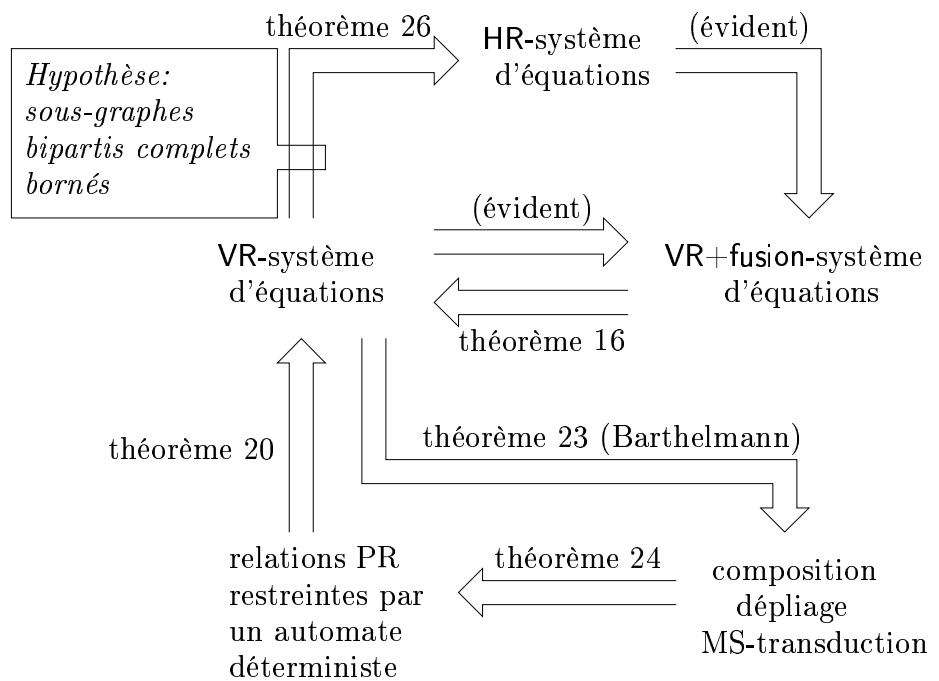


FIG. 4.5 – *Principaux résultats du paragraphe*

Enfin, une dernière équivalence structurelle nous intéresse. Il s'agit d'un résultat de Barthelmann énonçant que les graphes préfixes reconnaissables dont les sous-graphes bipartis complets sont bornés sont exactement les graphes HR-équationnels [Bar98]. Ce résultat devient dans le cadre présent : pour toute famille acceptable de systèmes d'équations Λ , les solutions de VR-systèmes d'équations dans Λ dont les sous-graphes bipartis complets sont bornés, sont les solutions des HR-systèmes d'équations de Λ . Ainsi, il est possible de caractériser uniformément et de manière externe les solutions des HR-systèmes au sein des solutions des VR-systèmes.

Tous ces résultats d'équivalence sont répartis dans différents théorèmes. Une carte des principales dépendances est donnée à la figure 4.5. Chaque flèche représente une inclusion des familles de graphes.

Le reste de ce paragraphe est construit comme suit. L'aspect équationnel est traité en premier (4.4.1). Le paragraphe 4.4.2 s'attache aux représentations internes de ces familles de structures. Les représentations par transformations sont ensuite abordées (4.4.3). Le paragraphe 4.4.4 est dédié aux liens entre ces familles de graphes et la hiérarchie de Caucal. Enfin, l'étude de ces familles s'achève par la preuve que les graphes dont les sous-graphes bipartis complets sont bornés solutions de VR-systèmes d'équations peuvent être traduits au moyen d'un transducteur déterministe en solutions de HR-systèmes d'équations.

4.4.1 Représentation équationnelle

L'objectif ici est de décrire les différents opérateurs permettant de représenter les graphes préfixes reconnaissables au moyen de systèmes finis d'équations. La finitude de ces systèmes n'étant jamais utilisée, ce paragraphe dépasse le simple cadre des graphes préfixes reconnaissables.

Nous commençons par la présentation des opérateurs VR qui travaillent sur le domaine des graphes (4.4.1.1). Le paragraphe 4.4.1.2 présente les opérateurs VR' qui sont une extension des opérateurs VR aux structures. Le paragraphe 4.4.1.2 présente un premier résultat de normalisation de ces opérateurs dont l'intérêt est principalement d'illustrer les techniques qui seront ensuite utilisées dans des cadres plus compliqués. Enfin, l'ajout d'un opérateur de fusion est envisagé au paragraphe 4.4.1.3. Celui-ci peut-être éliminé par un transducteur déterministe (théorème 16).

4.4.1.1 Opérateurs sur les graphes

Pour A un alphabet d'étiquettes et C un ensemble de couleurs, les opérateurs $\text{VR}[A, C]$ (ou plus simplement VR) opèrent sur les graphes colorés par C étiquetés

par A . Ils sont au nombre de quatre.

- L'union disjointe de graphes colorés, notée \oplus .
- Le graphe à un seul sommet de couleur c , noté $\text{sommet}[c]$. Pour $E \subseteq C$ un ensemble de couleurs, $\text{sommet}[E]$ est une abréviation représentant l'expression $\text{sommet}[c_1] \oplus \dots \oplus \text{sommet}[c_n]$, où $\{c_1, \dots, c_n\} = E$. Bien entendu, à isomorphisme près, la valeur de cette expression ne dépend pas du choix de numérotation des éléments de E .
- Le recoloriage par une application α des couleurs dans les couleurs, noté $\text{recol}[\alpha]$ qui a la même sémantique que pour les opérateurs HR.
- L'ajout d'arcs étiquetés par $a \in A$ entre les couleurs $c \in C$ et $c' \in C$. Il est noté $\text{arc}[c, a, c']$. Sa sémantique est de conserver le graphe inchangé, si ce n'est qu'un arc étiqueté par a est ajouté entre tout couple de sommets x, x' de couleurs respectives c et c' . Pour \mathcal{G} un graphe coloré,

$$\text{arc}[c, a, c'](\mathcal{G}) = \left| \begin{array}{l} c(x) \quad : \quad c(x) \quad \quad c \in C \\ x \xrightarrow{b} y \quad : \quad x \xrightarrow{b} y \quad \quad b \in A \\ x \xrightarrow{a} y \quad : \quad c(x) \wedge c'(y) \end{array} \right| (\mathcal{G}) .$$

Pour D une partie de $C \times A \times C$, $D = \{(c_1, a_1, c'_1), \dots, (c_n, a_n, c'_n)\}$, alors $\text{arc}[D]$ est une abréviation pour

$$\text{arc}[D](\mathcal{G}) = \text{arc}[c_1, a_1, c'_1](\dots \text{arc}[c_n, a_n, c'_n](\mathcal{G}) \dots) .$$

où $D = \{(c_1, a_1, c'_1), \dots, (c_n, a_n, c'_n)\}$. Là encore, cette notation ne dépend pas du choix de numérotation des éléments de D car les opérateurs arc commutent.

Les graphes obtenus comme plus petites solutions de VR-systèmes d'équations finis sont appelés graphes VR-équationnels.

Exemple 4.13 *La figure 4.6 présente un graphe coloré \mathcal{G} assimilable aux entiers positifs munis de la relation successeur étiquetée par a , de la relation \leq étiquetée par b , et coloré par 1 pour 0 et par 2 pour les autres entiers.*

On obtient le même graphe coloré \mathcal{G} en effectuant la succession d'opérations suivante à partir de \mathcal{G} :

1. *ajouter un nouveau sommet isolé de couleur 0,*
2. *ajouter un arc étiqueté par a entre le sommet de couleur 0 et le sommet de couleur 1,*
3. *ajouter un arc étiqueté par b du sommet de couleur 0 vers tous les autres sommets,*
4. *recolier le sommet de couleur 1 en couleur 2 et le sommet de couleur 0 en couleur 1.*

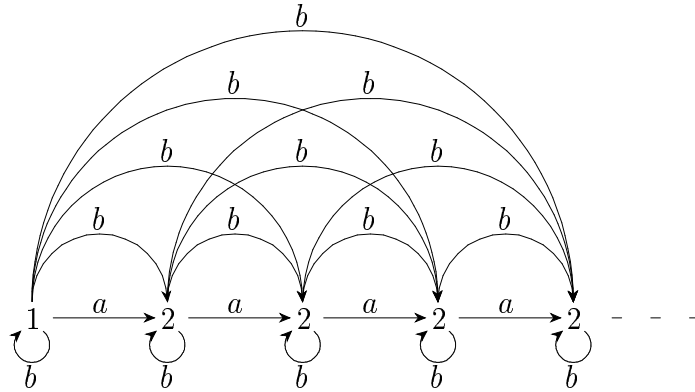


FIG. 4.6 – Les entiers munis de la relation successeur (a) et inférieur (b).

Ainsi, ce graphe est solution de l'équation :

$$\mathcal{G} = \text{recol} \begin{bmatrix} 0 & \rightarrow & 1 \\ 1 & \rightarrow & 2 \\ 2 & \rightarrow & 2 \end{bmatrix} (\text{arc}[\{(0, b, 0), (0, b, 1), (0, b, 2), (0, a, 1)\}](\mathcal{G} \oplus \text{sommet}[0])) .$$

Le paragraphe suivant montre comment il est possible d'étendre cette définition aux structures.

4.4.1.2 Extension aux structures

Le paragraphe précédent a permis de décrire des opérateurs sur les graphes infinis. Il s'agit ici d'étendre ces opérateurs aux structures. On construit ainsi les opérateurs VR' . Lorsque ces opérateurs servent à décrire des graphes, ils peuvent être traduits en opérateurs VR bien que nous n'établissions pas ce résultat (il s'agit d'un prolongement de la propriété 4.18).

Définition

L'extension des opérateurs VR aux structures se fait en unifiant les opérations de recolage et d'ajout d'arc : toutes deux sont des interprétations booléennes positives. Dans les opérateurs VR' , l'interprétation booléenne positive est une opération à part entière.

Pour Σ un alphabet de symboles relationnels, les opérateurs $\text{VR}'[\Sigma]$ (ou plus simplement VR') opèrent sur les Σ -structures. Ils sont au nombre de trois.

- La structure à un élément un , *c.a.d.* telle que $\mathcal{U} = \{0\}$ et l'interprétation de tous les symboles (sauf \sim) est vide. On s'autorise une variante $\text{un}[\mathcal{R}]$

où $R \in \Sigma$ est une symbole d'arité 1. Cette structure est égale à **un** si ce n'est que l'interprétation de R est $\{0\}$.

- L'union disjointe de structures, notée \oplus .
- L'interprétation booléenne positive notée comme l'interprétation elle-même.

Bien entendu, dans le cadre des graphes colorés, l'interprétation booléenne positive permet d'exprimer le recoloriage tout comme l'ajout d'arcs. Ainsi, les opérateurs \mathbf{VR}' sont naturellement plus expressifs que les opérateurs \mathbf{VR} .

Nous voyons maintenant comment montrer que \mathbf{VR} et \mathbf{VR}' coïncident sur les graphes colorés. Pour cela, il convient tout d'abord de normaliser les opérateurs \mathbf{VR}' . Tel est le propos du paragraphe suivant.

Normalisation

Dans ce paragraphe, nous montrons qu'il est possible de normaliser un système d'équations \mathbf{VR}' au moyen d'un transducteur déterministe (propriété 4.18). Cette forme normale est une restriction sur les formules booléennes positives autorisées dans les interprétations. Le rapport entre la version normalisée des opérateurs \mathbf{VR}' et les opérateurs sans restriction correspond au rapport entre automates non-déterministes et automates alternants. Ce résultat ne présente pas de difficulté particulière et peut être déduit d'autres constructions de ce travail. Néanmoins, cette preuve permet d'illustrer diverses techniques qui seront utilisées dans d'autres contextes ultérieurement.

Une *conjonction linéaire* est une formule booléenne positive formée uniquement de conjonctions et de prédicats relationnels, et dans laquelle chaque variable libre apparaît au plus une fois. Une formule booléenne est *en forme linéaire* si elle s'écrit comme une disjonction de conjonctions linéaires. Ainsi $a(x) \vee b(x)$ est en forme linéaire alors que $a(x) \wedge b(x)$ et $f(x, x)$ ne le sont pas. Une interprétation booléenne positive est en forme linéaire si toutes les formules qui la composent sont linéaires.

Une formule booléenne positive est *linéaire complète* vis à vis des variables $\{x_1, \dots, x_n\}$ si elle est linéaire, et que dans chacune des conjonctions apparaissent toutes les variables x_i . Ainsi, $a(x) \vee b(x)$ est linéaire complète vis à vis de $\{x\}$ mais ne l'est pas vis à vis de $\{x, y\}$. Remarquons que la formule **vrai** ne peut être mise sous forme linéaire complète que vis à vis de \emptyset alors que la formule **faux** l'est pour tout ensemble de variables. On dit qu'une interprétation booléenne positive est linéaire complète si pour toute règle $R(x_1, \dots, x_n) : \phi$, ϕ est linéaire complète vis à vis de $\{x_1, \dots, x_n\}$.

Il s'agit maintenant de transformer les \mathbf{VR}' -termes en \mathbf{VR} -termes dont les opérateurs interprétations sont linéaires complets. Soit donc Σ un ensemble fini de symboles relationnels, et soit N la plus grande des arités des symboles de Σ . Dorénavant, \mathcal{X} représente l'ensemble de variables du premier ordre $\{x_1, \dots, x_N\}$.

Cet ensemble est naturellement totalement ordonné par $x_i \leq x_j$ si $i \leq j$. De la sorte, pour tout ensemble fini de variables $X \subseteq \mathcal{X}$ de cardinal n , il existe une unique séquence de variables (y_1, \dots, y_n) strictement croissante telle que $X = \{y_1, \dots, y_n\}$. On note cette séquence \overline{X} .

Pour $X \subseteq \mathcal{X}$, $C(X)$ est l'ensemble des conjonctions booléennes positives dont les variables libres appartiennent à X . Modulo les invariants logiques usuels, $C(X)$ est fini. On introduit alors le nouvel ensemble de symboles relationnels suivant :

$$\Sigma' = \{ \langle X, \phi \rangle \mid X \subseteq \mathcal{X}, \phi \in C(X) \} \quad \text{avec } | \langle X, \phi \rangle | = |X| .$$

L'interprétation \mathcal{I}_0 des Σ -structures dans les Σ' -structures est définie comme suit,

$$\mathcal{I}_0 = \left| \langle X, \phi \rangle (\overline{X}) : \phi \right| .$$

On s'intéresse, pour une Σ -structure \mathcal{S} à construire une Σ -structure \mathcal{S}' telle que $\mathcal{I}_0(\mathcal{S}) = \mathcal{S}'$. Un problème subsiste : la relation entre \mathcal{S} et \mathcal{S}' ainsi définie ne relie aucun élément à la structure vide. En effet, $\perp \models \langle \emptyset, \text{vrai} \rangle$ et $\mathcal{I}_0(\perp) \neq \perp$. Cette relation ne permet pas de remplir les conditions de l'hypothèse 3 du lemme 3.59. Pour résoudre ce problème, on introduit l'interprétation booléenne positive \mathcal{I}'_0 définie comme suit.

$$\mathcal{I}'_0 = \left| \begin{array}{ll} \langle X, \phi \rangle (\overline{X}) & : \langle X, \phi \rangle (\overline{X}), & X \neq \emptyset, \phi \in C(X) \\ \langle \emptyset, \text{vrai} \rangle & : \text{vrai} \end{array} \right| .$$

La relation \mathcal{R} définie par $\mathcal{R}(\mathcal{S}, \mathcal{S}')$ si $\mathcal{I}_0(\mathcal{S}) = \mathcal{I}'_0(\mathcal{S}')$ est donc continue et satisfait $\mathcal{R}(\perp, \perp)$. Remarquons de plus que \mathcal{I}'_0 est linéaire complète.

Lemme 4.14 *Soit ϕ une formule booléenne positive sur la signature Σ de variables libres X . Il existe effectivement une formule ϕ^* linéaire complète vis à vis de X sur la signature Σ' telle que pour toute Σ -structure \mathcal{S} et toute valuation γ de X sur \mathcal{S} ,*

$$\gamma, \mathcal{S} \models \phi \quad \text{ssi} \quad \gamma, \mathcal{I}_0(\mathcal{S}) \models \phi^* .$$

Preuve. Il suffit de mettre la formule ϕ sous forme normale disjonctive $\phi = \phi_1 \vee \dots \vee \phi_k$ où les ϕ_k sont des conjonctions. On pose alors $\phi^* = \langle X, \phi_1 \rangle (\overline{X}) \vee \dots \vee \langle X, \phi_k \rangle (\overline{X})$. \square

Corollaire 4.15 *Pour toute interprétation booléenne positive \mathcal{I} des Σ -structures vers les Σ -structures, il existe une interprétation booléenne positive linéaire complète \mathcal{I}^* des Σ' -structures vers les Σ' -structures telle que :*

$$\mathcal{I}_0 \circ \mathcal{I} = \mathcal{I}^* \circ \mathcal{I}_0 .$$

Preuve. Il suffit d'appliquer le lemme 4.14 à chacune des formules booléennes positives apparaissant dans l'interprétation $\mathcal{I}_0 \circ \mathcal{I}$. \square

Lemme 4.16 *Il existe une interprétation booléenne positive linéaire complète \mathcal{I}_\oplus des Σ' -structures dans les Σ' -structures telle que pour toutes Σ -structures \mathcal{S}_1 et \mathcal{S}_2 ,*

$$\mathcal{I}_0(\mathcal{S}_1 \oplus \mathcal{S}_2) = \mathcal{I}_\oplus(\mathcal{I}_0(\mathcal{S}_1) \oplus \mathcal{I}_0(\mathcal{S}_2)) .$$

Preuve. Il s'agit d'une application du lemme 2.13. \square

Lemme 4.17 *Pour toute signature Σ , il existe un transducteur déterministe \mathbf{t}_0 des $\mathbf{VR}'[\Sigma]$ -termes dans les $\mathbf{VR}'[\Sigma']$ -termes à interprétations linéaires complètes tel que pour tout $\mathbf{VR}'[\Sigma]$ -terme t , on a :*

$$\llbracket \mathbf{t}_0(t) \rrbracket = \mathcal{I}_0(\llbracket t \rrbracket) .$$

Preuve. Comme déjà mentionné, $\mathcal{I}_0(\mathcal{S}) = \mathcal{I}'_0(\mathcal{S}')$ est continue et satisfait $\mathcal{I}_0(\perp) = \mathcal{I}'_0(\perp)$. Le corollaire 4.15 et le lemme 4.16 s'interprètent alors comme les hypothèses de récurrence démontrant la validité du transducteur déterministe suivant dont l'état initial est q_0 :

$$\begin{aligned} q_0(x) &\rightarrow \mathcal{I}'_0(q(x)) \\ q(x \oplus y) &\rightarrow \mathcal{I}_\oplus(\mathcal{I}'_0(q(x)) \oplus \mathcal{I}'_0(q(y))) && \text{(cf. lemme 4.16)} \\ q(\mathcal{I}(x)) &\rightarrow \mathcal{I}^*(\mathcal{I}'_0(q(x))) && \text{(cf. corollaire 4.15)} \\ q(\text{un}) &\rightarrow \mathcal{S}' && \text{avec } \mathcal{S}' = \mathcal{I}_0(\text{un}) . \end{aligned}$$

\square

On peut dès lors énoncer la propriété de normalisation qui nous intéresse.

Propriété 4.18 *Pour Σ un alphabet relationnel, il existe un transducteur déterministe \mathbf{t} des $\mathbf{VR}'[\Sigma]$ -termes dans les \mathbf{VR}' -termes à interprétations linéaires complètes telle que pour tout $\mathbf{VR}'[\Sigma]$ -terme t ,*

$$\llbracket \mathbf{t}(t) \rrbracket = \llbracket t \rrbracket .$$

Preuve. D'après le corollaire 4.15 appliqué à l'interprétation identité sur la signature Σ , il existe une interprétation linéaire complète \mathcal{I}_0^{-1} tel que pour toute Σ -structure \mathcal{S} , $\mathcal{I}_0^{-1}(\mathcal{I}_0(\mathcal{S})) = \mathcal{S}$. On pose alors simplement $\mathbf{t} = \mathcal{I}_0^{-1} \circ \mathbf{t}_0$ (\mathbf{t}_0 est le transducteur déterministe du lemme 4.17). \square

4.4.1.3 Un opérateur de fusion supplémentaire

Il s'agit d'autoriser une opération supplémentaire de fusion (similaire à celle des opérateurs HR) aux opérateurs VR'. On montre alors que, au moyen d'un transducteur déterministe, cette opération peut être éliminée d'un système d'équations (théorème 16). Une conséquence de ce résultat est la classique inclusion à isomorphisme près des structures HR-équationnelles dans les structures VR'-équationnelles [Cau96].

Ce résultat est comparable à celui énoncé par Courcelle et Makowsky dans le contexte des structures finies [CM02]. La démonstration présente emprunte une partie des techniques de ce travail bien que la manière de conclure soit très différente. Une autre preuve est donnée dans [CC03] de ce résultat. Elle est moins générale que celle présentée dans ce mémoire qui se rapproche plus de celle de Courcelle et Makowsky.

Le reste de ce paragraphe consiste tout d'abord à présenter plus formellement les opérateurs VR' + fusion. On présente ensuite une famille de formules logiques dites «existentielles gardées». Suivent cinq lemmes énonçant le bon comportement de ces formules vis à vis des opérateurs VR'. Le paragraphe suivant s'attarde sur la récurrence proprement dite. Le théorème 16 est présenté dans le dernier paragraphe.

Les opérateurs VR' + fusion

Les opérateurs VR' + fusion travaillent sur des *structures colorées avec quotient*. Les opérateurs VR' n'affectent pas les couleurs. Les couleurs ne sont modifiées que par l'opération de recoloriage (héritée des opérateurs HR). L'opération de fusion (notée fusion) est paramétrée par une couleur (là encore, il s'agit de la même que pour les opérateurs HR). Pour Σ un alphabet relationnel et C un ensemble fini de couleurs, les opérateurs (VR' + fusion)[Σ, C] (ou plus simplement VR' + fusion) sont les suivants.

- La structure à un seul élément de couleur c , notée **sommet**[c],
- L'union disjointe de structures colorées avec quotient, notée \oplus ,
- L'interprétation booléenne positive notée \mathcal{I} pour pour $\mathcal{I} = (\mathbf{vrai}, \Sigma, (\phi_R)_{R \in \Sigma})$. Sa sémantique est définie comme suit.

$$\mathcal{I}(\mathcal{S}) = \left| \begin{array}{ll} c(x) & : c(x), & \text{pour } x \in C \\ R(x_1, \dots, x_{|R|}) & : \phi_R(x_1, \dots, x_{|R|}), & \text{pour } R \in \Sigma \\ x \sim y & : x \sim y \end{array} \right|.$$

Ainsi \mathcal{I} se comporte comme une interprétation sans restriction sur les symboles de Σ , tout en gardant les couleurs de C non modifiées.

- Le recoloriage par une application α de C dans C , noté **recol**[α] est défini comme pour les opérateurs HR.

La fusion des éléments d'une même couleur c , notée $\text{fusion}[c]$, a la même sémantique que pour les opérateurs HR.

Remarque 4.19 *Une autre opération naturelle de fusion peut être envisagée. Cette opération se sert d'un symbole relationnel unaire quelconque (et non d'une couleur) pour décrire la fusion. Savoir si le théorème 16 reste valable quand l'opération de fusion est de ce type est une question ouverte (la question correspondante pour les structures finies l'est tout autant [CM02]).*

Dorénavant, nous supposons la signature Σ et l'ensemble de couleurs C fixés. L'entier N est la plus grande arité présente dans Σ .

Formules existentielles gardées

La démonstration du théorème 16 repose sur l'étude d'une famille précise de formules logiques. On s'intéresse aux formules logiques au premier ordre existentielles (*c.a.d.* ne possédant pas de quantification universelle) sur les prédicats $C \cup \Sigma$. On cherche à rendre chaque variable de telles formules indissociable de sa couleur. Cela revient à associer une couleur aux variables libres et à préciser la couleur des autres variables lors de leur introduction par une quantification.

Soit X un ensemble de variables, une *garde de X* est une application g de X dans C . On définit la formule \bar{g} par

$$\bar{g} = \bigwedge_{x \in X} (g(x))(x) .$$

Ainsi, une garde permet d'attacher une couleur à chacune des variables libres d'une formule. Pour g une garde, x une variable et c une couleur, alors $g, x \rightarrow c$ représente la garde g' qui coïncide avec g sur son domaine et telle que $g'(x) = c$.

On ajoute aussi une construction permettant d'attacher une couleur aux variables liées. Une *formule existentielle avec quantifications gardées* est une formule ϕ sur Σ telle que toutes les quantifications existentielles qu'elle contient sont de la forme $\exists x:c.\psi$, où x est une variable et $c \in C$ une couleur. Une formule $\exists x:c.\psi$ a la même interprétation que $\exists x.c(x) \wedge \psi$. Une telle construction quantifie donc uniquement sur les sommets d'une certaine couleur.

Une *formule existentielle gardée* est une conjonction $\bar{g} \wedge \phi$ où ϕ est une formule existentielle avec quantifications gardées et g est une garde de $FV(\phi)$.

La propriété suivante montre que l'on peut toujours se limiter à l'utilisation de telles formules.

Propriété 4.20 *Toute formule existentielle sur les prédicats $\Sigma \cup C$ est équivalente à une disjonction de formules existentielles gardées.*

Preuve. Considérons une formule existentielle ϕ , on construit alors la formule ϕ^* par :

$$\phi^* = \bigvee_{g \in C^{FV(\phi)}} (\bar{g} \wedge \phi^g)$$

où ϕ^g pour g une garde de $FV(\phi)$ est définie récursivement par :

$$\begin{aligned} (R(x_1, \dots, x_{|R|}))^g &= R(x_1, \dots, x_{|R|}) & (\exists x.\phi)^g &= \bigvee_{c \in C} \exists x:c.\phi^{g, x \rightarrow c}, \\ (c(x))^g &= \begin{cases} \text{vrai} & \text{si } g(x) = c, \\ \text{faux} & \text{sinon} \end{cases} & (x \sim y)^g &= x \sim y, \\ (\phi \wedge \psi)^g &= \phi^g \wedge \psi^g, & (\phi \vee \psi)^g &= \phi^g \vee \psi^g, \\ \text{vrai}^g &= \text{vrai}, & \text{faux}^g &= \text{faux}. \end{aligned}$$

On montre aisément par récurrence sur la structure de ϕ que pour toute structure \mathcal{S} , toute garde g de $FV(\phi)$ et toute valuation γ de $FV(\phi)$ sur \mathcal{S} ,

$$\gamma, \mathcal{S} \models \bar{g} \wedge \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \bar{g} \wedge \phi^g.$$

Notons que ceci n'est vrai que parce que l'interprétation des couleurs forme une partition de l'univers. Les conclusions du lemme en découle. \square

Il s'agit enfin de trouver une mesure de la complexité des formules existentielles gardées permettant d'assurer qu'elles soient finies modulo invariants syntaxiques. Pour cela, on compte le nombre d'imbrication de quantifications existentielles gardées par une même couleur. Pour $c \in C$ une couleur, et pour ϕ une formule gardée par C , on définit l'entier naturel $l_c(\phi)$ comme suit:

$$\begin{aligned} l_c(\text{vrai}) &= 0 & l_c(\text{faux}) &= 0 \\ l_c(\phi \wedge \psi) &= \max(l_c(\phi), l_c(\psi)) & l_c(\phi \vee \psi) &= \max(l_c(\phi), l_c(\psi)) \\ l_c(R(x_1, \dots, x_{|R|})) &= 0 & l_c(\exists y:c'.\phi) &= \begin{cases} l_c(\phi) & \text{si } c' \neq c \\ l_c(\phi) + 1 & \text{sinon.} \end{cases} \end{aligned}$$

On pose $l_C(\phi) = \max\{l_c(\phi) \mid c \in C\}$.

Comportement des formules existentielles gardées vis à vis des opérateurs VR'+fusion

Nous montrons ici comment, pour chacun des opérateurs VR'+fusion, la satisfaction d'une formule existentielle gardée par une structure après application de

l'opérateur se ramène à la satisfaction de (combinaisons de) formules existentielles gardées sur la/les structures arguments.

Suivent donc quatre lemmes — un pour chaque opérateur sauf pour **sommet** qui n'en nécessite pas — formalisant ce principe. Ces constructions sont classiques si ce n'est la construction associée à l'opérateur fusion (lemme 4.22).

Lemme 4.21 (opérateur interprétation) *Soit \mathcal{I} une interprétation booléenne positive des structures colorées par C dans les structures colorées par C . Pour toute formule existentielle gardée ϕ il existe une formule existentielle gardée $\phi^{\mathcal{I}}$ de mêmes variables libres telle que pour toute Σ -structure \mathcal{S} et toute valuation γ de $FV(\phi)$ sur \mathcal{S} , on a :*

$$\gamma, \mathcal{I}(\mathcal{S}) \models \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \phi^{\mathcal{I}} .$$

De plus, si $l_C(\phi) \leq N$, alors $l_C(\phi^{\mathcal{I}}) \leq N$.

Preuve. Il suffit de remplacer chaque occurrence d'un prédicat $R(y_1, \dots, y_n)$ dans ϕ par sa définition dans \mathcal{I} . Cette opération ne change pas les imbrications de quantifications car \mathcal{I} ne contient pas de quantifications. \square

Lemme 4.22 (opérateur fusion) *Soit $c_0 \in C$ une couleur. Pour toute formule existentielle gardée $\bar{g} \wedge \phi$, il existe une formule existentielle à quantifications gardées ϕ_{g,c_0} telle que*

- $FV(\phi_{g,c_0}) = \{x \in FV(\phi) \mid g(x) \neq c_0\}$,
- pour toute Σ -structure \mathcal{S} et toute valuation γ de $FV(\phi)$ sur \mathcal{S} , on a

$$\gamma, \text{fusion}[c_0](\mathcal{S}) \models \bar{g} \wedge \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \bar{g} \wedge \phi_{g,c_0} ,$$

- si $l_C(\phi) \leq N$, alors $l_C(\phi_{g,c_0}) \leq N$.

Preuve. La formule ϕ_{g,c_0} est définie récursivement comme suit.

$$\begin{aligned} R(y_1, \dots, y_{|R|})_{g,c_0} &= \exists z_{p_1}:c_0 \dots \exists z_{p_k}:c_0. R(z_1, \dots, z_n) , \\ &\text{avec } \{p_1, \dots, p_k\} = \{p \mid g(y_p) = c_0\} , \\ &\text{et } z_i = \begin{cases} y_i & \text{si } g(y) \neq c_0 , \\ \text{variable fraîche} & \text{sinon} , \end{cases} \\ (\exists y:c.\phi)_{g,c_0} &= \begin{cases} \phi_{g,y \rightarrow c_0,c_0} & \text{si } c = c_0 , \\ \exists y:c.\phi_{g,y \rightarrow c,c_0} & \text{sinon} , \end{cases} \end{aligned}$$

$$\begin{array}{ll}
\mathbf{vrai}_{g,c_0} = \mathbf{vrai} & \mathbf{faux}_{g,c_0} = \mathbf{faux} \\
(\phi \wedge \psi)_{g,c_0} = \phi_{g,c_0} \wedge \psi_{g,c_0} & (\phi \vee \psi)_{g,c_0} = \phi_{g,c_0} \vee \psi_{g,c_0}
\end{array}$$

Cette construction élimine toutes les quantifications de variables associées à la couleur c_0 . De nouvelles sont simultanément réintroduites au niveau des prédicats relationnels.

Exemple 4.23 *Supposons $C = \{1, 2\}$ et $\Sigma = \{R\}$ avec $|R| = 2$, alors on a*

$$\begin{array}{ll}
(\exists x:1.R(x, x))_{\emptyset,2} & = \exists x:1.R(x, x) , \\
(\exists x:1.R(x, x))_{\emptyset,1} & = \exists z_1:1.\exists z_2:1.R(z_1, z_2) .
\end{array}$$

Ainsi, dans le premier cas, lorsque l'on fusionne une couleur n'apparaissant pas dans la formule, celle-ci est inchangée. Le second cas montre comment, lorsqu'une variable de la couleur fusionnée est utilisée en plusieurs endroits, alors autant de nouvelles variables fraîches sont introduites (ici z_1 et z_2), celles-ci étant reliées par le fait qu'elles ont la même couleur.

Le dernier cas montre comment il est convenable de réinsérer la quantification au niveau du prédicat relationnel. Cette fois-ci $g(x_1) = 2$ et $g(x_2) = 2$.

$$(\exists y:1.R(x_1, y) \wedge R(y, x_2))_{g,1} = (\exists z:1.R(x_1, z)) \wedge (\exists z:1.R(z, x_2)) .$$

La preuve exacte de la validité de cette construction consiste à montrer par récurrence sur la structure de ϕ que :

$$\gamma, \text{fusion}[c_0](\mathcal{S}) \models \bar{g} \wedge \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \bar{g} \wedge \phi_{g,c_0} .$$

Le lemme 4.9 traite le cas des prédicats relationnels. □

Lemme 4.24 (opérateur recol) *Soit α une application de C dans C . Pour toute formule existentielle gardée $\bar{g} \wedge \phi$, il existe une formule existentielle avec quantifications gardées ϕ_α telle que :*

- $FV(\phi) \subseteq FV(\phi_\alpha)$,
- Pour toute Σ -structure \mathcal{S} et toute valuation γ de $FV(\phi)$ sur \mathcal{S} , on a

$$\gamma, \text{recol}[\alpha](\mathcal{S}) \models \bar{g} \wedge \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \bigvee_{\alpha \circ g' = g} \bar{g}' \wedge \phi_\alpha .$$

- Si $l_C(\phi) \leq N$, alors $l_C(\phi_\alpha) \leq N$.

Preuve. On définit ϕ_α par récurrence sur ϕ :

$$\begin{aligned} R(x_1, \dots, x_{|R|})_\alpha &= R(x_1, \dots, x_{|R|}) & (\exists x : c.\phi)_\alpha &= \bigvee_{c' \in \alpha^{-1}(c)} \exists x : c' . \phi_\alpha, \\ \text{vrai}_\alpha &= \text{vrai} & \text{faux}_\alpha &= \text{faux} \\ (\phi \wedge \psi)_\alpha &= \phi_\alpha \wedge \psi_\alpha & (\phi \vee \psi)_\alpha &= \phi_\alpha \vee \psi_\alpha \end{aligned}$$

On montre par récurrence sur ϕ que pour toute garde g' de $FV(\phi)$, toute structure \mathcal{S} et toute valuation de $FV(\phi)$ sur \mathcal{S} , on a

$$\gamma, \text{recol}[\alpha](\mathcal{S}) \models \overline{\alpha \circ g'} \wedge \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \overline{g'} \wedge \phi_\alpha .$$

□

Lemme 4.25 (opérateur \oplus) *Pour toute formule existentielle gardée $\overline{g} \wedge \phi$, il existe un ensemble fini ϕ_g^\oplus de paires de formules existentielles gardées tel que :*

- pour tout $(\psi_1, \psi_2) \in \phi_g^\oplus$, $FV(\psi_1)$ et $FV(\psi_2)$ sont disjoints et $FV(\psi_1) \cup FV(\psi_2) \subseteq FV(\psi)$,
- pour toutes Σ -structures colorées \mathcal{S}_1 et \mathcal{S}_2 et toute valuation γ de $FV(\phi)$ sur $\mathcal{S}_1 \oplus \mathcal{S}_2$ on a, $\gamma, \mathcal{S}_1 \oplus \mathcal{S}_2 \models \phi$ si et seulement si il existe $(\psi_1, \psi_2) \in \phi_g^\oplus$ tel que

$$\begin{aligned} FV(\psi_1) &\subseteq \text{dom}(\pi_1 \circ \gamma), & \text{et} & \quad \pi_1 \circ \gamma, \mathcal{S}_1 \models \overline{g|_{FV(\psi_1)}} \wedge \psi_1, \\ \text{et} \quad FV(\psi_2) &\subseteq \text{dom}(\pi_2 \circ \gamma), & \text{et} & \quad \pi_2 \circ \gamma, \mathcal{S}_2 \models \overline{g|_{FV(\psi_2)}} \wedge \psi_2 . \end{aligned}$$

- si $l_C(\phi) \leq N$, alors pour tout $(\psi_1, \psi_2) \in \phi_g^\oplus$, $l_C(\psi_1) \leq N$ et $l_C(\psi_2) \leq N$.

Preuve. Par application du lemme 2.13. □

Lemmes de récurrence

Soit $\mathcal{X} = \{x_1, \dots, x_N\}$. Pour tout $X \subseteq \mathcal{X}$, $F_\exists(X)$ représente l'ensemble des formules existentielles avec quantifications gardées ϕ dont les variables libres appartiennent à X et telles que $l_C(\phi) \leq N$. Modulo les équivalences syntaxiques habituelles, $F_\exists(X)$ est fini. Pour $X \subseteq \mathcal{X}$, \overline{X} représente l'unique suite de variables x_{p_1}, \dots, x_{p_n} telle que p est strictement croissant et $\{x_{p_i} \mid i \in [n]\} = X$.

On construit une nouvelle signature Σ' comme suit.

$$\begin{aligned} \Sigma' &= \{ \langle X, g, \phi \rangle \mid X \subseteq \mathcal{X}, g \in C^X, \phi \in F_\exists(X), \} , \\ &\text{avec pour tout } \langle X, g, \phi \rangle \in \Sigma', \quad | \langle X, g, \phi \rangle | = |X| . \end{aligned}$$

On définit aussi les interprétations \mathcal{I}_0 et \mathcal{I}'_0 comme suit.

$$\mathcal{I}_0 = \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) : \overline{g} \wedge \phi \\ c(x) : c(x) \end{array} \right| ,$$

$$\mathcal{I}'_0 = \left| \begin{array}{l} \langle \emptyset, \emptyset, \text{vrai} \rangle () : \text{vrai} \\ \langle X, g, \phi \rangle (\overline{X}) : \langle X, g, \phi \rangle (\overline{X}) \end{array} \right| .$$

L'interprétation \mathcal{I}_0 annote sur la structure l'ensemble des interprétations des formules de $F_{\exists}(X)$ pour $X \subseteq \mathcal{X}$. L'interprétation \mathcal{I}'_0 quant à elle laisse toute Σ' -structure inchangée si ce n'est que le symbole $\langle \emptyset, \emptyset, \text{vrai} \rangle$ d'arité 0 a pour nouvelle valeur vrai. Remarquons que $\mathcal{I}_0(\perp) = \mathcal{I}'_0(\perp)$ et $\mathcal{I}'_0 \circ \mathcal{I}_0 = \mathcal{I}_0$.

Soit $E \subseteq C$ un ensemble de couleurs et \mathcal{S} une structure colorée par C alors $\mathcal{S}|_E$ dénote la structure induite de \mathcal{S} par les éléments de couleurs dans E , *c.a.d.*

$$\mathcal{S}|_E = \mathcal{S}|_{\{u \in \mathcal{U}^{\mathcal{S}} \mid \exists c \in E, c(u)\}} .$$

Le lemme 4.26 montre que cette opération commute avec les interprétations booléennes positives.

Lemme 4.26 *Pour \mathcal{I} une interprétation booléenne positive et $E \subseteq C$ un ensemble de couleurs, alors pour toute structure colorée \mathcal{S} ,*

$$\mathcal{I}(\mathcal{S})|_E = \mathcal{I}(\mathcal{S}|_E) .$$

Les relations sur lesquelles reposent la construction sont alors les suivantes :

$$\text{pour } E \subseteq C, \quad \mathcal{R}[E](\mathcal{S}, \mathcal{S}') \text{ ssi } \mathcal{I}_0(\mathcal{S})|_E \approx \mathcal{I}'_0(\mathcal{S}') .$$

La relation $\mathcal{R}[E]$ relie continuellement les Σ -structures colorées aux Σ' -structures. L'interprétation \mathcal{I}'_0 ne sert qu'à obtenir $\mathcal{R}[E](\perp, \perp)$. Remarquons que si $\mathcal{I}_0(\mathcal{S}) = \mathcal{S}'$, alors $\mathcal{R}[E](\mathcal{S}, \mathcal{S}')$.

L'équivalence

En annexe (A.2) se trouvent les lemmes A.5, A.6, A.8, A.9 et A.7 établissant la validité de chacune des étapes de récurrences. Techniquement, ces lemmes sont des applications directes des lemmes 4.21, 4.22, 4.24 et 4.25.

Le théorème suivant énonce la conclusion de toutes les constructions qui ont été présentées précédemment.

Théorème 16 *Pour tout ensemble fini de couleurs C et pour tout alphabet relationnel Σ , il existe un transducteur déterministe \mathbf{t} des $(\mathbf{VR}' + \text{fusion})[\Sigma, C]$ -termes dans les \mathbf{VR}' -termes tel que pour tout $(\mathbf{VR}' + \text{fusion})[\Sigma, C]$ -terme t , on a :*

$$\llbracket \mathbf{t}(t) \rrbracket \approx \llbracket t \rrbracket .$$

Preuve. Les lemmes A.5, A.6, A.8, A.9 et A.7 correspondent aux étapes de récurrence de la preuve de validité du transducteur déterministe \mathbf{t}_0 défini comme suit.

- l'ensemble des états est $\{q[E] \mid E \subseteq X\}$,
- l'état initial est $q[C]$,
- les transitions sont définies par

$$\begin{aligned}
q[E](x \oplus y) &\rightarrow f^{\oplus, E}(q[E](x) \oplus q[E](y)) , \\
q[E](\text{recol}[\alpha](x)) &\rightarrow f^{\text{recol}[\alpha], E}(q[E^{\text{recol}[\alpha]}](x)) , \\
q[E](\mathcal{I}(x)) &\rightarrow f^{\mathcal{I}, E}(q[E](x)) , \\
q[E](\text{fusion}[c_0](x)) &\rightarrow q[E](x) && \text{si } \llbracket x \rrbracket_{c_0} = 0 , \\
q[E](\text{fusion}[c_0](x)) &\rightarrow f^{\text{fusion}[c_0], E}(q[E - \{c_0\}](x)) && \text{sinon.}
\end{aligned}$$

Des lemmes A.5, A.6, A.8, A.9, A.7 et 3.59 se déduit la validité de ce transducteur. On a donc pour tout $(\mathbf{VR}' + \text{fusion})[\Sigma, C]$ -terme t , $\mathcal{R}[C](\llbracket t \rrbracket, \llbracket \mathbf{t}_0(t) \rrbracket)$, c'est-à-dire que $\mathcal{I}_0(\llbracket t \rrbracket) \approx \mathcal{I}'_0(\llbracket \mathbf{t}_0(t) \rrbracket)$.

Soit l'interprétation booléenne positive suivante:

$$\mathcal{I}_1 = \left| \begin{array}{l} R(\overline{X}) : \bigvee_{g \in C^X} \langle X, g, \phi \rangle (\overline{X}) , \quad X = \{x_1, \dots, x_{|R|}\} \end{array} \right|.$$

Alors pour toute Σ -structure colorée \mathcal{S} , on a $\mathcal{I}_1(\mathcal{I}_0(\mathcal{S})) = \mathcal{S}$. On pose donc $\mathbf{t} = \mathcal{I}_1 \circ \mathcal{I}'_0 \circ \mathbf{t}_0$, et \mathbf{t} satisfait les conclusions du théorème 16. \square

4.4.2 Représentation interne

Les graphes solutions des VR-systèmes d'équations admettent une représentation interne sous forme de réécriture de mots. Nous présentons d'abord les définitions originales de Caucal puis le résultat d'équivalence énoncé par Barthelmann: ces travaux se limitent au cas des systèmes finis d'équations. Ensuite, après reformulation, nous généralisons ce résultat en en donnant une forme restant valable pour des systèmes d'équations non finis et pour les opérations VR'.

4.4.2.1 Graphes préfixes reconnaissables

La première définition que nous donnons de ces graphes correspond à la simple extension des graphes préfixes à des ensembles rationnels de règles.

Définition 4.27 (graphe préfixe reconnaissable) *Un graphe est préfixe reconnaissable si il est isomorphe au graphe de sommets V et dont les arcs étiquetés par a sont définis par*

$$\overset{a}{\rightarrow} = \{(uv, u'v) \in V^2 \mid (U, a, U') \in \Delta, u \in U, u' \in U'\} \quad \text{avec}$$

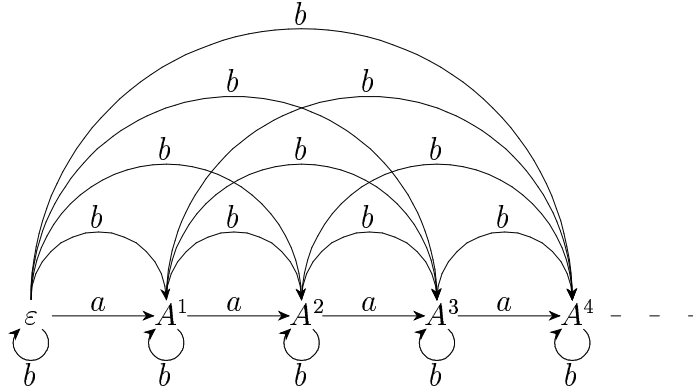


FIG. 4.7 – Un graphe préfixe reconnaissable.

- Σ un alphabet fini,
- V une partie rationnelle de Σ^* ,
- Δ un ensemble fini de triplets (U, a, U') où U et U' sont des parties rationnelles de Σ^* et a est une étiquette.

On peut noter pour plus de simplicité les relations de ce graphe par :

$$\xrightarrow{a} = \left(\bigcup_{(U,a,U') \in \Delta} (U \times U') \Sigma^* \right) \Big|_V .$$

Exemple 4.28 La figure 4.7 donne un exemple de graphe préfixe reconnaissable. Il s'agit de la demi-droite étiquetée par a et augmentée de la relation d'accessibilité étiquetée par b (c'est à dire les entiers munis de la relation successeur et de la relation \leq).

L'alphabet est $\Sigma = \{A\}$, l'ensemble de sommets est $V = A^*$ et

$$\Delta = \{(\varepsilon, a, A), (\varepsilon, b, A^*)\} .$$

Pour décrire un graphe préfixe reconnaissable il peut être pratique d'utiliser des règles gardées de réécriture. On peut ainsi utiliser un ensemble rationnel de suffixes autorisés pour chaque règle. Cette extension n'enrichit pas la famille de graphes comme le formalise le théorème suivant.

Théorème 17 (Caucal [Cau96]) *Un graphe est préfixe reconnaissable si et seulement si il est isomorphe au graphe de sommets V et dont les arcs étiquetés par a sont définis par*

$$\xrightarrow{a} = \{(uw, u'w) \mid (U, a, U', W) \in \Delta, u \in U, u' \in U', w \in W\} \text{ avec}$$

- Σ un alphabet fini,

- V une partie rationnelle de Σ^* ,
- Δ un ensemble fini de quadruplets (U, a, U', W) où a est une étiquette et U, U' et W sont des parties rationnelles de Σ^* .

Remarquons que cette équivalence n'est valable qu'à isomorphisme près. Comme pour la définition originale des préfixes reconnaissables, on peut donner une notation compacte de ce type de réécriture:

$$\xrightarrow{a} = \bigcup_{(U, a, U', W) \in \Delta} (U \times U')W .$$

Le résultat fondamental de décidabilité pour la famille des graphes préfixes reconnaissables est donné par le théorème 18.

Théorème 18 (Caucal [Cau96]) *La théorie monadique des graphes préfixes reconnaissables est décidable.*

Pour établir ce résultat, Caucal montre que ces graphes s'obtiennent pas interprétation monadique de l'arbre binaire complet. Nous empruntons un chemin similaire dans la suite de ce travail.

Théorème 19 (Barthelmann [Bar97]) *Les graphes préfixes reconnaissables sont les graphes solutions de VR-systèmes finis d'équations.*

Dans ce travail, nous étendons ce résultat aux structures (ce qui a déjà été considéré par Barthelmann [Bar97] et Blumensath [Blu01]), et surtout en autorisant la restriction du graphe préfixe reconnaissable par un langage reconnu par un automate infini. Dans ce paragraphe, une moitié de ce résultat est énoncé; il s'agit du théorème 20. La réciproque est montrée ultérieurement en passant par une étape intermédiaire (plus précisément en appliquant le théorème 23 puis le théorème 24).

4.4.2.2 Les relations PR

Il s'agit ici de généraliser la notation $(U \times V)W$ pour représenter cette fois-ci des relations n -aires sur les mots. On fait aussi le choix, pour des raisons d'homogénéité avec le reste de ce travail, de ne plus considérer des relations «préfixes reconnaissables» [Cau96], mais plutôt ce qui devrait symétriquement s'appeler des relations suffixes reconnaissables, *c.a.d.* correspondant à la notation $U(V \times W)$. Bien entendu, ces deux approches produisent des graphes isomorphes (pourvu que l'on ne mélange pas ces deux types de règles dans un même graphe). Les relations ainsi définies sont néanmoins appelées relations PR (pour préfixe reconnaissable) dans le reste de ce travail.

Les relations PR ont été introduites pour la première fois dans [CC03]. Une notion similaire a été présentée antérieurement par Blumensath [Blu01] ou encore

dans [NT01]. La définition peut aussi être attribuée à Caucau [Cau96] qui utilisait déjà la notation $(U \times V)W$ pour représenter les relations élémentaires dont sont formés les graphes préfixes reconnaissables.

Soit \mathbf{A} un alphabet fini, soit N son cardinal et soient a_1, \dots, a_N les lettres qui le composent. Les relations PR sur \mathbf{A} sont définies par récurrence comme suit :

- la relation non vide $\{()\}$ d'arité 0 est une relation PR,
- le singleton $\{\varepsilon\}$ est une relation PR d'arité 1 notée ε ,
- soit R une relation PR d'arité n et L un langage rationnel sur \mathbf{A}^* , la relation $L.R$ est une relation PR d'arité n définie par

$$L.R = \{(uv_1, \dots, uv_n) \mid u \in L, (v_1, \dots, v_n) \in R\},$$

- l'union de deux relations PR de même arité est une relation PR,
- soit R une relation PR d'arité n et R' une relation PR d'arité n' , la relation $R \times R'$ est une relation PR d'arité $n + n'$ définie par

$$R \times R' = \{(u_1, \dots, u_n, v_1, \dots, v_{n'}) \mid (u_1, \dots, u_n) \in R, (v_1, \dots, v_{n'}) \in R'\},$$

- soit R une relation PR d'arité n et σ une permutation de $[1, n]$, alors $\sigma(R)$ est une relation PR d'arité n définie par

$$\sigma(R) = \{(u_1, \dots, u_n) \mid (u_{\sigma(1)}, \dots, u_{\sigma(n)}) \in R\}.$$

Exemple 4.29 Les langages rationnels sont des relations PR d'arité 1 car un tel langage L s'écrit $L.\varepsilon$. On écrit alors simplement L à la place de $L.\varepsilon$. La relation vide d'arité $n \geq 1$ est PR car elle s'écrit $\emptyset.(\varepsilon \times \dots \times \varepsilon)$ (et \emptyset est rationnel). Toute relation singleton $\{(u_1, \dots, u_n)\}$ est PR car elle s'écrit $u_1 \times \dots \times u_n$ (et chaque $\{u_i\}$ est un langage rationnel). Toutes les relations finies sont PR car ce sont des unions de singletons. Plus généralement, toutes les relations reconnaissables sont PR : elles s'écrivent comme une union finie de produits de langages reconnaissables d'après le théorème de Mezei [Eil74].

La relation égalité $\{(u, u) \mid u \in \mathbf{A}^*\}$ est PR car elle s'écrit $\mathbf{A}^*.(\varepsilon \times \varepsilon)$. La relation différent $\{(u, v) \mid u \in \mathbf{A}^*, v \in \mathbf{A}^*, u \neq v\}$ est PR car elle s'écrit

$$\mathbf{A}^*.(\varepsilon \times \mathbf{A}^+) \cup \mathbf{A}^*.(\mathbf{A}^+ \times \varepsilon) \cup \bigcup_{\substack{a, b \in \mathbf{A} \\ a \neq b}} \mathbf{A}^*. (a.\mathbf{A}^* \times b.\mathbf{A}^*).$$

Pour R une relation sur \mathbf{A}^* , $Res(R)$ est le plus petit ensemble de relations contenant R et clos par :

Projection. Pour $R' \in Res(R)$ et $k \in [|R'|]$,

$$\{(u_1, \dots, u_{k-1}, u_{k+1}, \dots, u_{|R'|}) \mid (u_1, \dots, u_{|R'|}) \in R'\} \in Res(R).$$

Résidu à gauche. Pour $a \in \mathbf{A}$ et $R' \in Res(R)$,

$$\{(u_1, \dots, u_{|R'|}) \mid a.(u_1, \dots, u_{|R'|}) \in R'\} \in Res(R) .$$

Lemme 4.30 *Pour toute relation PR R sur \mathbf{A}^* , $Res(R)$ est fini et ne contient que des relations PR.*

Preuve. Standard. Par récurrence sur la définition de R , en utilisant le fait que les résidus d'un langage rationnel sont finis, et que le résidu d'un produit est le produit des résidus. \square

La remarque suivante, montre qu'une forme de réciproque est aussi vraie.

Remarque 4.31 *Une relation R est PR si et seulement si $Res(R)$ est fini.*

Lemme 4.32 *Pour toute relation R sur \mathbf{A}^* , il existe un ensemble fini $dec(R)$ de $(N+1)$ -uplets $(\sigma, R_1, \dots, R_N)$ où σ est une permutation de $[|R|]$, les $R_i \in Res(R)$ sont des relations et*

$$R = \bigcup_{(\sigma, R_1, \dots, R_N, k) \in dec(R)} \sigma(a_1.R_1 \times \dots \times a_N.R_N \times \varepsilon^k) \quad \text{où } k = |R| - \sum_{i \in [N]} |R_i| .$$

Preuve. Standard. Toujours par récurrence sur la structure de R . \square

Traduction de PR vers VR'

Considérons un ensemble fini T de relations PR tel que $Res(T) = T$. On définit la signature $\Sigma[T]$ par $\Sigma[T] = \{ \langle R \rangle \mid R \in T \}$ avec $|\langle R \rangle|$ l'arité de R . Soit \mathcal{S}_T la $\Sigma[T]$ -structure dont l'univers est \mathbf{A}^* et telle que $\langle R \rangle^{\mathcal{S}_T} = R$ pour tout $R \in T$.

Lemme 4.33 *Il existe un transducteur déterministe \mathbf{t}_T des $\text{Auto}_d[\mathbf{A}]$ -termes² dans les VR'-termes tel que pour tout $\text{Auto}_d[\mathbf{A}]$ -terme t ,*

$$\llbracket \mathbf{t}_T(t) \rrbracket \approx \mathcal{S}_T|_{\llbracket t \rrbracket} .$$

Construction. On utilise le transducteur \mathbf{t}_T possédant un unique état q (initial) et dont les transitions sont les suivantes:

$$\begin{aligned} q(a_1x_1 + \dots + a_Nx_n) &= \mathcal{I}(\mathcal{I}_{a_1}(q(x_1)) \oplus \dots \oplus \mathcal{I}_{a_N}(q(x_N))) \\ q(a_1x_1 + \dots + a_Nx_n + \varepsilon) &= \mathcal{I}(\mathcal{I}_{a_1}(q(x_1)) \oplus \dots \oplus \mathcal{I}_{a_N}(q(x_N)) \oplus \text{un}[\circ]) \end{aligned}$$

2. Tous les termes considérés dans ces résultats de conversions ne sont pas nécessairement finis. Nous ne précisons plus ce point dans la suite du mémoire.

$$\text{avec } \mathcal{I}_a = \left| \begin{array}{l} \langle R \rangle_a (x_1, \dots, x_{|R|}) : \langle R \rangle (x_1, \dots, x_{|R|}), \quad R \in T \\ \langle R \rangle (x_1, \dots, x_{|R|}) : \bigwedge_{i \in [N]} \langle R_i \rangle_{a_i} (y_1^i, \dots, y_{|R_i|}^i) \wedge \bigwedge_{j \in [k]} \circ(z_j) \\ \text{pour } (\sigma, R_1, \dots, R_N, k) \in \text{dec}(R) \\ \text{et } (y_1^1, \dots, y_{|R_1|}^1, \dots, y_1^N, \dots, y_{|R_N|}^N, z_1, \dots, z_k) \\ = (x_{\sigma(1)}, \dots, x_{\sigma(|R|)}) \end{array} \right|$$

□

Le théorème suivant énonce que toute structure dont l'univers est accepté par un automate déterministe et dont les relations sont PR (restreintes à l'univers) est isomorphe à la solution d'un VR'-système d'équations, et cette traduction s'effectue au moyen d'un transducteur déterministe. Ainsi, ce résultat établit un lien entre une représentation interne (un univers décrit par un automate et des relations décrites explicitement) et une représentation équationnelle. Nous verrons dans la suite que la réciproque est aussi vraie. Notons que la version originale de ce résultat ne fait pas intervenir la notion de transducteur déterministe.

Théorème 20 (Carayol, Colcombet [CC03]) *Soit \mathbf{A} un alphabet, $\dot{R}_1, \dots, \dot{R}_n$ des symboles relationnels et R_1, \dots, R_n des relations PR sur \mathbf{A}^* telles que l'arité de \dot{R}_i soit $|R_i|$. Il existe un transducteur déterministe \mathbf{t} des $\text{Auto}_d[\mathbf{A}]$ -termes dans les VR'-termes tel que*

$$\mathcal{S} \approx \llbracket \mathbf{t}(t) \rrbracket \quad \text{avec } \mathcal{S} \text{ tel que } \begin{cases} \mathcal{U}_{\mathcal{S}} &= \llbracket t \rrbracket, \\ \dot{R}_i^{\mathcal{S}} &= R_i \cap \llbracket t \rrbracket^{|R_i|}. \end{cases}$$

Preuve. Soit $T = \text{Res}(\{R_1, \dots, R_n\})$. L'ensemble T est fini d'après le lemme 4.30. En appliquant alors le lemme 4.33, on obtient un transducteur \mathbf{t}_T satisfaisant les conclusions du lemme. On a alors $\mathcal{S} \approx \mathcal{I}(\mathbf{t}_T(t))$ avec

$$\mathcal{I} = \left| \begin{array}{l} \dot{R}_i(x_1, \dots, x_{|R_i|}) : \langle R_i \rangle (x_1, \dots, x_{|R_i|}) \end{array} \right|.$$

□

4.4.3 Représentation par transformations

4.4.3.1 Premier contact

Une dernière façon de représenter les graphes préfixes reconnaissables est de partir de l'arbre binaire et d'effectuer une succession de transformations sur ces

graphes. Le premier de ces résultats est dû à Caucal et a permis la preuve de la décidabilité de la théorie monadique de ces graphes.

Théorème 21 (Caucal [Cau96]) *Les graphes préfixes reconnaissables sont les graphes obtenus par marquage rationnel, substitution rationnelle inverse et élimination de sommets marqués de l'arbre binaire.*

Cette approche ne convient qu'aux graphes puisque la substitution rationnelle inverse ne peut mettre en relation que deux sommets. Barthelmann présente une version logique de ce résultat, en termes d'interprétations.

Théorème 22 (Barthelmann [Bar97]) *Les graphes préfixes reconnaissables sont les transductions monadiques de l'arbre binaire complet.*

Ce résultat se prolonge naturellement en une définition de structures «préfixes reconnaissables» particulièrement simple : ce sont les structures obtenues par *transductions monadiques de l'arbre binaire*.

Nous verrons que cette définition est bien compatible avec les autres définitions : les solutions des VR'-systèmes finis d'équations, et les structures sur les mots dont l'interprétation de chaque symbole est une relation PR.

Théorème 23 (Barthelmann [Bar97]) *Pour toute signature Σ il existe une interprétation monadique \mathcal{I} telle que pour tout VR' $[\Sigma]$ -terme t ,*

$$\mathcal{I}(t) \approx \llbracket t \rrbracket .$$

Nous présentons ci-dessous une preuve similaire pour une famille plus générale de structures. La remarque 5.29 page 146 revient sur ce point.

4.4.3.2 Des transductions monadiques vers les relations PR

Il nous reste donc à établir que toute structure obtenue comme interprétation monadique du dépliage d'un graphe peut être obtenue comme valeur d'un VRS'-système d'équations (théorème 24). De plus, passer du graphe au système d'équations s'effectue par application d'une interprétation monadique.

Théorème 24 (Carayol, Colcombet [CC03]) *Pour tout alphabet \mathbf{A} , tout alphabet relationnel Σ et toute interprétation monadique \mathcal{I} des \mathbf{A} -arbres déterministes dans les Σ -structures, il existe des relations PR $(\dot{R})_{R \in \Sigma}$ sur $\{0, 1\}^*$ telles que $|\dot{R}| = |R|$ et une transduction monadique T des graphes déterministes vers les Auto_d -systèmes d'équations tels que pour tout graphe enraciné déterministe \mathcal{G} , on a*

$$\mathcal{I}(\text{Depliage}(\mathcal{G})) \approx \mathcal{S} \quad \text{avec } \mathcal{S} \text{ tel que } \begin{cases} \mathcal{U}_{\mathcal{S}} & = \text{Solution}(T(\mathcal{G})) , \\ \mathcal{R}^{\mathcal{S}} & = \dot{R} \cap \text{Solution}(T(\mathcal{G}))^{|R|} . \end{cases}$$

Nous ne prouvons pas cette implication qui est similaire à la preuve du lemme 5.30 (cette dernière est en fait plus complexe).

4.4.4 Application à la hiérarchie de Caucal

Le dépliage des graphes finis donne les arbres réguliers de degré fini. L'application des substitutions rationnelles inverses sur les arbres réguliers donne les graphes préfixe reconnaissables. Que se passe-t-il quand on itère ces deux opérations? Cela définit les graphes de la hiérarchie de Caucal.

Définition 4.34 *Les graphes du niveau zéro sont les graphes finis. Pour un entier strictement positif n , les graphes du niveau n de la hiérarchie s'obtiennent par application d'un dépliage suivi d'une substitution rationnelle inverse et d'une élimination de sommets marqués³ à un graphe du niveau $n - 1$ de la hiérarchie.*

Il s'agit de la première définition de cette famille [Cau98]. Seule une propriété de ces graphes était connue à ce moment : ils possèdent une théorie monadique décidable. Cette propriété est évidente puisqu'elle résulte de la préservation bien connue de la décidabilité de la théorie monadique par toutes les transformations mises en jeu.

L'intérêt pour cette famille n'est revenu que plus tard à la lumière de récents travaux sur les termes infinis. Knapik, Niwinsky et Urzyczyn montrent que les termes infinis solutions de schémas de programme récursifs d'ordre supérieur (sous une certaine hypothèse de «sûreté») possèdent une théorie monadique décidable [KNU02]. (Courcelle avait déjà montré une version à l'ordre deux de ce résultat [Cou95]). A peu près simultanément, Courcelle et Knapik montrent un résultat similaire sur une famille de termes obtenus par applications successives d'une certaine forme de «substitutions» [CK02]. Caucal montre alors que ces deux familles de termes sont égales et qu'elles coïncident avec les termes obtenus par dépliage des graphes de la hiérarchie.

D'autres travaux sur le sujet montrent la robustesse de cette famille. L'aspect représentation est étudié par Carayol et l'auteur dans [CC03]. Des investigations de structures et de clôture sont présentées dans [CW03]. Dans une thématique plus éloignée, des travaux sont menés sous l'angle des jeux de parité⁴ [Cac03].

4.4.5 Liens entre VR et HR

Remarquons tout d'abord que les familles de graphes obtenues comme solutions de VR-systèmes d'équations sont nécessairement distinctes des familles de graphes obtenues comme solution de HR-systèmes d'équations. Il s'agit d'une conséquence de la propriété 4.35.

3. L'élimination des sommets marqués n'existe pas dans le travail original de Caucal. Elle arrive naturellement lorsque l'on considère ces graphes sous l'angle de la logique.

4. Les jeux de parité donnent une description alternative sous forme de jeu du μ -calcul modal. Décider si un joueur possède une stratégie gagnante d'un de ces jeux est équivalent à décider de la satisfaction d'une formule du μ -calcul modal. Cette équivalence est particulièrement riche d'enseignements.

Propriété 4.35 *Il existe un graphe solution d'un VR-système fini d'équations qui n'est pas de largeur arborescente bornée.*

Preuve. Il suffit de considérer le graphe préfixe reconnaissable de la figure 4.7 (page 117). Modulo nommage, orientation des arcs et absence des boucles, il s'agit de la clique infinie qui n'est pas de largeur arborescente bornée. \square

Barthelmann a montré que, si l'on contraint un graphe solution d'un VR-système d'équations fini à ne pas contenir de graphe biparti complet aussi grand que l'on veut (une contrainte moins forte que la finitude de la largeur arborescente), alors ce graphe est HR-équationnel.

Théorème 25 (Barthelmann [Bar98]) *Les graphes VR-équationnels dont les sous-graphes bipartis complets sont bornés sont HR-équationnels.*

Nous redémontrons ce résultat dans le présent paragraphe. Le principe des preuves est similaire. L'apport consiste à utiliser les transducteurs déterministes comme unique outil de traduction. La preuve reste ainsi valable pour tout système d'équations.

4.4.5.1 Intuitions sur la preuve

Supposons que l'on applique une opération $\text{arc}[c, a, c']$ à un graphe contenant un unique sommet de couleur c et un «grand nombre de sommets» de couleur c' . Il est impossible d'obtenir le même résultat au moyen d'opérateurs HR. En effet, ajouter un arc à un sommet de couleur c' à l'aide d'opérateurs HR nécessite de fusionner les sommets de couleur c' et ceci n'a absolument pas l'effet souhaité (il n'y aurait plus qu'un seul sommet de couleur c' au lieu d'un «grand nombre»).

La seule solution est donc d'anticiper l'opérateur $\text{arc}[c, a, c']$ et de produire un graphe dans lequel ces arcs sont déjà présents. Ceci justifie la présence, dans chaque état du transducteur que l'on construit, d'un ensemble (appelé D dans la suite) collectant l'ensemble des arcs qu'il faut produire «en plus». Dans le cas présent, cet ensemble est augmenté de (c, a, c') lorsque le transducteur consomme l'opérateur $\text{arc}[c, a, c']$. Il s'agit de la technique fondamentale de la construction.

Tout ceci fonctionne bien jusqu'à la rencontre d'un opérateur d'union disjointe. Il est alors nécessaire de produire de nouveaux sommets utilisant des couleurs fraîches afin de pouvoir propager les arcs à produire vers les deux arguments de l'union. L'ensemble des nouvelles couleurs (noté E dans la suite) est ainsi susceptible d'augmenter. Il est néanmoins possible d'en borner le nombre en utilisant l'hypothèse sur les graphes bipartis complets. Ceci permet de garantir que le transducteur possède un nombre fini d'états.

Ces remarques permettent de donner une première ébauche d'hypothèse de récurrence. Étant donné un graphe coloré par C (décrit par un VR-terme), un

nouvel ensemble de couleur E et un ensemble d'arcs $D \subseteq (E \cup C) \times A \times (E \cup C)$, il s'agit de produire un graphe \mathcal{H} (décrit par un HR-terme) coloré par $E \cup C$ tel que :

$$\text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}) \approx \mathcal{H} .$$

Le détail de la preuve est technique. La première partie de la démonstration (4.4.5.2) consiste à normaliser les VR-termes afin que chaque couleur utile soit utilisée par un unique sommet ou par un «grand nombre» de sommets. Certaines propriétés générale sont ensuite mentionnées (4.4.5.3). On présente alors les hypothèses de récurrence (4.4.5.4). Le paragraphe 4.4.5.5 conclut.

4.4.5.2 Normalisation

L'opération fusion ne peut opérer que sur une couleurs à la fois, et tous les sommets de cette couleur se retrouvent identifiés. Ce n'est pas le cas de l'opération arc. Pour cette raison, les couleurs n'affectant qu'un unique sommet dans un graphe jouent un rôle particulier dans la preuve du théorème 26. L'objectif de ce paragraphe, formalisé par le lemme 4.37, est de normaliser un système d'équations vis à vis de cette problématique.

Le premier outil dont nous avons besoin est la possibilité de compter le nombre de sommets d'une certaine couleur dans un graphe. Soient donc \mathcal{G} un graphe coloré et c une couleur, on pose $|\mathcal{G}|_c = |\{v \in V_{\mathcal{G}} \mid \mathcal{G} \models c(v)\}|$.

Lemme 4.36 *Soient k un entier positif et c une couleur, l'ensemble $\{\mathcal{G} \mid |\mathcal{G}|_c = k\}$ est VR- μ -définissable.*

Preuve. On a les égalités suivantes :

$$\begin{aligned} |\text{sommet}[c_0]|_c &= \begin{cases} 0 & \text{si } c \neq c_0 \\ 1 & \text{si } c = c_0 \end{cases}, & |\mathcal{G} \oplus \mathcal{G}'|_c &= |\mathcal{G}|_c + |\mathcal{G}'|_c, \\ |\text{recol}[\alpha](\mathcal{G})|_c &= \sum_{\alpha(c')=c} |\mathcal{G}|_{c'}, & |\text{arc}[c_0, a, c_1](\mathcal{G})|_c &= |\mathcal{G}|_c, \\ |\perp|_c &= 0 \end{aligned}$$

Ces équivalences s'interprètent comme un transducteur déterministe sans anticipation des VR-termes vers les N -termes (voir exemple ?? page ??). Le singleton $\{k\}$ étant N - μ -définissable et l'image inverse d'un ensemble μ -définissable par un transducteur déterministe étant μ -définissable (lemme 3.48), on obtient que $\{\mathcal{G} \mid |\mathcal{G}|_c = k\}$ est VR- μ -définissable. \square

Supposons k fixé. On peut dès lors, en faisant usage d'un transducteur déterministe, assurer que toutes les couleurs apparaissant dans un calcul affectent 0,

1 ou plus de k sommets. Ainsi, on dit qu'un graphe \mathcal{G} satisfait la propriété (*) si pour toute couleur c , $|\mathcal{G}|_c \notin \{2, \dots, k\}$. Le lemme 4.37 nous permet de supposer cette propriété lorsque l'on utilise les opérateurs VR.

Lemme 4.37 *Soient C un ensemble de couleurs et A un ensemble d'étiquettes. Il existe un nouvel ensemble de couleurs C' , un transducteur déterministe \mathbf{t} des VR[A, C]-termes dans les VR[A, C']-termes et une application β de C' dans C tels que*

1. *pour tout VR[A, C]-terme t , $\text{recol}[\beta](\llbracket \mathbf{t}(t) \rrbracket) = \llbracket t \rrbracket$,*
2. *pour tout sous-terme t' de $\mathbf{t}(t)$, $\llbracket t' \rrbracket$ satisfait (*).*

Le preuve se trouve à l'annexe A.3, page A.3.

4.4.5.3 Premières propriétés

On suppose dorénavant que k est pair, et on s'intéresse à la propriété pour un graphe de ne pas contenir un sous-graphe biparti complet de taille $k/2$. Le choix de $k/2$ est explicité par le lemme ci-dessous.

Lemme 4.38 *Soient \mathcal{G} un graphe, c, c' deux couleurs et a une étiquettes. Si $|\mathcal{G}|_c > k$ et $|\mathcal{G}|_{c'} \geq k$, alors $\text{arc}[c, a, c'](\mathcal{G})$ contient un graphe biparti complet de taille $k/2$.*

Dans la suite, l'ensemble de couleurs C est fixé. Soit \mathcal{G} un graphe coloré par C satisfaisant (*). On pose $C_0(\mathcal{G}) = \{c \in C \mid |\mathcal{G}|_c = 0\}$, $C_1(\mathcal{G}) = \{c \in C \mid |\mathcal{G}|_c = 1\}$ et $C_+(\mathcal{G}) = \{c \in C \mid |\mathcal{G}|_c \geq k\}$. La propriété (*) équivaut alors à $C = C_0(\mathcal{G}) \cup C_1(\mathcal{G}) \cup C_+(\mathcal{G})$.

Soient $E \subseteq C$ et $E' \subseteq C$ deux ensembles de couleurs, alors $D(E, E')$ représente l'ensemble $(E \times A \times E') \cup (E' \times A \times E)$, c'est-à-dire le plus grand ensemble D tel que $\text{arc}[D]$ n'ajoute des arcs qu'entre des sommets de couleurs E et des sommets de couleur E' . Pour $D \subseteq D(C, C)$, alors $C(D)$ représente l'ensemble $\{c \mid (c, a, c') \in D \text{ ou } (c', a, c) \in D\}$, c'est-à-dire l'ensemble des couleurs apparaissant dans D .

Nous mentionnons ici certaines propriétés des opérateurs VR+fusion qui seront utilisées implicitement par la suite.

$$\begin{aligned} \text{fusion}[E] \circ \text{fusion}[E'] &= \text{fusion}[E \cup E'] = \text{fusion}[E'] \circ \text{fusion}[E] \\ \text{arc}[D] \circ \text{arc}[D'] &= \text{arc}[D \cup D'] = \text{arc}[D'] \circ \text{arc}[D] \\ \text{fusion}[E] \circ \text{arc}[D] &= \text{arc}[D] \circ \text{fusion}[E] \end{aligned}$$

$$\begin{aligned} \text{fusion}[E](\mathcal{G}) &= \mathcal{G} \quad \text{si } E \cap C(\mathcal{G}) \subseteq C_1(\mathcal{G}) \\ \text{fusion}[E](\mathcal{G}) &\approx \text{fusion}[E](\mathcal{G} \oplus \text{sommet}[E]) \quad \text{si } E \subseteq C(\mathcal{G}) \end{aligned}$$

Enfin, deux dernières égalités nous sont utiles.

Lemme 4.39 Soient deux graphes colorés \mathcal{G} et \mathcal{H} , $E \subseteq C$ et $D \subseteq D(C, C)$.

$$\text{Si } C(\mathcal{H}) \cap C(D) \subseteq C(\mathcal{G}) \cap E \quad \text{alors} \\ \text{fusion}[E](\text{arc}[D](\mathcal{G} \oplus \mathcal{H})) = \text{fusion}[E](\text{arc}[D](\mathcal{G}) \oplus \mathcal{H}) .$$

La seconde égalité s'obtient par double application de la précédente.

Lemme 4.40 Soient deux graphes colorés \mathcal{G} et \mathcal{H} , $E \subseteq C(\mathcal{G}) \cap C(\mathcal{H})$ et $D \subseteq D(E, C)$. Alors,

$$\text{fusion}[E](\text{arc}[D](\mathcal{G} \oplus \mathcal{H})) = \text{fusion}[E](\text{arc}[D](\mathcal{G}) \oplus \text{arc}[D](\mathcal{H})) .$$

4.4.5.4 Hypothèses de récurrence

On construit maintenant les relations continues \mathcal{R} et \mathcal{R}' qui correspondent aux invariants associés aux états du transducteur déterministe que l'on cherche à obtenir.

La relation \mathcal{R} a pour paramètres \mathcal{G}^∞ , E et D satisfaisant (A),(B) et (C) définis comme suit.

- (A) E est un ensemble de couleurs fraîches (c. a. d. $C \cap E = \emptyset$),
- (B) $D \subseteq D(C \cup E, C \cup E)$,
- (C) $\text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}^\infty)$ ne contient pas de graphe biparti complet de taille $k/2$.

La relation $\mathcal{R}[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$, où \mathcal{G} est un graphe coloré par C et \mathcal{H} un graphe coloré par $C \cup E$ est alors satisfaite si

- $\mathcal{G} \subseteq \mathcal{G}^\infty$,
- $C_1(\mathcal{G}) = C_1(\mathcal{G}^\infty)$,
- et $\text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}) \approx \mathcal{H}$.

Pour tous \mathcal{G}^∞ , E et D , $\mathcal{R}[\mathcal{G}^\infty, E, D]$ relie continuellement les graphes colorés par C aux graphes colorés par $C \cup E$.

La relation $\mathcal{R}'[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$ est un raffinement de \mathcal{R} . En effet, $\mathcal{R}'[\mathcal{G}^\infty, E, D]$ n'est définie que si

- (D) $D \subseteq D(C_1(\mathcal{G}^\infty) \cup E, C_+(\mathcal{G}^\infty))$,
- (E) et $E \subseteq C(D)$,

et dans ce cas $\mathcal{R}'[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$ ssi $\mathcal{R}[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$.

La relation \mathcal{R}' permet, par rapport à \mathcal{R} , de borner le nombre de couleurs nécessaire dans E (et par conséquent le nombre d'états du transducteur déterministe).

Lemme 4.41 Il existe un entier K (dépendant de k , $|C|$ et $|A|$) tel que pour tout graphe coloré \mathcal{G}^∞ , E et D satisfaisant (A-E), on a $|E| \leq K$.

Preuve. Soit $C'_1 = C_1(\mathcal{G}^\infty) \cup E$, $C_+ = C_+(\mathcal{G}^\infty)$ et soit $K = 2k|C||A|$. D'après (D-E), $|D| \geq |E| \geq K$. Supposons sans perdre de généralité (cf hypothèse D) que $|D \cap (C'_1 \times A \times C_+)| \geq K/2$. Alors, il existe a et $c \in C_+(\mathcal{G}^\infty)$ tel que $|D'| \geq k$, avec $D' = D \cap (C'_1 \times \{a\} \times \{c\})$. Par suite $\text{arc}[D'](\text{sommet}[E] \oplus \mathcal{G}^\infty)$ contient un sous-graphe biparti complet de taille $k/2$, et il en est donc de même pour $\text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}^\infty)$. Cela contredit (C). \square

Le détail des lemmes de récurrence ainsi que les preuves principales sont en annexes (cf. page 197). Il s'agit des lemmes A.10, A.11, A.12, A.13 et A.14.

4.4.5.5 Le résultat

Théorème 26 *Pour tout ensemble de couleurs C , tout ensemble d'étiquettes A et tout entier positif k , il existe un transducteur déterministe \mathbf{t} des $\text{VR}[A, C]$ -termes vers les HR -termes tel que pour tout $\text{VR}[A, C]$ -terme t , si $\llbracket t \rrbracket$ ne contient pas de sous-graphe biparti complet de taille k , alors $\mathbf{t}(t)$ est défini et*

$$\llbracket \mathbf{t}(t) \rrbracket \approx \llbracket t \rrbracket .$$

Preuve. Il suffit d'utiliser les lemmes A.10, A.11, A.12, A.13 et A.14 comme autant de règles de validité d'un transducteur déterministe. Il ne manque que l'hypothèse 3 du lemme de validité 3.59. Pour cela, on utilise une relation supplémentaire $\mathcal{R}''[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$ qui, sous les hypothèses (A-C), est définie par

$$\text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}) \approx \text{fusion}[E](\text{sommet}[E] \oplus \mathcal{H}) .$$

Bien entendu, on a $\mathcal{R}''[\mathcal{G}^\infty, E, D](\perp, \perp)$. De plus si $\mathcal{R}[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$ est vrai, alors $\mathcal{R}''[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$ l'est aussi. Réciproquement si $\mathcal{R}''[\mathcal{G}^\infty, E, D](\mathcal{G}, \mathcal{H})$ alors $\mathcal{R}[\mathcal{G}^\infty, E, D](\mathcal{G}, \text{fusion}[E](\text{sommet}[E] \oplus \mathcal{H}))$. Ces deux équivalences s'interprètent comme une transition supplémentaire permettant de remplir les conditions de satisfaction de l'hypothèse 3 du lemme de validité 3.59. \square

Corollaire 4.42 (Barthelmann) *Les familles de graphes suivantes sont équivalentes :*

- les graphes VR -équationnels de largeur arborescente bornée,
- les graphes VR -équationnels uniformément creux,
- les graphes VR -équationnels dont les sous-graphes bipartis complets sont bornés,
- les graphes HR -équationnels de largeur arborescente bornée.

Chapitre 5

Structures automatiques sur les termes

Dans le chapitre 4, les structures considérées étaient construites sur un univers de mots et par réécriture suffixe (ou préfixe). Les structures étudiées ici sont définies sur des ensembles de termes et les relations sont décrites au moyen d'automates. Les propriétés de cette famille sont fondamentalement différentes de celles de la famille précédente.

Nous commençons par un rappel sur les langages de termes et présentons les opérateurs correspondants (paragraphe 5.1). Le paragraphe 5.2 s'intéresse à la famille des structures à présentation terme-automatique proprement dite.

5.1 Langages de termes

Si les langages sur les mots sont bien connus, c'est moins le cas des langages de termes qui en sont pourtant une extension naturelle. Nous présentons dans ce paragraphe quelques résultats classiques de cette théorie. Une étude détaillée de ce domaine se trouve en ligne [CDG⁺97].

Nous abordons les langages de termes sous deux angles, celui des automates (paragraphe 5.1.1), puis celui des systèmes d'équations (paragraphe 5.1.2).

5.1.1 Automates d'arbres

Nous présentons tout d'abord les automates d'arbres et les langages de termes qu'ils reconnaissent (paragraphe 5.1.1.1), puis le cas particulier des automates déterministes descendants 5.1.1.2.

5.1.1.1 Le cas général

Un automate d'arbres sur l'alphabet gradué \mathcal{F} est un triplet (Q, δ, F) où

- Q est un ensemble fini d'états,
- δ est un ensemble de transitions de la forme $(f(q_1, \dots, q_{|f|}), q)$, avec $f \in \mathcal{F}$ et $q_1, \dots, q_{|f|}, q \in Q$,
- $F \subseteq Q$ est l'ensemble des états d'acceptation.

On suppose dorénavant que Q et \mathcal{F} sont disjoints, et on attribue à chaque élément de Q l'arité 0 afin de pouvoir l'utiliser comme un symbole. On étend alors δ en une unique plus petite relation $\dot{\delta}$ de $\mathcal{T}(\mathcal{F} \uplus Q) \times Q$ satisfaisant :

- pour tout $q \in Q$, $(q, q) \in \dot{\delta}$,
- si il existe $(f(q_1, \dots, q_{|f|}), q) \in \delta$ telle que pour tout i , $(t_i, q_i) \in \dot{\delta}$ alors $(f(t_1, \dots, t_{|f|}), q) \in \dot{\delta}$.

Dans la pratique, on identifie δ et $\dot{\delta}$, et on note $\delta^{-1}(q)$ l'ensemble des termes t tels que $(t, q) \in \delta$. On dit qu'un automate d'arbres (Q, δ, F) *accepte* un terme t si $t \in \delta^{-1}(F)$. Un langage de termes L est *rationnel* (ou reconnaissable) si il existe un automate d'arbres (Q, δ, F) tel que $L = \delta^{-1}(F) \cap \mathcal{T}(\mathcal{F})$.

Exemple 5.1 *Considérons l'alphabet gradué $\mathcal{F} = \{\text{vrai} : 0, \text{faux} : 0, \neg : 1, \wedge : 2, \vee : 2\}$. Chaque terme sur cet alphabet correspond à une formule booléenne. Il s'agit de déterminer si une formule s'évalue à **vrai**. Pour cela, on prend $Q = \{q_0, q_1\}$, $F = \{q_1\}$ et les transitions suivantes :*

$$\delta = \left\{ \begin{array}{l} (\text{vrai}, q_1), (\text{faux}, q_0), \\ (\neg(q_0), q_1), (\neg(q_1), q_0), \\ (\vee(q_0, q_0), q_0), (\vee(q_1, q_0), q_1), \\ (\vee(q_0, q_1), q_1), (\vee(q_1, q_1), q_1), \\ (\wedge(q_0, q_0), q_0), (\wedge(q_1, q_0), q_0), \\ (\wedge(q_0, q_1), q_0), (\wedge(q_1, q_1), q_1) \end{array} \right\} .$$

*On vérifie alors qu'un terme est accepté par un automate si et seulement si la formule booléenne correspondante s'évalue à **vrai**.*

Un automate (Q, δ, F) est *déterministe ascendant* si pour tout symbole f et pour tous états $q_1, \dots, q_{|f|}$, il existe au plus un état q tel que $(f(q_1, \dots, q_{|f|}), q) \in \delta$. Il est *complet* si il existe toujours un tel état. Pour cette forme de déterminisme, le résultat classique de déterminisation des automates sur les mots s'étend sans difficulté aux termes comme le montre le résultat suivant.

Propriété 5.2 *Un langage est rationnel si et seulement si il est l'ensemble des termes acceptés par un automate d'arbres déterministe ascendant (cf. [CDG⁺97]).*

5.1.1.2 Automates descendants déterministes

Une autre forme de déterminisme est néanmoins possible pour les automates d'arbres. Il s'agit du déterminisme descendant. Un automate (Q, δ, F) est *déterministe descendant* si F est un singleton et pour tout symbole f et tout état q , il existe au plus une transition de la forme $(f(q_1, \dots, q_{|f|}), q)$. En ce sens, il n'est plus possible de déterminer tous les langages rationnels comme l'illustre l'exemple 5.3.

Exemple 5.3 *Considérons sur l'alphabet $\{f : 2, a : 0, b : 0\}$ le langage $L = \{f(a, b), f(b, a)\}$. Supposons qu'un automate déterministe descendant (Q, δ, F) accepte ce langage. Supposons $F = \{q_0\}$ et $(f(q, q'), q_0)$ comme seule transition associée au symbole f et à l'état q_0 . Comme $f(a, b) \in L$, on a $(a, q) \in \delta$. Comme $f(b, a) \in L$, on a $(a, q') \in \delta$. Il s'ensuit que $f(a, a) \in L$. Contradiction.*

En fait, on peut caractériser précisément l'ensemble des termes reconnus par un automate déterministe descendant par leurs branches.

Propriété 5.4 (folklore) *Un langage de termes L est accepté par un automate déterministe descendant si et seulement si il existe un langage rationnel de mots B tel que :*

$$L = \{t \mid \mathcal{LB}(t) \subseteq B\} .$$

5.1.2 Représentation équationnelle

Tout comme pour les mots, les automates d'arbres admettent une représentation équationnelle. Nous considérons deux telles représentations, l'une pour les automates quelconques et l'autre pour les automates déterministes descendant.

5.1.2.1 Langages rationnels

Les langages rationnels de termes sont les solutions des systèmes d'équations finis sur les opérateurs $\text{Auto}^{\mathcal{T}}[\mathcal{F}]$ (ou plus simplement $\text{Auto}^{\mathcal{T}}$) suivants:

- L'union de langage (notée $+$).
- L'application d'un symbole $f \in \mathcal{F}$, qui à $|f|$ langages $L_1, \dots, L_{|f|}$ associe le langage $f(L_1, \dots, L_{|f|})$.

Propriété 5.5 *Un langage de termes est rationnel si et seulement si il est solution d'un $\text{Auto}^{\mathcal{T}}$ -système fini d'équations.*

5.1.2.2 Automates descendants déterministes

Les automates descendants déterministes admettent aussi une représentation équationnelle. Pour \mathcal{F} un alphabet gradué, les opérateurs $\text{Auto}_{\text{dd}}^{\mathcal{T}}[\mathcal{F}]$ (ou plus

simplement $\text{Auto}_{\text{dd}}^{\mathcal{T}}$) sont d'une unique forme :

- pour $f_1, \dots, f_n \in \mathcal{F}$ des symboles distincts, on considère l'opérateur qui aux langages $L_1^1, \dots, L_{|f_1|}^1, \dots, L_1^n, \dots, L_{|f_n|}^n$ associe

$$\sum_{i \in [n]} f_i(L_1^i, \dots, L_{|f_i|}^i) .$$

Bien entendu, un langage est accepté par un automate déterministe descendant si et seulement si il est solution d'un $\text{Auto}_{\text{dd}}^{\mathcal{T}}$ -système fini d'équations. En fait, la propriété 5.4 se généralise à tout système d'équations.

Propriété 5.6 *Soit \mathcal{F} un alphabet gradué.*

Il existe un transducteur déterministe \mathbf{t} des $\text{Auto}_{\text{dd}}^{\mathcal{T}}$ -termes dans les Auto_{d} -termes tel que pour tout $\text{Auto}_{\text{dd}}^{\mathcal{T}}$ -terme t ,

$$\llbracket t \rrbracket = \{u \in \mathcal{T}(\mathcal{F}) \mid \mathcal{LB}(u) \subseteq \llbracket \mathbf{t}(t) \rrbracket\} .$$

Réciproquement, il existe un transducteur déterministe \mathbf{t}' des Auto_{d} -termes dans les $\text{Auto}_{\text{dd}}^{\mathcal{T}}$ -termes tel que pour tout Auto_{d} -terme t ,

$$\llbracket \mathbf{t}'(t) \rrbracket = \{u \in \mathcal{T}(\mathcal{F}) \mid \mathcal{LB}(u) \subseteq \llbracket t \rrbracket\} .$$

La preuve de cette propriété est sans difficulté particulière.

5.2 Structures terme-automatiques

Les structures terme-automatiques sont des structures qui, à isomorphisme près, ont un ensemble rationnel de termes comme univers, et dont les interprétations des symboles sont décrites «au moyen d'automates».

Là encore, nous donnons une triple représentation de ces structures.

La représentation interne s'obtient au moyen d'automates de termes. Elle est décrite au paragraphe 5.2.1.

La représentation équationnelle au moyen des opérateurs VRS' qui sont une extension des opérateurs VR' par la possibilité d'utiliser un produit synchrone et de restreindre l'univers. Il s'agit de l'objet du paragraphe 5.2.2.

La représentation par transformations utilise une transformation particulière, l'interprétation monadique des parties finies, appliquée à l'arbre binaire complet. Une interprétation des parties finies consiste, au moyen de formules monadiques, à définir une nouvelle structure dont l'univers est un ensemble de parties finies de la structure originale. Cette représentation est étudiée au paragraphe 5.2.3.

Nous montrons que ces trois représentations sont équivalentes.

Pour finir, nous énonçons au paragraphe 5.2.4 comment ces résultats peuvent être adaptés à la famille des structures automatiques, un cas particulier des structures terme-automatiques.

5.2.1 Présentation terme-automatique

Nous commençons par présenter la famille des relations «terme-synchronisées» sur les termes (paragraphe 5.2.1.1), puis comment l'appliquer pour décrire des structures : les structures à présentation terme-automatique (paragraphe 5.2.1.2).

5.2.1.1 Relations terme-synchronisées

Dorénavant Ω est un nouveau symbole d'arité 0. Soit \mathcal{F} un alphabet gradué. Considérons les alphabets gradués \mathcal{F}_Ω^n et \mathcal{F}_Ω^* définis par

$$\begin{aligned} \mathcal{F}_\Omega^n &= \{[f_1, \dots, f_n] \mid f_1, \dots, f_n \in \mathcal{F} \uplus \{\Omega\}\}, \\ \text{avec } |[f_1, \dots, f_n]| &= \max_{i \in [n]} |f_i|, \\ \text{et } \mathcal{F}_\Omega^* &= \bigcup_{n \in \mathbb{N}} \mathcal{F}_\Omega^n. \end{aligned}$$

La projection sur la i ème composante pour $i \in [n]$ de $[f_1, \dots, f_n] \in \mathcal{F}_\Omega^*$, notée $\pi_i([f_1, \dots, f_n])$ est f_i . On étend π_i aux \mathcal{F}_Ω^n -termes comme suit. Pour $i \in [n]$ et t un \mathcal{F}_Ω^n -terme, $\pi_i(t)$ est le $\mathcal{F} \cup \{\Omega\}$ -terme défini par l'équation suivante.

$$\pi_i([f_1, \dots, f_n](t_1, \dots, t_k)) = f_i(\pi_i(t_1), \dots, \pi_i(t_{|f_i|})) \quad \text{avec } k = \max_{j \in [n]} |f_j|.$$

Le n -uplet de termes $\pi(t)$ est alors égal à $(\pi_1(t), \dots, \pi_n(t))$.

Par exemple, considérons l'alphabet gradué $\mathcal{F} = \{f : 2, a : 0, b : 0\}$ et le \mathcal{F}_Ω^2 -terme $t = [f, f]([a, b], [f, a]([a, \Omega], [b, \Omega]))$, alors $\pi(t) = (f(a, f(a, b)), f(b, a))$.

Un \mathcal{F}_Ω^n -terme t est *bien formé* si pour tout sous-terme de t de la forme $[f_1, \dots, f_n](t_1, \dots, t_k)$, on a :

$$\text{pour tout } i \in [k], j \in [n], \quad \pi_i(t_j) = \Omega \quad \text{ssi } j > |f_i|.$$

Soit N_Ω^n le langage des \mathcal{F}_Ω^n -termes bien formés tel que le symbole à la racine ne contient pas Ω . Ce langage est rationnel (en fait, il est même reconnu par un automate d'arbres déterministe descendant).

Lemme 5.7 (Dauchet et Tison [DT90]) *Pour tout n , l'application π est une bijection de N_Ω^n sur $\mathcal{T}(\mathcal{F})^n$.*

Une relation $R \subseteq \mathcal{T}(\mathcal{F})^n$ est *terme-synchronisée* (**RR** dans les travaux de Dauchet et Tison) si elle est l'image par π d'un langage rationnel $L \subseteq N_\Omega^n$. Les propriétés de clôture des langages rationnels s'étendent alors aux relations terme-synchronisées directement grâce au lemme 5.7.

Lemme 5.8 ([DT90]) *Les relations terme-synchronisées sont closes par union, intersection et complémentation.*

Comme pour le lemme 5.8, les deux propriétés ci-dessous sont classiques et simples à établir.

Lemme 5.9 ([DT90]) *Les relations terme-synchronisées sont closes par projection et cylindrification.*

5.2.1.2 Structures à présentation terme-automatique

Nous utilisons ici la terminologie introduite par Blumensath [Blu99]. Une *présentation terme-automatique*¹ est un quadruplet $P = (\mathcal{F}, U, \Sigma, I)$ où \mathcal{F} est un alphabet gradué, $U \subseteq \mathcal{T}(F)$ est un langage rationnel, Σ est un alphabet relationnel fini et I est une application qui à chaque $R \in \Sigma$ associe une partie rationnelle de $\mathcal{T}(\mathcal{F}_\Omega^{|R|})$. La Σ -structure $[P]$ associée à P a pour univers U , pour signature Σ et $\pi(I(R))$ comme interprétation d'un symbole $R \in \Sigma$. Une structure \mathcal{S} admet une présentation terme-automatique si il existe une présentation terme-automatique P telle que \mathcal{S} est isomorphe à $[P]$.

Les lemmes 5.8 et 5.9 permettent alors de simplement établir le théorème fondamental 27.

Théorème 27 (Dauchet et Tison [DT90]) *La théorie au premier ordre des structures admettant une présentation terme-automatique est décidable.*

La représentation donnée précédemment est pratique pour effectuer des codages. Il est néanmoins possible de la simplifier en utilisant un alphabet restreint à un symbole binaire et une constante. Il s'agit du lemme 5.10.

Lemme 5.10 *Toute structure admettant une présentation terme-automatique, admet une présentation terme-automatique sur l'alphabet gradué $\{p : 2, c : 0\}$.*

Preuve. Considérons une présentation $(\mathcal{F}, U, \Sigma, I)$. Appelons \mathcal{F}_0 l'alphabet gradué $\{p : 2, c : 0\}$. Il s'agit de se donner un codage «convenable» sous forme d'un morphisme injectif de $\mathcal{T}(\mathcal{F})$ dans $\mathcal{T}(\mathcal{F}_0)$. Pour cela, soient f_1, \dots, f_K les symboles de \mathcal{F} . On suppose que l'on dispose de $K + 1$ \mathcal{F}_0 -termes finis distincts $\dot{f}_1, \dots, \dot{f}_K$ et $\dot{\Omega}$. Soit N la plus grande des arité des symboles de \mathcal{F} . On définit alors le morphisme ρ des \mathcal{F} -termes dans les \mathcal{F}_0 termes par l'équation suivante:

$$\rho(f(t_1, \dots, t_{|f|})) = p^{N+1}(f, \rho(t_1), \dots, \rho(t_{|f|}), \overbrace{\dot{\Omega}, \dots, \dot{\Omega}}^{N-|f| \text{ fois}}),$$

avec pour tout $n \in \mathbb{N}$, $p^n(t_1, \dots, t_n) = p(t_1, p(t_2, \dots, p(t_n, \dot{\Omega}) \dots))$.

1. Pour être précis, il s'agit d'une présentation injective. Une présentation en générale dispose d'une relation terme-synchronisée supplémentaire permettant de «quotienter» la structure. Quand les classes d'équivalences de cette relation sont réduites à un unique élément, on parle de présentation injective. Blumensath montre que toute présentation peut être rendue injective.

L'application ρ ainsi définie est un morphisme, donc $U' = \rho(U)$ est une partie rationnelle de $\mathcal{T}(\mathcal{F}_0)$. Il est évident de plus que ρ est injective. Il s'agit donc d'une bijection de U sur U' .

Il reste à montrer que l'image d'une relation terme-synchronisée par ρ est elle-même terme-synchronisée. Nous ne donnons pas ici le détail de cette preuve. Notons juste que la construction est telle que la position des arguments d'un symbole après codage par ρ ne dépend que de leur numéro et non du symbole lui-même. C'est cette propriété essentielle qui fait fonctionner la construction. \square

5.2.2 Représentation équationnelle

Nous donnons ici une définition des structures à présentation terme-automatique au moyen de systèmes finis d'équations. Les opérateurs correspondants sont présentés au paragraphe 5.2.2.1. Suit un résultat de normalisation (5.2.2.2). L'équivalence avec la présentation terme-automatique est établie au paragraphe 5.2.2.3.

5.2.2.1 Définition

Les opérateurs VRS' sont une extension des opérateurs VR' par (principalement) l'ajout d'un opérateur de produit synchrone. Ainsi, VRS' signifie VR' +produit Synchrone. Pour Σ un alphabet relationnel, les opérateurs $VRS'[\Sigma]$ (ou plus simplement VRS') sont les suivants:

- La structure à un élément $un[R]$ où $R \in \Sigma$ est un symbole d'arité 1. On se contente aussi parfois de un dont l'interprétation de tous les symboles est vide.
- L'union disjointe.
- L'interprétation booléenne positive.
- Le produit synchrone.
- La restriction à l'interprétation d'un symbole relationnel unaire R , notée $restrict[R]$, a pour sémantique

$$restrict[R](\mathcal{S}) = \mathcal{S}|_{R^s} .$$

Ainsi, les opérateurs VRS' ne sont qu'une extension des opérateurs VR' par la possibilité d'utiliser un produit synchrone et la restriction. Cette dernière opération, si elle n'augmente pas l'expressivité des opérateurs VR' , est cruciale dans le cas présent. Concernant l'opérateur de produit et d'après la propriété 2.22, il est possible d'utiliser indifféremment le produit synchrone ou le produit généralisé.

Le paragraphe 5.2.2.2 montre comment normaliser ces systèmes d'équations.

5.2.2.2 Normalisation

Le lemme 4.17 (page 108) énonce que les opérateurs \mathbf{VR}' admettent une version normalisée dans laquelle les interprétations booléennes positives sont en forme linéaire complète. Nous montrons dans ce paragraphe que ce résultat reste valable dans le cas des opérateurs \mathbf{VRS}' . En fait, nous montrons ici un résultat plus fort qui, de plus, précise qu'une seule restriction est suffisante (lemme 5.13). La première étape consiste à éliminer les opérations de restriction (lemme 5.11). Le reste de la preuve n'est alors plus qu'une simple extension du lemme 4.17.

Retard de la restriction

Il s'agit ici d'éliminer les opérateurs de restriction d'un \mathbf{VRS}' -système d'équations. Il n'est pas possible de les éliminer totalement. En revanche, il est possible de les retarder afin qu'un seul de ces opérateurs soit nécessaire.

La construction se fait au moyen d'un transducteur déterministe. La solution naturelle consistant à utiliser un nouveau symbole relationnel \bullet d'arité 1 «marquant» les éléments de l'univers qui n'ont pas encore été éliminés par une restriction fonctionne parfaitement.

Ainsi, étant donnée une Σ -structure \mathcal{S} (obtenue en utilisant éventuellement des opérateurs de restriction), on cherche à produire une $\Sigma \uplus \{\bullet\}$ -structure \mathcal{S}' telle que :

$$\mathcal{I}_\bullet(\mathcal{S}) = \text{restrict}[\bullet](\mathcal{S}') ,$$

$$\text{avec } \mathcal{I}_\bullet = \left| \begin{array}{ll} R(x_1, \dots, x_{|R|}) & : R(x_1, \dots, x_{|R|}), \quad R \in \Sigma \\ \bullet(x) & : \text{vrai} \end{array} \right| .$$

Supposons $\mathcal{I}_\bullet(\mathcal{S}_1) = \text{restrict}[\bullet](\mathcal{S}'_1)$ et $\mathcal{I}_\bullet(\mathcal{S}_2) = \text{restrict}[\bullet](\mathcal{S}'_2)$. Alors

$$\begin{aligned} \mathcal{I}_\bullet(\mathcal{S}_1 \oplus \mathcal{S}_2) &= \mathcal{I}_\bullet(\mathcal{S}_1) \oplus \mathcal{I}_\bullet(\mathcal{S}_2) \\ &= \text{restrict}[\bullet](\mathcal{S}'_1) \oplus \text{restrict}[\bullet](\mathcal{S}'_2) \\ &= \text{restrict}[\bullet](\mathcal{S}'_1 \oplus \mathcal{S}'_2) \end{aligned}$$

$$\text{et de même, } \quad \mathcal{I}_\bullet(\mathcal{S}_1 \otimes \mathcal{S}_2) = \text{restrict}[\bullet](\mathcal{S}'_1 \otimes \mathcal{S}'_2) .$$

Supposons maintenant $\mathcal{I}_\bullet(\mathcal{S}) = \text{restrict}[\bullet](\mathcal{S}')$, alors,

$$\begin{aligned} \mathcal{I}_\bullet(\text{restrict}[R](\mathcal{S})) &= \text{restrict}[R](\mathcal{I}_\bullet(\mathcal{S})) \\ &= \text{restrict}[R](\text{restrict}[\bullet](\mathcal{S}')) \\ &= \text{restrict}[\bullet](\mathcal{I}_{\text{restrict}[R]}(\mathcal{S}')) \end{aligned}$$

$$\text{avec } \mathcal{I}_{\text{restrict}[R]} = \left| \begin{array}{ll} R'(x_1, \dots, x_{|R'|}) & : R'(x_1, \dots, x_{|R'|}) \quad R' \in \Sigma \\ \bullet(x) & : \bullet(x) \wedge R(x) \end{array} \right| .$$

On a aussi pour $\mathcal{I} = \left| \begin{array}{l} R(x_1, \dots, x_{|R|}) \\ \phi_R \end{array} \right|$,

$$\begin{aligned} \mathcal{I}_\bullet(\mathcal{I}(\mathcal{S})) &= \mathcal{I}_\mathcal{I}(\mathcal{I}_\bullet(\mathcal{S})) \quad \text{où} \quad \mathcal{I}_\mathcal{I}(\mathcal{S}) = \left| \begin{array}{l} R(x_1, \dots, x_{|R|}) \\ \bullet(x) \end{array} \right| : \begin{array}{l} \phi_R \\ \bullet(x) \end{array} \\ &= \mathcal{I}_\mathcal{I}(\text{restrict}[\bullet](\mathcal{S}')) \\ &= \text{restrict}[\bullet](\mathcal{I}_\mathcal{I}(\mathcal{S}')) \end{aligned}$$

et pour finir,

$$\mathcal{I}_\bullet(\text{un}) = \text{restrict}[\bullet](\mathcal{I}_\bullet(\text{un})) .$$

Enfin, remarquons que $\mathcal{I}_\bullet(\perp) = \text{restrict}[\bullet](\perp)$.

Appliquons maintenant ces égalités et le lemme 3.59 pour éliminer l'opérateur de restriction. On obtient le lemme ci-dessous.

Lemme 5.11 *Pour toute signature Σ , il existe un transducteur déterministe \mathbf{t} des $\text{VRS}'[\Sigma]$ -termes dans les VRS' -termes sans opérateurs de restriction tel que pour tout $\text{VRS}'[\Sigma]$ -terme t ,*

$$\begin{aligned} \llbracket t \rrbracket &= \mathcal{I}_e(\text{restrict}[\bullet](\llbracket \mathbf{t}(t) \rrbracket)) , \\ \text{avec } \mathcal{I}_e &= \left| \begin{array}{l} R(x_1, \dots, x_{|R|}) \\ R(x_1, \dots, x_{|R|}) \end{array} \right| : \begin{array}{l} R \in \Sigma, R \neq \bullet \end{array} \end{array} \end{aligned}$$

Normalisation

Maintenant qu'il n'y a plus d'opérateur de restriction, on effectue la normalisation en réutilisant la preuve du lemme 4.17 et en y ajoutant un lemme traitant du cas du produit synchronisé. On reprend donc les notations du paragraphe 4.4.1.2, en particulier la définition de \mathcal{I}_0 (page 107).

Le lemme traitant du produit synchronisé est alors des plus simple.

Lemme 5.12 *Pour toutes Σ -structures \mathcal{S}_1 et \mathcal{S}_2 , on a l'égalité suivante:*

$$\mathcal{I}_0(\mathcal{S}_1 \otimes \mathcal{S}_2) = \mathcal{I}_0(\mathcal{S}_1) \otimes \mathcal{I}_0(\mathcal{S}_2) .$$

Il est dès lors possible de conclure.

Lemme 5.13 *Pour toute signature Σ , il existe un transducteur déterministe \mathbf{t}_0 des $\text{VRS}'[\Sigma]$ -termes dans les VRS' -termes à interprétations linéaires complètes sans restriction et une interprétation \mathcal{I}_e linéaire complète tels que pour tout $\text{VRS}'[\Sigma]$ -terme t , on a :*

$$\mathcal{I}_e(\text{restrict}[\bullet](\llbracket \mathbf{t}_0(t) \rrbracket)) = \mathcal{I}_0(\llbracket t \rrbracket) .$$

Preuve. Comme pour le lemme 4.17, on utilise le lemme 5.12 pour traiter l'opérateur \otimes . \square

En combinant les lemmes 5.11 et 5.13 on obtient le résultat de normalisation complet énoncé ci-dessous.

Propriété 5.14 *Pour tout alphabet relationnel fini Σ , il existe un transducteur déterministe \mathbf{t}' des $\text{VRS}'[\Sigma]$ -termes vers les VRS' -termes sans restriction et à interprétation linéaire complète, et une interprétation linéaire complète \mathcal{I}'_e tels que pour tout $\text{VRS}'[\Sigma]$ -terme t ,*

$$\mathcal{I}'_e(\text{restrict}[\bullet][\mathbf{t}'(t)]) = \llbracket t \rrbracket .$$

5.2.2.3 Équivalence avec la représentation automatique

Il s'agit d'établir le résultat suivant.

Théorème 28 *Une structure admet une présentation terme-automatique si et seulement si elle est isomorphe à la solution d'un VRS' -système fini d'équations.* Ce résultat s'obtient en combinant les lemmes 5.16 et 5.17. Ces deux lemmes sont successivement présentés.

De la présentation terme-automatique au système d'équations

On suppose pour commencer que l'alphabet est égal à $\{p : 2, c : 0\}$; cela est possible d'après le lemme 5.10.

Informellement, une structure d'univers $\mathcal{T}(\{p : 2, c : 0\})$ a un univers isomorphe à la solution de l'équation $X = (X \otimes X) \oplus \text{un}$. En effet, le produit code la présence des deux arguments de l'opérateur p , et l'union permet d'ajouter un nouvel élément représentant le terme c . La structure du système d'équations que nous produisons est la même; il ne manque que les interprétations permettant de «calculer» les relations de la structure proprement dite.

On considère donc une structure terme-automatique représentée par $(\mathcal{F}, U, \Sigma, I)$, avec $\mathcal{F} = \{p : 2, c : 0\}$. La première étape de la démonstration (lemme 5.15) consiste à montrer le résultat sans restriction à un univers rationnel, c'est à dire pour $U = \mathcal{T}(\mathcal{F})$. L'entier N représente dorénavant la plus grande des arités des symboles dans Σ . L'ensemble de variables $\{x_1, \dots, x_N\}$ est noté \mathcal{X} . pour $E \subseteq [n]$, \mathcal{X}_E représente l'ensemble de variables $\{x_i \mid i \in E\}$, et $\overline{\mathcal{X}_E}$ représente le vecteur de variables $(x_{\sigma(1)}, \dots, x_{\sigma(|E|)})$ où σ est l'unique bijection croissante de $\llbracket E \rrbracket$ sur E .

Remarque de notation : les symboles de \mathcal{F}_Ω^* peuvent être d'arité 2 ou 0 suivant qu'ils contiennent le symbole p ou non. Ainsi, les transitions de δ sont de la forme $([f_1, \dots, f_n](q_1, \dots, q_n), q)$, où $n = |[f_1, \dots, f_n]|$, et $n = 0$ ou $n = 2$. Dans ce qui suit, toutes les notations sont énoncées pour $n = 2$. La version $n = 0$ se déduit directement et ne provoque pas de conflit.

Chaque langage $I(R)$ est accepté par un automate de termes \mathcal{A}_R . On considère l'automate \mathcal{A} formé de l'union (disjointe) de tous les automates \mathcal{A}_R pour $R \in \Sigma$. L'ensemble des états de cet automate est Q et l'ensemble des transitions est δ . Pour tout $R \in \Sigma$, $q_R \in Q$ est l'état initial de l'automate \mathcal{A}_R dans \mathcal{A} . Pour tout $q \in Q$, L_q représente le langage de $\mathcal{F}_\Omega^{|q|}$ accepté par \mathcal{A} avec pour état initial q . De plus, $F(q)$ représente l'ensemble $\{i \in [|q|] \mid \exists t \in L_q, \pi_i(t) \neq \Omega\}$. Soit $\gamma = ([f_1, \dots, f_n](q_1, q_2), q)$ une transition, $F_p(\gamma)$ représente l'ensemble $\{i \in [n] \mid f_i = p\}$ et de même, $F_c(\gamma) = \{i \in [n] \mid f_i = c\}$

Si L est un langage de \mathcal{F}_Ω^n et $E \subseteq [n]$, $L|_E$ représente le langage L «projeté» sur les composantes d'indice $i \in E$. Formellement et récursivement,

pour $t = ([f_1, \dots, f_n](t_1, \dots, t_k)) \in \mathcal{T}(\mathcal{F}_\Omega^n)$, $t|_E = [f_{\sigma(1)}, \dots, f_{\sigma(n)}](t_{1|E}, \dots, t_{k|E})$,

où σ est la bijection croissante de $[|E|]$ sur E ,

et pour $L \subseteq \mathcal{T}(\mathcal{F}_\Omega^n)$, $L|_E = \{t|_E \mid t \in L\}$.

Bien entendu, si un terme $t \in \mathcal{F}_\Omega^n$ est bien formé, alors il en va de même de $t|_E$ pour tout $E \subseteq [n]$.

L'alphabet relationnel qui nous permet de construire la structure est le suivant :

$$\begin{aligned} \Sigma' = & \{ \langle q, E \rangle \mid q \in Q, E \subseteq F(q), E \neq \emptyset \} \\ & \cup \{ \langle \gamma, E \rangle_c \mid \gamma \in \delta, E \subseteq F_c(\gamma), E \neq \emptyset \} \\ & \cup \{ \langle \gamma, E \rangle_p \mid \gamma \in \delta, E \subseteq F_p(\gamma), E \neq \emptyset \} , \\ \text{avec pour tous } & q \in Q, E \subseteq F(q), E \neq \emptyset, \quad | \langle q, E \rangle | = |E| , \\ & \text{pour tous } \gamma \in \delta, E \subseteq F_c(\gamma), E \neq \emptyset, \quad | \langle \gamma, E \rangle_c | = |E| , \\ & \text{et pour tous } \gamma \in \delta, E \subseteq F_p(\gamma), E \neq \emptyset, \quad | \langle \gamma, E \rangle_p | = |E| . \end{aligned}$$

On s'intéresse alors à la structure \mathcal{S} dont l'univers est $\mathcal{T}(\mathcal{F})$ et pour tout $q \in Q$, et tout $E \subseteq F(q)$ tel que $E \neq \emptyset$, on a $\langle q, E \rangle^{\mathcal{S}} = \pi(L_q|_E)$.

On pose alors :

$$\mathcal{I} = \left| \begin{array}{l} \langle q, E \rangle (\overline{\mathcal{X}_E}) : \left\{ \begin{array}{ll} \langle \gamma, E_c \rangle_c (\overline{\mathcal{X}_{E_c}}) & \text{si } E_p = \emptyset \\ \langle \gamma, E_p \rangle_p (\overline{\mathcal{X}_{E_p}}) & \text{si } E_c = \emptyset \\ \langle \gamma, E_p \rangle_p (\overline{\mathcal{X}_{E_p}}) & \\ \wedge \langle \gamma, E_c \rangle_c (\overline{\mathcal{X}_{E_c}}) & \text{sinon} \end{array} \right. \\ \text{pour } \begin{array}{l} \gamma \in \delta , \\ E_c = E \cap F_c(\gamma) , \\ E_p = E \cap F_p(\gamma) \end{array} \end{array} \right|$$

et pour tout $i \in [2]$,

$$\mathcal{I}_i = \left| \begin{array}{l} \langle \gamma, E \rangle_p (\overline{\mathcal{X}_E}) : \langle q_i, E \rangle (\overline{\mathcal{X}_E}) \\ \text{avec } \gamma = ([f_1, \dots, f_{|q|}](q_1, q_2), q) \in \delta \end{array} \right|$$

La structure \mathcal{S} satisfait l'équation suivante (sur les opérateurs VRS').

$$\begin{aligned} \mathcal{S} &\approx \mathcal{I}((\mathcal{I}_1(\mathcal{S}) \otimes \mathcal{I}_2(\mathcal{S})) \oplus \mathcal{I}_c(\text{un})) , \\ &\text{avec } \mathcal{I}_c = \left| \langle \gamma, E \rangle (\overline{\mathcal{X}_E}) : \text{vrai} \quad \gamma \in \delta, E \neq \emptyset, E \subseteq F_c(\gamma) \right| . \end{aligned}$$

Il s'agit en fait, à isomorphisme près, de la plus petite solution de cette équation. La démonstration ne pose pas de problème car les relations $I(R)$ sont elles-mêmes définies par un plus petit point fixe. Notons tout de même que les symboles relationnels d'arité 0 ont été évités dans cette construction. Il aurait fallu sinon les traiter séparément.

Lemme 5.15 *Pour toute structure \mathcal{S} admettant une présentation terme-automatique $(\{c : 0, p : 2\}, \mathcal{T}(\{c : 0, p : 2\}), \Sigma, I)$, il existe un VRS'-système fini d'équations S tel que*

$$\mathcal{S} \approx \text{Solution}(S) .$$

Preuve. Une fois la structure \mathcal{S} obtenue par un VRS'-système d'équations, il est aisé de réobtenir les relations $I(R)$. Il suffit d'appliquer à \mathcal{S} l'interprétation suivante :

$$\left| R(x_1, \dots, x_{|R|}) : \langle q_R, \{x_1, \dots, x_{|R|}\} \rangle (x_1, \dots, x_{|R|}) \text{ pour } R \in \Sigma \right| .$$

□

Lemme 5.16 *Pour toute structure \mathcal{S} admettant une présentation terme-automatique, il existe un VRS'-système fini d'équations S tel que :*

$$\mathcal{S} \approx \text{Solution}(S) .$$

Preuve. On se ramène au cas de l'alphabet $\mathcal{F}_0 = \{p : 2, c : 0\}$ par le lemme 5.10. Soit $P = (\mathcal{F}_0, U, \Sigma, I)$ la présentation correspondante. L'univers U étant rationnel, il s'agit d'un cas particulier de relation synchronisée d'arité 1. On ajoute à la signature un nouveau symbole \bullet d'arité 1 dont l'interprétation est U . Soit $\mathcal{F}_\bullet = \mathcal{F}_0 \cup \{\bullet\}$. Après application du lemme 5.15, on obtient un VRS'-système fini d'équations S dont la solution est isomorphe à $[(\mathcal{F}_\bullet, \mathcal{T}(\mathcal{F}_\bullet), \Sigma, I')]$ où I' coïncide avec I sur Σ et $I'(\bullet) = U$. On a alors :

$$\begin{aligned} [P] &\approx \mathcal{I}_e(\text{restrict}[\bullet]) , \\ &\text{avec } \mathcal{I}_e = \left| R(x_1, \dots, x_{|R|}) : R(x_1, \dots, x_{|R|}) \quad R \neq \bullet \right| . \end{aligned}$$

□

On peut remarquer que la structure définie ainsi est déjà normalisée au sens du lemme 5.13.

Du système d'équations à la présentation terme-automatique

Lemme 5.17 *Pour tout VRS'-système fini d'équations S , $Solution(S)$ admet une présentation terme-automatique.*

Nous ne donnons pas la preuve ici de ce résultat; elle est très similaire à la preuve du lemme 5.28 (remplacer pour cela le μ -calcul par un $Auto^T$ -système d'équations).

5.2.3 Représentation logique

Il s'agit dans ce paragraphe de montrer l'équivalence entre les solutions de VRS'-systèmes d'équations et les structures obtenues par une certaine transformation appliquée au dépliage d'un graphe (théorème 29). La transformation en question est une «interprétation des parties finies» présentée au paragraphe 5.2.3.1. Le résultat d'équivalence est énoncé par le théorème 29. Les paragraphes 5.2.3.2 et 5.2.3.3 établissent chacun l'une des directions de l'équivalence.

5.2.3.1 Interprétation des parties

Une façon de construire une structure possédant une théorie au premier ordre décidable consiste à partir d'une structure \mathcal{S}_0 ayant une théorie au second ordre monadique décidable, puis de définir l'univers comme un ensemble MSO-définissable de parties de l'univers de \mathcal{S}_0 et d'utiliser des formules monadiques pour décrire l'interprétation des relations.

Formellement, une *interprétation monadique des parties* est un quadruplet $\mathcal{I} = (\Sigma, \Sigma', \delta, (\phi_R))$ où Σ et Σ' sont des alphabets relationnels, $\delta(X)$ est une formule monadique sur Σ possédant une variable libre monadique X , et pour tout $R \in \Sigma'$, $\phi_R(X_1, \dots, X_{|R|})$ est une formule monadique sur Σ possédant $|R|$ variables libres monadiques $X_1, \dots, X_{|R|}$.

L'application de $\mathcal{I} = (\Sigma, \Sigma', \delta, (\phi_R))$ à une Σ -structure \mathcal{S} est la Σ' -structure $\mathcal{I}(\mathcal{S})$ définie comme suit.

$$\mathcal{U}_{\mathcal{I}(\mathcal{S})} = \{U \subseteq \mathcal{U}_{\mathcal{S}} \mid \mathcal{S} \models \delta(U)\} ,$$

et pour $R \in \Sigma'$, $R^{\mathcal{I}(\mathcal{S})} = \{(U_1, \dots, U_{|R|}) \subseteq (\mathcal{U}_{\mathcal{I}(\mathcal{S})})^{|R|} \mid \mathcal{S} \models \phi_R(U_1, \dots, U_{|R|})\} .$

Cette construction peut produire des structures ayant un univers non dénombrable. Ainsi, appliquée à l'arbre binaire complet, on obtient les structures ω -terme-automatiques selon la terminologie de Blumensath [Blu99].

On restreint donc cette définition aux structures dont l'univers est formé d'ensembles finis. Ainsi une *interprétation monadique des parties finies* est une interprétation monadique des parties $\mathcal{I} = (\Sigma, \Sigma', \delta, (\phi_R))$ telle que pour toute Σ -structure \mathcal{S} et tout $U \subseteq \mathcal{U}_{\mathcal{S}}$, si $\mathcal{S} \models \delta(U)$ alors U est fini.

L'intérêt de l'introduction de ces définitions est énoncé par le lemme suivant et son corollaire.

Lemme 5.18 *Pour \mathcal{I} une interprétation monadique des parties et ϕ une formule du premier ordre, il existe une formule monadique Φ telle que pour toute structure \mathcal{S} , $\mathcal{I}(\mathcal{S}) \models \phi$ si et seulement si $\mathcal{S} \models \Phi$.*

Preuve. Il suffit de transformer les variables de ϕ en variables du second ordre monadique et de substituer aux relations leur définition dans \mathcal{I} . \square

Corollaire 5.19 *Si \mathcal{S} est une structure ayant une théorie monadique décidable et \mathcal{I} est une interprétation des parties, alors $\mathcal{I}(\mathcal{S})$ a une théorie au premier ordre décidable.*

Il est maintenant possible d'énoncer le résultat qui nous intéresse.

Théorème 29 *Les familles de graphes suivantes sont isomorphes :*

- les solutions des VRS'-systèmes d'équations,
- les images par interprétation monadique des parties finies du dépliage d'un graphe,

et ces conversions sont effectives.

Les détails concernant l'effectivité des conversions se trouvent dans les deux lemmes 5.28 et 5.30 dont le théorème 29 est la combinaison.

Avant de s'intéresser à ces démonstrations, précisons quelques unes des conséquences.

Corollaire 5.20 *Pour ϕ une formule du premier ordre et Σ une signature, l'ensemble des modèles de ϕ est VRS' $[\Sigma]$ -reconnaisable.*

Corollaire 5.21 *Les solutions des VRS'-systèmes d'équations de théorie monadique décidable ont une théorie au premier ordre décidable.*

Remarque 5.22 *(extension) Blumensath [Blu99] considère une logique plus expressive que la simple logique du premier ordre. Il s'agit d'étendre cette logique par des quantifications de la forme $\exists^\omega x.\phi$ signifiant «il existe une infinité de valuations possibles pour x telles que ϕ ». L'opération duale $\forall x.\phi = \neg\exists^\omega x.\neg\phi$ signifiant « ϕ est satisfaite pour toute valuation de x sauf pour un nombre fini» est aussi*

autorisée. Blumensath montre que cette logique est décidable sur les structures à présentation terme-automatique. C'est en fait toujours le cas pour les solutions des VRS'-systèmes d'équations de théorie monadique décidable.

Il suffit pour cela de montrer qu'un quantificateur $\exists^{\omega, F} X. \phi$ signifiant «il existe une infinité de valuations de X finies vérifiant ϕ » peut être ajouté à la logique monadique pourvu que l'on dispose d'un prédicat $F(Y)$ exprimant la finitude de la valuation d'une variable monadique Y .

En effet, pour toute formule monadique ϕ , toute variable monadique X , toute structure \mathcal{S} et toute valuation γ de $FV(\phi) - \{X\}$, on a

$$\gamma, \mathcal{S} \models \exists^{\omega, F} X. \phi \quad \text{ssi} \quad \gamma, \mathcal{S} \models \forall Y. F(Y) \Rightarrow \exists X. (X \not\subseteq Y) \wedge \phi .$$

Il se trouve que, dans le cas présent, la structure \mathcal{S} est un arbre (le dépliage du système d'équations). Il est donc possible d'exprimer le prédicat $F(Y)$ par une formule monadique (voir remarque 2.12 page 21).

5.2.3.2 D'un système d'équations à la représentation logique

Il s'agit dans ce paragraphe d'établir le lemme 5.28 permettant de passer de la sémantique des opérateurs VRS' à une interprétation monadique des parties finies.

On considère une signature Σ et un VRS' $[\Sigma]$ -terme t . Supposons pour commencer que les interprétations du terme soient sous forme linéaire complète sans relation d'arité 0. Supposons de plus que t ne comporte pas d'opérateurs de restriction. Dans la suite de ce paragraphe, on note \mathcal{F} cet ensemble d'opérateurs.

Il s'agit de coder pour tout \mathcal{F} -terme chaque élément de $\llbracket t \rrbracket$ par une partie finie des sommets principaux de t . Pour cela, on définit pour tout terme t la fonction partielle ρ_t des parties finies des sommets principaux de t dans l'univers de $\llbracket t \rrbracket$. La fonction ρ_t est décrite par récurrence (cette récurrence a bien un sens car $\rho_t(U)$ n'est défini que pour U fini) comme suit:

si $t = \text{un}[R]$ alors $\rho(\{\text{racine}^t\}) = 0$,

si $t = \mathcal{I}(t')$ alors $\rho_t = \rho_{t'}$,

si $t = t_1 \oplus t_2$ alors $\rho_t(X)$ est défini si il existe $i \in [2]$ tel que $X \in \text{dom}(\rho_{t_i})$ et dans ce cas $\rho_t(X) = (i, \rho_{t_i}(X))$,

si $t = t_1 \otimes t_2$ alors $\rho_t(X)$ est défini si $X = X_1 \cup X_2$ avec pour tout $i \in [2]$, $X_i \in \text{dom}(\rho_{t_i})$, et dans ce cas, $\rho_t(X) = (\rho_{t_1}(X_1), \rho_{t_2}(X_2))$.

Pour comprendre cette définition, il convient de revenir à la définition interne de chaque opérateur. Ainsi, le sommet $\rho(\{\text{racine}^t\}) = 0$ du graphe $\text{un}[R]$ est-il son seul sommet. Les cas de l'union disjointe et du produit s'interprètent similairement.

Exemple 5.23 *Considérons le terme suivant (le paramètre des opérateurs un est*

omis).

$$t = (\underbrace{\text{un}}_{v_0} \oplus \underbrace{\text{un}}_{v_1}) \otimes (\underbrace{\text{un}}_{v_2} \oplus \underbrace{\text{un}}_{v_3}) .$$

Les sommets correspondants aux opérateurs `un` sont identifiés par v_0, v_1, v_2 et v_3 .

La valeur de ce terme produit une structure dont l'univers contient quatre éléments $((1, 0), (1, 0)), ((2, 0), (1, 0)), ((1, 0), (2, 0))$ et $((2, 0), (2, 0))$. L'application ρ_t est alors définie comme suit:

$$\begin{aligned} \rho_t(\{v_0, v_2\}) &= ((1, 0), (1, 0)) \\ \rho_t(\{v_1, v_2\}) &= ((2, 0), (1, 0)) \\ \rho_t(\{v_0, v_3\}) &= ((1, 0), (2, 0)) \\ \rho_t(\{v_1, v_3\}) &= ((2, 0), (2, 0)) \end{aligned}$$

et ρ_t n'est pas défini partout ailleurs.

Ainsi, ρ_t est une bijection de son domaine sur l'univers de $\llbracket t \rrbracket$. Le lemme suivant généralise cette remarque.

Lemme 5.24 Pour tout terme t , ρ_t est une bijection de $\text{dom}(\rho_t)$ sur $\mathcal{U}_{\llbracket t \rrbracket}$.

Preuve. Par récurrence sur la définition de ρ . □

Lemme 5.25 L'ensemble $\{(t, U) \mid U \in \text{dom}(\rho_t)\}$ est $\mathcal{F}_{[1]}$ -reconnaissable.

Construction. Considérons un ensemble U de sommets principaux de t . Alors U appartient au domaine de ρ_t si et seulement si $(t, U) \in \text{sommet}[\]$ avec `sommet` une formule du μ -calcul définie comme suit:

$$\begin{aligned} \text{vide} &= \nu y.(\neg 1) \wedge (\mathcal{I}(y) \cup (y \oplus y) \cup (y \otimes y) \cup \text{un}) \\ \text{sommet} &= \mu x.(1 \cap \text{un}) \cup ((\neg 1) \cap (\mathcal{I}(x) \cup (x \oplus \text{vide}) \cup (\text{vide} \oplus x) \cup (x \otimes x))) . \end{aligned}$$

La formule `vide` vérifie qu'aucune marque n'est présente dans un terme. La formule `sommet` décompose un terme en suivant les règles définissant ρ . Le plus petit point fixe provient du fait que seuls les ensembles finis sont considérés. □

Soit N la plus grande arité des symboles de Σ . Pour $R \in \Sigma$ et π une injection de $\llbracket R \rrbracket$ dans $[N]$, on définit l'ensemble de $\mathcal{F}_{[N]}$ -termes $L_{R,\pi}$ comme suit:

$$L_{R,\pi} = \{(t, U_1, \dots, U_N) \mid R^{\llbracket t \rrbracket}(\rho_t(U_{\pi(1)}), \dots, \rho_t(U_{\pi(\llbracket R \rrbracket)}))\} .$$

La preuve du lemme 5.28 revient alors principalement à montrer que $L_{R,\pi}$ pour tout R et toute injection π est $\mathcal{F}_{[N]}$ -reconnaissable. Pour cela, on construit un

système d'équations (au sens de l'extension vectorielle du μ -calcul) possédant les variables suivantes:

$$V = \{ \langle R, \pi \rangle \mid \pi \text{ injection de } \llbracket R \rrbracket \text{ dans } [N] \} .$$

Soient $R \in \Sigma$ et π une injection de $\llbracket R \rrbracket$ dans $[N]$. La μ -formule $\phi_{R,\pi}$ est définie comme suit:

$$\begin{aligned} \phi_{R,\pi} = & (\text{un}[R] \cap \bigcap_{i \in \text{dom}(\pi)} i \cap \bigcap_{i \in [N] - \text{dom}(\pi)} \neg i) \\ & \cup \bigcap_{i \in [N]} \neg i \\ & \cap \left(\langle R, \pi \rangle \oplus \mathbf{1} \right) \cup \left(\mathbf{1} \oplus \langle R, \pi \rangle \right) \\ & \cup \left(\langle R, \pi \rangle \otimes \langle R, \pi \rangle \right) \\ & \cup \bigcup_{\mathcal{I}} \mathcal{I} \left(\bigcap_{i \in [k]} \langle R_i, \pi \circ \alpha_i \rangle \right) \\ & \quad \text{où } R(x_1, \dots, x_{|R|}) : \bigwedge_{i \in [k]} R_i(x_{\alpha_i(1)}, \dots, x_{\alpha_i(|R_i|)}) \\ & \quad \text{est une règle de } \mathcal{I} \text{ (une conjonction linéaire complète).} \end{aligned}$$

Lemme 5.26 *Pour tout symbole $R \in \Sigma$ et toute injection π de $\llbracket R \rrbracket$ dans $[N]$, on a*

$$L_{R,\pi} = \phi_{R,\pi}[\gamma] \quad \text{avec } \gamma(\langle S, \alpha \rangle) = L_{S,\alpha} \text{ pour tout } \langle S, \alpha \rangle \in V .$$

Preuve. En traitant séparément le cas de chaque opérateur. □

Lemme 5.27 *Pour tout symbole $R \in \Sigma$ et toute injection π de $\llbracket R \rrbracket$ dans $[N]$, on a*

$$(\mu \langle R, \pi \rangle . \{ \langle S, \alpha \rangle = \phi_{S,\alpha} \})[] = L_{R,\pi} .$$

On en déduit le lemme suivant.

Lemme 5.28 *Pour toute signature finie Σ , il existe une interprétation monadique des parties finies \mathcal{I} telle que pour tout $\text{VRS}'[\Sigma]$ -terme t ,*

$$\mathcal{I}(t) \approx \llbracket t \rrbracket .$$

Preuve. On suppose que le terme est normalisé en appliquant la propriété 5.14. À chaque formule ψ_R du μ -calcul est associé une formule monadique $\Psi_R(X_1, \dots, X_N)$ par le lemme 3.38. On construit alors l'interprétation des parties

$$\mathcal{I} = (\Sigma[\text{VRS}'], \Sigma, \delta, (\Psi'_{R,\pi}))$$

avec

$$\begin{aligned}\delta(X) &= \exists X_2, \dots, X_N. \Psi_{\bullet}(X, X_2, \dots, X_N) , \\ \Psi'_R(X_1, \dots, X_{|R|}) &= \exists X_{|R|+1}, \dots, X_N. \Psi_{R, id_{[|R|]}}(X_1, \dots, X_N) .\end{aligned}$$

□

Remarque 5.29 (Cas particulier en l'absence de produit) *Supposons que le terme t soit un VR'-terme. Dans ce cas, la fonction ρ n'est définie que sur des ensembles singletons. Un tel ensemble se représente naturellement par une variable du premier ordre. On en déduit que pour toute signature Σ , il existe une interprétation monadique \mathcal{I} telle que pour tout VR' $[\Sigma]$ -terme t , on a $\mathcal{I}(t) \approx \llbracket t \rrbracket$. Cette remarque établit le théorème 23 de Barthelmann.*

5.2.3.3 De la représentation logique à un système d'équations

Le propos de ce paragraphe est d'établir le lemme ci-dessous.

Lemme 5.30 *Pour toute signature Σ , tout ensemble fini d'étiquettes A et toute interprétation des parties finies \mathcal{I}_F des arbres déterministes étiquetés par A vers les Σ -structures, il existe une transduction monadique T des graphes déterministes étiquetés par A vers les VRS'-systèmes d'équations telle que pour tout graphe déterministe \mathcal{G} étiqueté par A on a*

$$\mathcal{I}_F(\text{Depliage}(\mathcal{G})) \approx \text{Solution}(T(\mathcal{G})) .$$

Soit $\mathcal{I}_F = (\mathbf{A}, \Sigma, \delta, (\phi'_R)_{R \in \Sigma})$. Soit $\mathcal{F} = \{p : 2, c : 0\}$. Le lemme 3.39 permet de se ramener à la situation dans laquelle \mathcal{G} est un \mathcal{F} -système d'équations. Il n'est plus dans la suite question que du dépliage de ce système d'équations: *c. a. d.* d'un \mathcal{F} -terme.

Pour commencer, on suppose que $\delta = \text{vrai}$ et on ajoute un symbole relationnel δ d'arité 1 tel que $\phi_\delta = \delta$. Ainsi, les sommets à conserver sont traités comme l'interprétation d'un symbole relationnel.

On considère les formules du μ -calcul ϕ_R obtenues à partir des formules monadiques ϕ'_R au moyen du lemme 3.38. Le lemme 3.39 nous précise que l'interprétation des parties finies \mathcal{I}_F ne fait référence qu'aux sommets P_t^c . Grâce à cela, il est possible de supposer que les formules ϕ_R n'utilisent que l'alphabet $\mathcal{F}'_{[|R|]}$ défini comme suit:

$$\text{pour } n \in \mathbb{N}, \quad \mathcal{F}'_n = \{(p, \emptyset) : 2\} \cup \{(c, E) : 0 \mid E \subseteq [n]\} .$$

Soit N la plus grande arité d'un symbole apparaissant dans Σ . On pose pour tout entier $n \in [M]$, $M_n = Cl(\{\phi_R \mid R \in \Sigma, |R| = n\})$. Soit $\Sigma' = \{\langle \phi, n, E \rangle \mid n \in$

$[N]$, $\phi \in M_n$, $E \subseteq [n] \cup \{\#\}$ avec $|\langle \phi, n, E \rangle| = |E|$ et $|\#| = 1$. Pour tout \mathcal{F} -terme t , on définit la structure \mathcal{S}_t comme la Σ' -structure dont l'univers est l'ensemble des parties finies de P_t^c et dont l'interprétation d'un symbole relationnel $\langle \phi, n, E \rangle \in \Sigma'$ est défini comme suit:

$$\begin{aligned} \langle \phi, n, E \rangle^{\mathcal{S}_t} &= \{(U_{\sigma(1)}, \dots, U_{\sigma(|E|)}) \mid (t_0, U_1, \dots, U_n) \in \phi[], \forall i, U_i \neq \emptyset \Rightarrow i \in E\}, \\ \#^{\mathcal{S}_t} &= \{U \subseteq P_t^c \mid U \text{ fini}, U \neq \emptyset\}. \end{aligned}$$

Pour $t' \sqsubseteq t$, on pose de plus $\mathcal{S}_{t,t'} = \mathcal{S}_t|_{2^{P_{t'}$.

La structure $\mathcal{S}_{t,t'}$ peut contenir une infinité de sommets. Ils sont tous marqués par $\#$ sauf exactement un qui correspond à \emptyset . Si $t' = \perp$, alors t' ne contient aucun sommet c et donc $\mathcal{S}_{t,t'}$ contient \emptyset comme unique sommet.

Lemme 5.31 *Il existe un entier k , une partition de $\mathcal{T}^\omega(\mathcal{F})$ en k parties L_1, \dots, L_k et k applications f_1, \dots, f_k VRS'-définissable telles que pour tout $i \in [k]$, tout \mathcal{F} -terme $t \in L_i$ et tout $t' \sqsubseteq t$ on a*

$$\mathcal{S}_{t,t'} \approx f_i(\mathcal{S}_{t,t'}) \quad \text{et} \quad \mathcal{S}_{t,\perp} \approx f_i(\emptyset).$$

Preuve. L'égalité $\mathcal{S}_{t,\perp} = \emptyset$ n'est pas satisfaite pour deux raisons. D'une part, les relations d'arité 0 peuvent avoir des interprétations non vides. D'autre part, l'univers de $\mathcal{S}_{t,\perp}$ n'est pas vide car il contient l'élément \emptyset .

On construit alors la partition de $\mathcal{T}^\omega(\mathcal{F})$ induite par M_n . Ainsi, pour tout $\phi \in M_n$, notons $\phi|_0$ la formule obtenue en remplaçant partout dans ϕ , (f, \emptyset) par f , (c, \emptyset) par c et (c, E) pour $E \neq \emptyset$ par \emptyset . Cette formule est telle que $\phi|_0[] = \{t \in \mathcal{T}^\omega(\mathcal{F}) \mid (t, \emptyset, \dots, \emptyset) \in \phi[]\}$.

Soit pour $W \subseteq M$, $\psi_W = (\cap_{\phi \in W} \phi|_0) \cap (\cap_{\phi \in M-W} (\mathcal{T}^\omega(\mathcal{F}) - \phi|_0))$. Les ψ_W forment une partition de $\mathcal{T}^\omega(\mathcal{F})$. On a de plus pour toute formule $\phi \in M$, $\phi[] = \cup_{W \ni \phi} \psi_W[]$. Soit maintenant \mathcal{I}_W définie par

$$\mathcal{I}_W = \left| \begin{array}{ll} \langle \phi, n, \emptyset \rangle () & : \text{vrai} & \text{si } \phi \in W \\ \langle \phi, n, E \rangle (x_1, \dots, x_{|E|}) & : \langle \phi, n, E \rangle (x_1, \dots, x_{|E|}) \\ \#(x) & : \#(x) \end{array} \right|$$

Pour tout \mathcal{F} -terme t et tout terme $t' \sqsubseteq t$, si $t \in \psi_W[]$ alors

$$\mathcal{S}_{t,t'} = \mathcal{I}_W(\text{net}(\mathcal{S}_{t,t'}))$$

où net est l'interprétation qui laisse la structure invariante si ce n'est pour les symboles d'arité 0 qui ont pour nouvelle interprétation \emptyset .

Il reste maintenant à traiter l'élément \emptyset de $\mathcal{S}_{t,t'}$. Pour cela, on restreint d'abord la structure aux éléments de l'interprétation de $\#$, c'est-à-dire que l'on élimine l'élément \emptyset , puis on insère un nouvel élément pour le remplacer. Le symbole

Démonstration du lemme 5.30

Considérons le transducteur déterministe \mathbf{t}_0 d'état initial q suivant :

$$\begin{aligned} q(t) &= f_i(q'(t)) && \text{si } t \in L_i \text{ pour } i \in [k] \\ q'(p(t_1, t_2)) &= \mathcal{I}_p(t_1 \otimes_g t_2) \\ q'(c) &= \mathcal{S}_{c,c} \end{aligned}$$

où k , L_i et f_i sont définis par le lemme 5.31 et \mathcal{I}_p est l'interprétation du lemme 5.32.

Soit $t \in L_i$. On définit les relations continues suivantes :

$$\begin{aligned} \mathcal{R}[t](t', t'') &\text{ si } \mathcal{S}_{t,t'} \approx \llbracket t'' \rrbracket , \\ \mathcal{R}'[t](t', t'') &\text{ si } \mathcal{S}_{t,t'} \approx f_i(\llbracket t'' \rrbracket) . \end{aligned}$$

Il s'agit d'appliquer le lemme 3.59. La relation \mathcal{R} correspond à l'état q et la relation \mathcal{R}' à l'état q' . Le lemme 5.31 montre que l'ensemble inévitable $\{q'\}$ remplit les conditions de l'hypothèse 3 du lemme 3.59. Les lemmes 5.31 et 5.32 établissent les invariants correspondant aux deux premières règles. La troisième transition est évidente. Le transducteur \mathbf{t}_0 satisfait donc pour tout \mathcal{F} -terme t l'isomorphisme suivant :

$$\mathcal{S}_t \approx \llbracket \mathbf{t}_0(t) \rrbracket .$$

Pour retrouver \mathcal{I}_F , il suffit de restreindre les éléments à $\langle \phi_\delta, 1, [1] \rangle$, puis d'appliquer l'interprétation suivante :

$$\mathcal{I} = \left| R(x_1, \dots, x_{|R|}) : \langle \phi_R, |R|, \llbracket R \rrbracket \rangle (x_1, \dots, x_{|R|}), R \in \Sigma, R \neq \delta \right| .$$

Posons $\mathbf{t} = \mathcal{I} \circ \text{restrict}[\langle \phi_\delta, 1, [1] \rangle] \circ \mathbf{t}_0$. Ainsi, pour tout \mathcal{F} -terme t , $\llbracket \mathbf{t}(t) \rrbracket \approx \mathcal{I}_F(t)$. Le lemme 11 appliqué à \mathbf{t} nous fournit alors une transduction monadique \mathbf{t}^* telle que pour tout \mathcal{F} -système d'équations, $\text{Solution}(\mathbf{t}^*(S)) \approx \mathcal{I}_F(\text{Depliage}(S))$.

5.2.4 Structures automatiques

La terminologie automatique [KN94] correspond originellement à des structures dont l'univers est un ensemble rationnel de mots et pour lesquelles les relations sont définies au moyen d'automates (de transducteurs synchronisés de mots). Dans notre cas, une structure admet une *présentation automatique* si elle admet une présentation terme-automatique définie sur des alphabet ne contenant que des symboles unaires et des constantes. Il s'agit d'une représentation interne. Une représentation équationnelle et une représentation par transformations de la famille des structures automatiques sont données ci-dessous. Nous ne précisons pas les démonstrations. Celles-ci sont très similaires aux précédentes.

Représentation équationnelle

Les opérateurs VRS'_1 sont les suivants.

- La structure à un élément.
- L'union disjointe.
- L'interprétation booléenne positive.
- Le produit synchrone *avec une structure finie*.
- La restriction à l'interprétation d'un symbole relationnel unaire.

Ainsi, la seule distinction par rapport au cas des structures à présentation terme-automatique concerne l'utilisation restreinte du produit.

Nous pouvons maintenant énoncer le lemme attendu.

Propriété 5.33 *Une structure admet une présentation automatique si et seulement si elle est solution d'un VRS'_1 -système fini d'équations.*

Notons que des résultats de normalisation comparables à ceux concernant les VRS' -systèmes d'équations sont possibles.

Représentation par transformation

La représentation par transformation est elle aussi très simple.

Propriété 5.34 *Une structure admet une présentation automatique si et seulement si elle est obtenue par interprétation monadique des parties finies de la demi-droite des entiers.*

Chapitre 6

Graphes de réécriture suffixe de termes

Nous avons étudié au paragraphe précédent les structures solutions de VR'-systèmes d'équations étendus par (principalement) un opérateur de produit synchrone. Il s'agit d'une extension naturelle de ces opérateurs. Pourtant, il semble qu'il y ait de la place entre ces deux familles de structures. En particulier, si l'on considère la représentation interne, entre la réécriture suffixe de mots et les relations terme-synchronisées, il est naturel de considérer des relations de réécriture suffixe de termes. Il s'agit du propos de ce paragraphe. Cette étude n'est menée que sur les graphes et non sur les structures.

La réécriture suffixe de termes procure une représentation interne de cette famille de graphes (paragraphe 6.1). Une représentation équationnelle est présentée au paragraphe 6.2. L'équivalence de ces deux représentations est établie au paragraphe 6.3. Ces graphes possèdent des propriétés de décidabilité plus nombreuses que les graphes termes-automatiques. En particulier, ils possèdent une théorie au premier ordre avec accessibilité décidable (paragraphe 6.4). Nous montrons ensuite un résultat similaire à celui de Barthelmann pour les graphes préfixes reconnaissables : les graphes de réécriture suffixe de termes de largeur arborescente bornée sont les graphes HR-équationnels (paragraphe 6.5). Enfin, nous montrons qu'une extension naturelle de ce dernier résultat (les graphes de réécriture suffixe de termes de largeur de clique bornée sont-ils les graphes préfixes reconnaissables?) n'est pas vérifiée (paragraphe 6.6).

6.1 Représentation interne

Il s'agit ici de proposer une définition d'une famille de graphes obtenus par réécriture suffixe de termes. Comme pour les graphes de réécriture suffixe de mots, plusieurs approches successives ont été proposées. Nous les présentons tout d'abord

(paragraphe 6.1.1). Le paragraphe 6.1.2 présente quant à lui la définition que nous considérons par la suite.

6.1.1 Contexte

Étant donné un alphabet gradué \mathcal{F} , un *système de réécriture suffixe* est un ensemble de règles $\Delta \subseteq \mathcal{T}(\mathcal{F}) \times \mathcal{T}(\mathcal{F})$. La relation de réécriture suffixe par Δ , \rightarrow_Δ , reliant les \mathcal{F} -termes, est définie par récurrence comme suit:

- si $(t, t') \in \Delta$ alors $t \rightarrow_\Delta t'$,
- si $t = f(t_1, \dots, t_{|f|})$ et $t' = f(t'_1, \dots, t'_{|f|})$ pour $f \in \mathcal{F}$ et il existe i tel que $t_i \rightarrow_\Delta t'_i$ et pour tout $i \neq j$, $t_j = t'_j$ alors $t \rightarrow_\Delta t'$.

La relation d'accessibilité par Δ , notée \rightarrow_Δ^* , est la clôture réflexive et transitive de \rightarrow_Δ .

Théorème 30 (Dauchet et Tison [DT90]) *Pour \mathcal{F} un alphabet gradué, les graphes dont*

- *l'univers est $\mathcal{T}(\mathcal{F})$,*
- *les arcs sont décrits par réécriture suffixe de termes ou par accessibilité d'une relation de réécriture suffixe de terme,*

ont une théorie au premier ordre décidable.

Ce résultat repose sur le fait que ces graphes admettent une présentation terme-automatique. Le résultat est alors une conséquence directe du théorème de décidabilité de la théorie du premier ordre sur ces structures 5.19.

La définition des relations de réécriture suffixe de termes est tout à fait similaire à celle des relations de réécriture suffixe de mots. En fait, il s'agit d'une extension. On a vu dans le cadre des mots qu'il était naturel pour construire des graphes (ou des structures) de considérer un ensemble rationnel de sommets (resp. un univers rationnel). Ce type de restriction ne nous convient pas dans le cas présent car il fait perdre le résultat de décidabilité du théorème 30.

Théorème 31 (Löding [LÖ2c]) *Il existe un graphe dont l'ensemble des sommets est une partie rationnelle de $\mathcal{T}(\mathcal{F})$, les interprétations des arcs sont des relations de réécriture suffixe et pour lequel l'accessibilité n'est pas décidable.*

La solution utilisée par Löding pour éviter de perdre les propriétés de décidabilité en effectuant une restriction consiste à fixer un terme particulier, et à restreindre le graphe aux sommets accessibles à partir de ce sommet. Il suit en cela l'orientation des travaux originaux de Muller et Schupp.

Formellement, un *graphe de réécriture suffixe avec restriction par accessibilité* (originellement GTRS dans [LÖ2a]) est décrit par un quadruplet $P = (\mathcal{F}, A, \Delta, t_0)$ où \mathcal{F} est un alphabet gradué, A est un ensemble d'étiquettes, t_0 est un \mathcal{F} -terme et Δ est une application de A dans les parties finies de $\mathcal{T}(\mathcal{F}) \times \mathcal{T}(\mathcal{F})$. Le graphe $[P]$

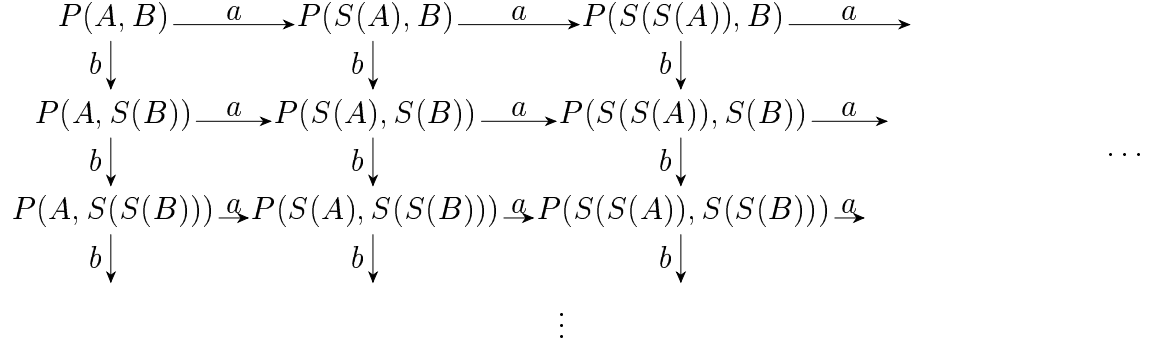


FIG. 6.1 – Grille infinie comme graphe de réécriture suffixe restreint par accessibilité.

représenté par P a alors pour ensemble de sommets $V_{[P]} = \{t \mid t_0 \xrightarrow{*}_{\cup_{a \in A} \Delta(a)} t\}$ et un arc étiqueté par $a \in A$ relie $t \in V_{[P]}$ à $t' \in V_{[P]}$ si $t \xrightarrow{\Delta(a)} t'$.

Exemple 6.1 (grille infinie) Considérons l'alphabet gradué $\mathcal{F} = \{P:2, S:1, A:0, B:0\}$, l'ensemble d'étiquettes $A = \{a, b\}$, les règles de réécriture Δ sont définies par $\Delta(a) = \{(A, S(A))\}$ et $\Delta(b) = \{(B, S(B))\}$ et l'état initial $t_0 = P(A, B)$. Le graphe présenté par $(\mathcal{F}, A, \Delta, t_0)$ est illustré à la figure 6.1.

On déduit de cet exemple et de la propriété 2.9 que les graphes de réécriture suffixe ne possèdent pas une théorie monadique décidable. Par conséquence, ils ne sont pas inclus dans la famille des graphes préfixes reconnaissables.

En fait, l'inclusion inverse n'est pas non plus vérifiée. En effet, la définition des graphes de réécriture suffixe restreint par accessibilité ne produit que des graphes connexes alors que certains graphes préfixes reconnaissables ne le sont pas. De plus, les graphes de réécriture suffixe restreint par accessibilité sont de degré fini (non nécessairement borné), alors que ce n'est pas nécessairement le cas des graphes préfixes reconnaissables.

Le paragraphe suivant montre comment définir une famille de graphes reposant sur la réécriture suffixe de termes et étendant les graphes préfixes reconnaissables.

6.1.2 Graphes de réécriture suffixe de termes

Nous présentons dans ce paragraphe une famille de graphes étendant les graphes préfixes reconnaissables tout en étant définie par réécriture suffixe de termes.

Une première extension consiste à considérer des règles de réécriture reconnaissables (au lieu de finies). Rappelons que pour \mathcal{F} un alphabet gradué, une partie de $\mathcal{T}(\mathcal{F}) \times \mathcal{T}(\mathcal{F})$ est reconnaissable si elle est égale à une union finie de produits

de parties reconnaissables de $\mathcal{T}(\mathcal{F})$ (les parties rationnelles).

Engelfriet montre que les relations de réécriture suffixe reconnaissables partagent les mêmes propriétés de clôture transitive que les relations de réécriture suffixe finies [Eng99]. Cela signifie qu'autoriser $\Delta(a)$ à être une partie reconnaissable dans la définition des graphes de réécriture suffixe avec restriction par accessibilité préserve les propriétés de décidabilité.

Il reste toujours le problème de la restriction. Dans la définition de Löding, seuls les sommets accessibles à partir de la racine sont considérés. Cette solution ne permet pas de capturer les graphes préfixes reconnaissables. La solution proposée ici consiste à utiliser un ensemble de sommets reconnu par un automate d'arbres déterministe descendant [Col02a].

Ainsi, un *graphe de réécriture suffixe* est décrit par la donnée d'un quadruplet $P = (\mathcal{F}, A, \Delta, L)$ où \mathcal{F} est un alphabet gradué, A est un ensemble d'étiquettes, $L \subseteq \mathcal{T}(\mathcal{F})$ est le langage accepté par un automate déterministe descendant et Δ est une application de A dans les parties reconnaissables de $\mathcal{T}(\mathcal{F}) \times \mathcal{T}(\mathcal{F})$. Le graphe $[P]$ présenté par P a alors pour ensemble de sommets $V_{[P]} = L$, et un arc étiqueté par a relie un sommet t à un sommet t' si $t \rightarrow_{\Delta(a)} t'$.

On a alors bien la propriété souhaitée de capturer les graphes préfixes reconnaissables.

Propriété 6.2 *Les graphes préfixes reconnaissables sont, à isomorphisme près, des graphes de réécriture suffixe.*

Il suffit pour montrer cela d'utiliser un codage direct des mots dans les termes. Tout fonctionne bien car pour les termes ne contenant que des symboles d'arité 0 ou 1, la notion de rationalité coïncide avec les langages reconnus par les automates déterministes descendants.

En fait, il est encore plus simple d'utiliser la représentation équationnelle présentée au paragraphe 6.2 pour établir ce résultat.

Exemple 6.3 *Reprenons l'exemple 6.1. L'ensemble des sommets de ce graphe est l'ensemble des termes $\{P(S^n(A), S^m(B)) \mid n, m \in \mathbb{N}\}$ (où S^n représente naturellement la puissance n -ième de l'application $t \mapsto S(t)$). Ce langage de termes est reconnu par un automate déterministe descendant. En effet, il s'agit de la valeur de X solution du système d'équations suivant :*

$$\begin{aligned} X &= P(Y, Z) \\ Y &= A + S(Y) \\ Z &= B + S(Z) . \end{aligned}$$

La grille présentée à la figure 6.1 est donc aussi un graphe de réécriture suffixe au sens de ce paragraphe.

Remarque 6.4 *Une question naturelle est de savoir si tous les graphes de réécriture suffixe restreint par accessibilité sont des graphes de réécriture suffixe de termes comme définis ici. Une réponse négative à cette question a été établie par Löding. Nous ne la présentons pas ici.*

6.2 Représentation équationnelle

Tout comme les opérateurs VRS' étaient une extension des opérateurs VR' par un opérateur de produit, les opérateurs VRA sont une extension des opérateurs VR par un opérateur de produit, asynchrone cette fois-ci.

Soit C un ensemble de couleurs et \mathbf{A} un ensemble d'étiquettes. Les opérateurs $\text{VRA}[C, \mathbf{A}]$ (ou plus simplement VRA) sont les suivants :

- Le graphe coloré à un sommet de couleur $c \in C$ noté $\text{sommet}[c]$,
- L'union disjointe de graphes colorés \oplus ,
- L'ajout d'arcs étiquetés par $a \in \mathbf{A}$ entre un sommet de couleur $c \in C$ et un sommet de couleur $c' \in C$, noté $\text{arc}[c, a, c']$,
- Le recoloriage par une application α de C dans C , noté $\text{recol}[\alpha]$,
- Le produit asynchrone \square_η où η est une application de C^2 dans C . L'opérateur \square_η appliqué à deux graphes colorés \mathcal{G}_1 et \mathcal{G}_2 produit le graphe coloré $\mathcal{H} = \mathcal{G}_1 \square_\eta \mathcal{G}_2$ défini par :

$$\begin{aligned} \mathcal{U}_{\mathcal{H}} &= \mathcal{U}_{\mathcal{G}_1} \times \mathcal{U}_{\mathcal{G}_2} , \\ \text{pour tout } c \in C, \quad c^{\mathcal{H}} &= \{(v_1, v_2) \in c_1^{\mathcal{G}_1} \times c_2^{\mathcal{G}_2} \mid (c_1, c_2) \in \eta^{-1}(c)\} , \\ \text{pour tout } a \in \mathbf{A}, \quad \xrightarrow[a]{\mathcal{H}} &= \{((v_1, v_2), (v'_1, v_2)) \mid v_1 \xrightarrow[a]{\mathcal{G}_1} v'_1\} \\ &\quad \cup \{((v_1, v_2), (v_1, v'_2)) \mid v_2 \xrightarrow[a]{\mathcal{G}_2} v'_2\} . \end{aligned}$$

Exemple 6.5 (Grille infinie) *On se fixe l'ensemble d'étiquettes $A = \{a, b\}$ et l'ensemble des couleurs $C = \{0, 1, 2, 3\}$. La demi-droite des entiers étiquetée par a est solution de l'équation (sur les opérateurs VR) suivante :*

$$X_a = \text{recol} \begin{bmatrix} 0 \rightarrow 1 \\ 1 \rightarrow 1 \\ 2 \rightarrow 0 \end{bmatrix} (\text{arc}[2, a, 0](X_a \oplus \text{sommet}[2])) .$$

Une équation similaire sur la variable X_b a pour solution la demi-droite des entiers étiquetée par b . On construit alors le graphe solution de l'équation supplémentaire $X = X_a \square_\eta X_b$ où $\eta(n, n') = n + n'$. Le résultat est présenté à la figure 6.2.

Ainsi, si l'on compare le graphe de cet exemple à celui de l'exemple 6.1 (figure 6.1), ils sont isomorphes (pourvu que l'on omette les couleurs). On remarque aussi

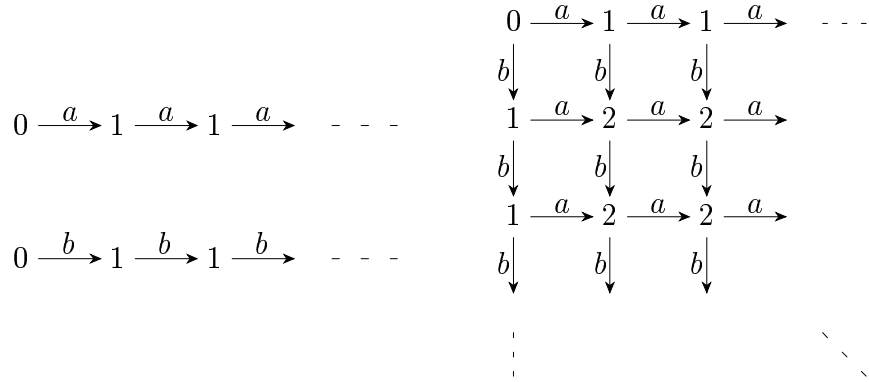


FIG. 6.2 – Les demi-droites des entiers étiquetées par a et b , et leur produit asynchrone.

que l'ensemble des sommets d'une couleur donnée dans la grille de la figure 6.2 correspond à un ensemble rationnel de termes à la figure 6.1. Cet ensemble n'est pas nécessairement accepté par un automate déterministe descendant (considérer la couleur 1 par exemple). Le paragraphe suivant a pour objet de généraliser cet isomorphisme.

6.3 Équivalence de représentations

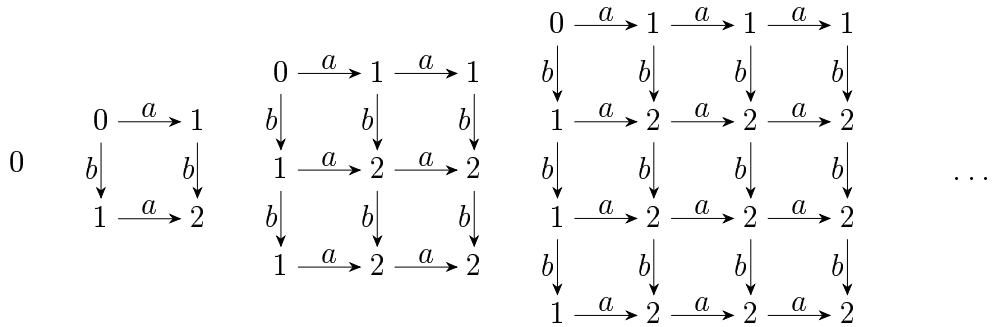


FIG. 6.3 – Union disjointe de grilles carrées colorées.

Indiquons pour commencer le résultat démontré dans ce paragraphe (il s'agit en fait d'une extension du résultat présenté dans [Col02a]).

Théorème 32 *Les familles de graphes suivantes sont isomorphes :*

- les solutions des VRA-systèmes d'équations,

- les graphes de réécriture suffixe reconnaissable de termes restreints par les solutions d'Auto $_{\text{dd}}^{\mathcal{T}}$ -systèmes d'équations,

et ces conversions sont effectives.

Les détails concernant l'effectivité des conversions se trouvent dans les deux lemmes 6.8 et 6.11 dont le théorème 32 est la combinaison.

Commençons par illustrer ce résultat.

Exemple 6.6 *Considérons de nouveaux l'ensemble des couleurs $C = \{0, 1, 2, 3\}$ et l'ensemble des étiquettes $A = \{a, b\}$. On définit pour $n \in \mathbb{N}$ le graphe $\mathcal{G}[a]_n$ comme suit:*

$$\mathcal{G}[a]_n = \overbrace{0 \xrightarrow{a} 1 \xrightarrow{a} 1 \cdots 1 \xrightarrow{a} 1}^{n+1 \text{ sommets}} .$$

Ce graphe se définit par récurrence en utilisant les équations suivantes :

$$\begin{aligned} \mathcal{G}[a]_0 &= \text{sommet}[0] , \\ \mathcal{G}[a]_{n+1} &= \text{recol} \begin{bmatrix} 0 \rightarrow 1 \\ 1 \rightarrow 1 \\ 2 \rightarrow 0 \end{bmatrix} (\text{arc}[2, a, 0](\text{sommet}[2] \oplus \mathcal{G}[a]_n)) . \end{aligned}$$

Des équations comparables permettent de décrire les graphes $\mathcal{G}[b]_n$ définis similairement. On peut maintenant construire le graphe \mathcal{H}_n par l'équation suivante:

$$\mathcal{H}_n = \mathcal{H}_{n+1} \oplus (\mathcal{G}[a]_n \square_{\eta} \mathcal{G}[b]_n) .$$

Le graphe \mathcal{H}_0 décrit par ces équations est présenté à la figure 6.3. Il est formé de l'union disjointe des graphes $\mathcal{G}[a]_n \square_{\eta} \mathcal{G}[b]_n$, chacun étant une grille carrée de dimension n .

Exemple 6.7 *Le même résultat peut être obtenu par un système de réécriture suffixe restreint par un langage algébrique déterministe. Reprenons pour cela l'alphabet, les étiquettes et les règles de réécriture de l'exemple 6.1, mais considérons le langage suivant comme ensemble de sommets :*

$$L = \{S^k(P(S^n(A), S^m(B))) \mid k \in \mathbb{N}, 0 \leq n < k, 0 \leq m < k\} .$$

Le graphe obtenu est de nouveau l'union disjointe des grilles carrées (cf. figure 6.3). L'entier k permet de numéroter la grille. Les contraintes $0 \leq n < k$ et $0 \leq m < k$ assurent que cette grille est bien carrée de taille $k \times k$.

Il nous reste à montrer que le langage L est solution d'un Auto $_{\text{dd}}^{\mathcal{T}}$ -système d'équations. Pour cela, commençons par construire les langages U_k pour $k \in \mathbb{N}$ définis par $L[A]_k = \{S^n(A) \mid 0 \leq n < k\}$. Ces langages s'obtiennent comme solution des équations suivantes :

$$\begin{aligned} L[A]_0 &= \emptyset \\ L[A]_{k+1} &= S(L[A]_k) + A \end{aligned}$$

Des équations similaires produisent le langage $L[B]_k$ (définis comme on s'y attend). Le langage $L_l = \{S^k(P(S^n(A), S^m(B))) \mid k \in \mathbb{N}, 0 \leq n < k+l, 0 \leq m < k+l\}$ pour $l \in \mathbb{N}$ est alors défini par les équations suivantes :

$$L_l = S(L_{l+1}) + P(L[A]_l, L[B]_l) .$$

Les systèmes d'équations apparaissant dans les exemples 6.6 et 6.7 sont structurellement similaires (plus concrètement, il est possible de passer de l'un à l'autre par une transduction monadique). Ce n'est bien entendu pas anodin et les paragraphes 6.3.1 et 6.3.2 établissent ces équivalences dans le cas général.

6.3.1 Des équations aux récritures

Pour la suite de ce paragraphe, on se fixe un ensemble C de couleurs et un ensemble A d'étiquettes. On construit l'alphabet et les relations de récriture indépendamment du système d'équations. Celui-ci ne sert qu'à déterminer l'ensemble des sommets.

On considère donc l'alphabet gradué \mathcal{F} suivant :

$$\begin{aligned} \mathcal{F} = & \{U_1 : 1, U_2 : 1\} \\ & \cup \{P[\eta] : 2 \mid \eta \in C^{C \times C}\} \\ & \cup \{A[c, a, c'] : 1 \mid c, c' \in C, a \in A\} \\ & \cup \{R[\alpha] : 1 \mid \alpha \in C^C\} \\ & \cup \{S[c] : 0 \mid c \in C\} . \end{aligned}$$

Bien entendu, chacun de ces symboles correspond à un des opérateurs $\text{VRA}[C, A]$, à savoir U_1 et U_2 pour \oplus , $P[\eta]$ pour \square_η , $A[c, a, c']$ pour l'opérateur $\text{arc}[c, a, c']$, $R[\alpha]$ pour $\text{recol}[\alpha]$ et enfin $S[c]$ pour $\text{sommet}[c]$. Il est à noter que le produit et l'union disjointe ont un traitement différent correspondant à leurs sémantiques distinctes.

Construisons maintenant les ensembles reconnaissables. Ceux-ci correspondent directement aux couleurs. On définit donc l' $\text{Auto}^\mathcal{T}$ -système d'équations dont les variables sont les L_c pour $c \in C$ et les équations sont les suivantes :

$$\begin{aligned} L_c = & U_1(L_c) + U_2(L_c) \\ & + \sum_{\substack{\eta \in C^{C \times C} \\ \eta(c_1, c_2) = c}} P[\eta](L_{c_1}, L_{c_2}) \\ & + \sum_{\substack{c_1, c_2 \in C \\ a \in A}} A[c_1, a, c_2](L_c) \\ & + \sum_{\substack{\alpha \in C^C \\ \alpha(c') = c}} R[\alpha](L_{c'}) \\ & + S[c] . \end{aligned}$$

Les relations de réécriture reconnaissables sont alors les suivantes.

$$\Delta(a) = \bigcup_{c_1, c_2 \in C} A[c_1, a, c_2](L_{c_1}) \times A[c_1, a, c_2](L_{c_2}) .$$

Lemme 6.8 *Il existe un transducteur déterministe \mathbf{t} des $\mathbf{VRA}[C, A]$ -termes dans les $\text{Auto}_{\text{dd}}^{\mathcal{T}}[\mathcal{F}]$ -termes tel que pour tout $\mathbf{VRA}[C, A]$ -terme t , $\llbracket t \rrbracket$ est isomorphe à la structure de réécriture suffixe de termes de présentation $(\mathcal{F}, A, \Delta, \llbracket \mathbf{t}(t) \rrbracket)$.*

Construction. Le transducteur possède un unique état (initial) q et ses règles de transition sont les suivantes :

$$\begin{aligned} q(x \oplus y) &\rightarrow U_1(q(x)) + U_2(q(y)) \\ q(x \square_{\eta} y) &\rightarrow P[\eta](q(x), q(y)) \\ q(\text{arc}[c, a, c'](x)) &\rightarrow A[c, a, c'](q(x)) \\ q(\text{recol}[\alpha](x)) &\rightarrow R[\alpha](q(x)) \\ q(\text{sommet}[c](x)) &\rightarrow S[c] \end{aligned}$$

On montre alors par récurrence sur $n \in \mathbb{N}$ que :

$$\text{nb}(\llbracket t \downarrow_n \rrbracket) \approx [(\mathcal{F}, A, \Delta, \llbracket \mathbf{t}(t) \downarrow_n \rrbracket)] .$$

□

Remarque 6.9 *Dans cette construction, si le \mathbf{VRA} -terme t ne contient pas d'opérateur produit, alors l'alphabet utilisé par le terme $\mathbf{t}(t)$ ne possède que des symboles d'arité 0 ou 1. Le terme résultat peut donc être assimilé à un mot. Le graphe résultat est alors suffixe reconnaissable restreint par un langage déterministe.*

6.3.2 De la réécriture aux équations

Considérons un alphabet gradué \mathcal{F} , un ensemble d'étiquettes A et une application Δ de A dans les parties reconnaissables de $\mathcal{T}(\mathcal{F}) \times \mathcal{T}(\mathcal{F})$.

À chaque étiquette $a \in A$ correspond par Δ une partie reconnaissable de $\mathcal{T}(\mathcal{F}) \times \mathcal{T}(\mathcal{F})$.

Lemme 6.10 *Il existe un automate d'arbres (Q, δ) (sans états terminaux) déterministe ascendant et complet tel que pour tout a , il existe $\Delta'(a) \subseteq Q \times Q$ tel que :*

$$\Delta(a) = \bigcup_{(q, q') \in \Delta'(a)} \delta^{-1}(q) \times \delta^{-1}(q') .$$

Preuve. En considérant l'ensemble des automates impliqués dans la description de Δ comme un unique automate, puis en le déterminisant. \square

Lemme 6.11 *Il existe un transducteur déterministe des $\text{Auto}_{\text{dd}}^{\mathcal{T}}[\mathcal{F}]$ -termes vers les VRA-termes tel que pour tout $\text{Auto}_{\text{dd}}^{\mathcal{T}}[\mathcal{F}]$ -terme t ,*

$$[(\mathcal{F}, A, \Delta, \llbracket t \rrbracket)] \approx \text{nb}(\llbracket \mathbf{t}(t) \rrbracket) .$$

Construction. Soient Q et δ tels qu'obtenus par le lemme 6.10.

L'ensemble des couleurs du terme produit par le transducteur déterministe est Q .

On utilise une version généralisée à toute arité des symboles \oplus et \square (y compris en ce qui concene le paramètre η de \square).

Le tansducteur déterministe est alors défini comme suit:

$$q \left(\sum_{f \in F} f(x_{f,1}, \dots, x_{f,|f|}) \right) = \text{arc}[D] \left(\bigoplus_{f \in F} \square_{\eta_{f,i \in [|f|]}} q(x_{f,i}) \right)$$

avec $D = \{(q, a, q') \mid (q, q') \in \Delta'(a)\}$

$$\text{et } \eta_f(q_1, \dots, q_{|f|}) = q \quad \text{tel que } (f(q_1, \dots, q_{|f|}), q) \in \delta.$$

\square

6.4 Premier ordre avec accessibilité

La logique du premier ordre avec accessibilité est une simple extension de la logique du premier ordre sur les graphes par un opérateur d'accessibilité. Formellement, étant donné un ensemble d'étiquettes $B \subseteq A$, la relation binaire $\xrightarrow{B^*}$ pour interprétation l'ensemble des couples de sommets (u, v) tels qu'il existe un chemin étiqueté par B^* reliant u à v .

Nous montrons ici que l'ensemble des modèles d'une formule du premier ordre augmenté des prédicats d'accessibilité sont VRA-reconnaissable. En fait, le principe de la preuve est simplement de ramener au moyen d'un transducteur déterministe le problème à un problème de décidabilité pour un VRS'-système d'équations.

L'hypothèse de récurrence

Supposons l'ensemble d'étiquettes \mathbf{A} fixé. On s'intéresse aux structures sur l'ensemble des symboles relationnels suivants :

$$\begin{array}{ll}
 \Sigma = \{ \xrightarrow{a} \mid a \in \mathbf{A} \} & \text{avec } | \xrightarrow{a} | = 2 , \\
 \uplus C & |c| = 1 , \\
 \uplus \{ \langle B^* \rangle \mid B \subseteq \mathbf{A} \} & | \langle B^* \rangle | = 2 , \\
 \uplus \{ \langle cB^* \rangle \mid B \subseteq \mathbf{A}, c \in C \} & | \langle cB^* \rangle | = 1 , \\
 \uplus \{ \langle B^*c \rangle \mid B \subseteq \mathbf{A}, c \in C \} & | \langle B^*c \rangle | = 1 , \\
 \uplus \{ \langle cB^*c' \rangle \mid c, c' \in C, B \subseteq \mathbf{A} \} & \text{et } | \langle cB^*c' \rangle | = 0 .
 \end{array}$$

L'objectif est, à partir d'un graphe coloré par C et étiqueté par A , de construire une Σ' -structure codant à la fois le graphe et les relations d'accessibilité. Les symboles \xrightarrow{a} représentent les arcs originaux du graphe et les symboles binaires $\langle B^* \rangle$ identifient les couples de sommets reliés par un chemin étiqueté par un mot de B^* . Les symboles $\langle cB^* \rangle$, $\langle B^*c \rangle$ et $\langle cB^*c' \rangle$ servent aux constructions intermédiaires : ils identifient les chemins dont l'origine et/ou la destination est/sont d'une certaine couleur.

Formellement, étant donné un graphe \mathcal{G} coloré par C et étiqueté par A , on construit la structure $*(\mathcal{G})$ définie comme suit.

$$\begin{aligned}
 \mathcal{U}_{*(\mathcal{G})} &= V_{\mathcal{G}} , \\
 c^{*(\mathcal{G})} &= c^{\mathcal{G}} , \\
 \xrightarrow{a}_{*(\mathcal{G})} &= \xrightarrow{a}_{\mathcal{G}} , \\
 \langle B^* \rangle^{*(\mathcal{G})} &= \{ (v, v') \mid \exists w \in B^*. v \xrightarrow{w}_{\mathcal{G}} v' \} , \\
 \langle cB^* \rangle^{*(\mathcal{G})} &= \{ v \mid \exists v' \in c^{\mathcal{G}}, w \in B^*. v' \xrightarrow{w}_{\mathcal{G}} v \} , \\
 \langle B^*c \rangle^{*(\mathcal{G})} &= \{ v \mid \exists v' \in c^{\mathcal{G}}, w \in B^*. v \xrightarrow{w}_{\mathcal{G}} v' \} , \\
 \langle cB^*c' \rangle^{*(\mathcal{G})} &= \{ () \mid \exists v \in c^{\mathcal{G}}. \exists v' \in c'^{\mathcal{G}}. \exists w \in B^*. v \xrightarrow{w}_{\mathcal{G}} v' \} .
 \end{aligned}$$

Récurrence

On s'intéresse maintenant à transformer un VRA-système d'équations en un VRS-système d'équations tel que la solution du second s'obtienne par application de $*$ à la solution du premier. Cette traduction est effectuée par un transducteur déterministe. Il s'agit donc de présenter les propriétés permettant de valider cette construction, et qui sont données par les lemmes 6.12, 6.13, 6.14 et 6.15.

Lemme 6.12 *Pour toute application α de C dans C , il existe un interprétation $\mathcal{I}_{\text{recol}[\alpha]}$ booléenne positive des Σ' -structures dans les Σ' -structures telle que pour tout graphe coloré \mathcal{G} , on a :*

$$*(\text{recol}[\alpha](\mathcal{G})) = \mathcal{I}_{\text{recol}[\alpha]}(*(\mathcal{G})) .$$

Construction. On pose simplement

$$\mathcal{I}_{\text{recol}[\alpha]} = \left| \begin{array}{lll} c(x) & : & c'(x) & \alpha(c') = c \\ x \xrightarrow{a} y & : & x \xrightarrow{a} y & \\ \langle B^* \rangle (x, y) & : & \langle B^* \rangle (x, y) & \\ \langle cB^* \rangle (x) & : & \langle c'B^* \rangle (x) & \alpha(c') = c \\ \langle B^*c \rangle (x) & : & \langle B^*c' \rangle (x) & \alpha(c') = c \\ \langle c_1B^*c_2 \rangle & : & \langle c'_1B^*c'_2 \rangle & \alpha(c'_1) = c_1 \text{ et } \alpha(c'_2) = c_2 \end{array} \right| .$$

□

Lemme 6.13 *Pour toutes couleurs $c_1, c_2 \in C$ et toute étiquette $a \in A$, il existe une interprétation $\mathcal{I}_{\text{arc}[c_1, a, c_2]}$ telle que pour tout graphe coloré \mathcal{G} on a :*

$$*(\text{arc}[c_1, a, c_2](\mathcal{G})) = \mathcal{I}_{\text{arc}[c_1, a, c_2]}(*(\mathcal{G}))$$

Preuve.

$$\mathcal{I}_{\text{arc}[c_1, a, c_2]} = \left| \begin{array}{lll} c(x) & : & c(x) \\ x \xrightarrow{b} y & : & x \xrightarrow{b} y & \text{pour } b \in A \\ x \xrightarrow{a} y & : & c_1(x) \wedge c_2(y) \\ \langle B^* \rangle (x, y) & : & \langle B^* \rangle (x, y) \\ \langle B^* \rangle (x, y) & : & \langle B^*c_1 \rangle (x) \wedge \langle c_2B^* \rangle (y) & \text{pour } a \in B \\ \langle cB^* \rangle (x) & : & \langle cB^* \rangle (x) \\ \langle cB^* \rangle (x) & : & \langle cB^*c_1 \rangle \wedge \langle c_2B^* \rangle (x) & \text{pour } a \in B \\ \langle B^*c \rangle (x) & : & \langle B^*c \rangle (x) \\ \langle B^*c \rangle (x) & : & \langle B^*c_1 \rangle (x) \wedge \langle c_2B^*c \rangle & \text{pour } a \in B \\ \langle cB^*c' \rangle & : & \langle cB^*c' \rangle \\ \langle cB^*c' \rangle & : & \langle cB^*c_1 \rangle \wedge \langle c_2B^*c' \rangle & \text{pour } a \in B \end{array} \right| .$$

La construction fonctionne sur la remarque suivante : un chemin entre deux sommets v et v' de $\text{arc}[c_1, a, c_2](\mathcal{G})$ peut être de l'une des deux formes suivantes :

- il s'agit d'un chemin n'empruntant aucun des arcs ajoutés par l'opérateur $\text{arc}[c_1, a, c_2]$, et dans ce cas, il était déjà présent dans \mathcal{G} ,

– ou bien ce chemin emprunte au moins l'un des arcs ajoutés par $\text{arc}[c_1, a, c_2]$. Dans ce cas, ce chemin s'écrit comme la concaténation de trois chemins de la forme suivante :

1. un chemin entre v et un sommet de couleur c_1 n'empruntant aucun des arcs ajoutés par $\text{arc}[c_1, a, c_2]$; ce chemin est donc déjà présent dans \mathcal{G} ,
2. un chemin entre un sommet de couleur c_1 et un sommet de couleur c_2 ,
3. un chemin entre un sommet de couleur c_2 et v' , n'empruntant aucun des arcs ajoutés par $\text{arc}[c_1, a, c_2]$; ce chemin est donc déjà présent dans \mathcal{G} .

Il suffit pour tester l'existence d'un tel chemin de vérifier la présence du premier et du troisième. Le second est en fait toujours présent d'après la sémantique de $\text{arc}[c_1, a, c_2]$.

Cette décomposition explique les deux cas utilisés pour chacun des symboles de chemin. \square

Lemme 6.14 *Pour tous graphes colorés \mathcal{G} et \mathcal{H} , $*(\mathcal{G} \oplus \mathcal{H}) = *(\mathcal{G}) \oplus *(\mathcal{H})$.*

Preuve. Évident. \square

Le lemme ci-dessous traite du cas du produit.

Lemme 6.15 *Il existe trois interprétations booléennes positives $\mathcal{I}_{\eta,1}$, $\mathcal{I}_{\eta,2}$ et \mathcal{I} telles que pour tous graphes colorés \mathcal{G}_1 et \mathcal{G}_2 , on a*

$$*(\mathcal{G}_2 \square_{\eta} \mathcal{G}_1) = \mathcal{I}_{\eta}(\mathcal{I}_{\eta,1}(*(\mathcal{G}_1)) \otimes \mathcal{I}_{\eta,2}(*(\mathcal{G}_2))) .$$

Construction. On utilise l'équivalence avec le produit généralisé (propriété 2.22).

$$\mathcal{I}_{\eta} = \left| \begin{array}{ll} c(x) & : \quad c_1^1(x) \wedge c_2^2(x) \\ & \quad \text{pour } c = \eta(c_1, c_2) \\ x \xrightarrow{a} y & : \quad x \xrightarrow{a^1} y \wedge x \sim^2 y \\ x \xrightarrow{a} y & : \quad x \sim^1 y \wedge x \xrightarrow{a^2} y \\ \langle B^* \rangle(x, y) & : \quad \langle B^* \rangle^1(x, y) \wedge \langle B^* \rangle^2(x, y) \\ \langle B^* c \rangle(x) & : \quad \langle B^* c_1 \rangle^1(x) \wedge \langle B^* c_2 \rangle^2(x) \\ & \quad \text{pour } c = \eta(c_1, c_2) \\ \langle c B^* \rangle(x) & : \quad \langle c_1 B^* \rangle^1(x, y) \wedge \langle c_2 B^* \rangle^2(x, y) \\ & \quad \text{pour } c = \eta(c_1, c_2) \\ \langle c B^* c' \rangle() & : \quad \langle c_1 B^* c_2 \rangle^1() \wedge \langle c'_1 B^* c'_2 \rangle^2() \\ & \quad \text{pour } c = \eta(c_1, c_2) \wedge c' = \eta(c'_1, c'_2) \end{array} \right| .$$

On a alors $\mathcal{I}_{\eta}(*(\mathcal{G}_1) \otimes_g *(\mathcal{G}_2)) = *(\mathcal{G}_1 \square_{\eta} \mathcal{G}_2)$. \square

Le résultat

Lemme 6.16 *Pour tout ensemble d'étiquettes A et tout ensemble de couleurs C , il existe un transducteur déterministe \mathbf{t} des $\text{VRA}[C, A]$ -termes dans les VRS -termes tel que pour tout $\text{VRA}[C, A]$ -terme t ,*

$$\llbracket \mathbf{t}(t) \rrbracket = *(\llbracket t \rrbracket) .$$

Corollaire 6.17 (Colcombet [Col02a]) *Soit ϕ une formule en logique du premier ordre avec accessibilité étendue par les modalités de comptage \exists^ω et \forall^ω , l'ensemble des modèles de ϕ est VRA -reconnaisable.*

6.5 Largeur arborescente

Dans ce paragraphe, nous énonçons un résultat similaire à celui de Barthelmann (théorème 25) montrant que les graphes préfixes reconnaissables de largeur arborescente bornée sont les graphes HR -équationnels. Un premier résultat dans cette direction concernant la réécriture suffixe de termes restreinte par accessibilité a été établi par Löding.

Théorème 33 (Löding [LÖ2a]) *Les graphes de réécriture suffixe avec restriction par accessibilité et de largeur arborescente bornée sont des graphes préfixes.*

Löding a ultérieurement étendu ce travail aux systèmes de réécriture suffixe reconnaissables [LÖ2b].

Nous montrons ici un résultat similaire, pour les graphes solutions de VRA -systèmes d'équations.

Théorème 34 (Colcombet [Col02a]) *Les graphes solutions de VRA -systèmes finis d'équations de largeur arborescente bornée sont les graphes HR -équationnels.*

Pour montrer ce résultat, il s'agit d'abord d'éliminer les opérateurs produit du système d'équations. Une première étape de normalisation est nécessaire (paragraphe 6.5.1). On obtient alors un système VR -équationnel de largeur arborescente bornée. Le résultat de Barthelmann permet alors de transformer ce système d'équations en un système HR -équationnel.

6.5.1 Croissance pour la largeur arborescente

On cherche dans ce paragraphe à assurer que dans tout «calcul» effectué par le système d'équations, la largeur arborescente du résultat est supérieure à la largeur arborescente des arguments. Ce raisonnement est à la base de la preuve qui suit.

Pour cela, on introduit le préordre \leq sur les graphes colorés défini comme suit. Pour deux graphes colorés \mathcal{G} et \mathcal{H} , $\mathcal{G} \leq \mathcal{H}$ si \mathcal{G} est un sous-graphe de \mathcal{H}

à recoloriage près. Le lemme 6.20 permet d'assurer que, dans un VRA-système d'équations, le résultat de tout opérateur est supérieur à ses arguments pour l'ordre \leq .

Le lemme 6.18 nous enseigne comment se comportent les opérateurs VRA vis à vis de la relation \leq .

Lemme 6.18 *Pour tous graphes colorés \mathcal{G} et \mathcal{H} , toute application α de C dans C , toutes couleurs c et c' , toute étiquette a et toute application η de C^2 dans C , on a,*

$$\begin{array}{ll} \mathcal{G} \leq \text{recol}[\alpha](\mathcal{G}) , & \mathcal{G} \leq \text{arc}[c, a, c'](\mathcal{G}) , \\ \mathcal{G} \leq \mathcal{G} \oplus \mathcal{H} , & \mathcal{H} \leq \mathcal{G} \oplus \mathcal{H} , \\ \text{si } \mathcal{H} \neq \emptyset, \quad \mathcal{G} \leq \mathcal{G} \square_{\eta} \mathcal{H} , & \text{si } \mathcal{G} \neq \emptyset, \quad \mathcal{H} \leq \mathcal{G} \square_{\eta} \mathcal{H} . \end{array}$$

Preuve. Évident au cas par cas. □

Il faut donc éliminer le cas du graphe vide \emptyset si l'on veut garantir que le résultat de l'application d'un opérateur de VRA produit un résultat supérieur pour \leq à ses arguments. Pour cela, on dit qu'un VRA-terme t est *normalisé* si pour tout sous-terme t' de t , $\llbracket t' \rrbracket \neq \emptyset$. On obtient alors directement le lemme suivant.

Lemme 6.19 *Pour tout VRA-terme normalisé t et tout sous-terme t' de t ,*

$$\llbracket t' \rrbracket \leq \llbracket t \rrbracket .$$

Il s'agit donc de normaliser les termes.

Lemme 6.20 *Soit C un ensemble fini de couleurs et A un ensemble fini d'étiquettes. Il existe un transducteur déterministe des $\text{VRA}[C, A]$ -termes dans les VRA -termes tel que pour tout $\text{VRA}[C, A]$ -terme t , si $\llbracket t \rrbracket \neq \emptyset$ alors $\mathbf{t}(t)$ est défini, normalisé et satisfait $\llbracket t \rrbracket \approx \llbracket \mathbf{t}(t) \rrbracket$.*

Construction. L'identité sur C est notée id . On utilise le transducteur déterministe à un seul état suivant :

$$\begin{array}{l} q(x \square_{\eta} y) \rightarrow q(x) \square_{\eta} q(y) , \\ q(x \oplus y) \rightarrow \text{recol}[id](q(x)) \quad \text{si } \llbracket y \rrbracket = \emptyset , \\ q(x \oplus y) \rightarrow \text{recol}[id](q(y)) \quad \text{si } \llbracket x \rrbracket = \emptyset , \\ q(\text{arc}[c, a, c'](x)) \rightarrow \text{arc}[c, a, c'](q(x)) , \\ q(\text{recol}[\alpha](x)) \rightarrow \text{recol}[\alpha](q(x)) , \\ q(\text{sommet}[c]) \rightarrow \text{sommet}[c] . \end{array}$$

Les anticipations sont validées par le corollaire 6.17 car être vide s'exprime en logique du premier ordre. L'utilisation de $\text{recol}[id]$ permet de garantir l'hypothèse sur les règles de production du lemme 3.59. \square

Enfin, le lemme 6.21 montre comment l'ordre \leq est relié à la largeur arborescente.

Lemme 6.21 *Soient deux graphes colorés \mathcal{G} et \mathcal{H} ,*

$$\text{si } \mathcal{G} \leq \mathcal{H} \quad \text{alors} \quad LA(\mathcal{G}) \leq LA(\mathcal{H}) .$$

Preuve. Évident¹. \square

Corollaire 6.22 *Si $\llbracket t \rrbracket$ est normalisé et de largeur arborescente bornée, alors pour tout sous-terme t' de t , $\llbracket t' \rrbracket$ est de largeur arborescente bornée.*

6.5.2 Élimination des produits

Il s'agit maintenant de faire disparaître les opérateurs de produit asynchrone d'un terme dont la valeur est de largeur arborescente bornée. Dans la suite de ce paragraphe, on omet le paramètre η des opérateurs \square quand il n'intervient pas dans le raisonnement.

L'élimination se fait en deux étapes. Tout d'abord, il s'agit d'éliminer les opérateurs de produit apparaissant une infinité de fois dans une même branche, c'est à dire les termes de la forme $t = t' \square t''$, où t est un sous-terme de t' ou de t'' . La seconde étape élimine les opérateurs de produit restants.

Étape 1 : élimination des produits récursifs

Considérons un terme normalisé de la forme $t = t' \square t''$ où t est un sous-terme de t' (le cas de t'' étant naturellement symétrique). Le terme t' est normalisé et t est un sous-terme de t' , donc $\llbracket t \rrbracket \leq \llbracket t' \rrbracket$ (lemme 6.19). D'où $\llbracket t \rrbracket \square_\eta \llbracket t'' \rrbracket \leq \llbracket t \rrbracket$. On en déduit que pour tout $n \in \mathbb{N}_+$

$$\square^n(\llbracket t'' \rrbracket) \leq \llbracket t \rrbracket , \quad \text{avec } \square^n(\mathcal{G}) = \overbrace{\mathcal{G} \square \dots \square \mathcal{G}}^{n \text{ fois}} .$$

Comme t est de largeur arborescente bornée, il existe un $k \in \mathbb{N}$ tel que pour tout $n \in \mathbb{N}_+$, $LA(\square^n(\llbracket t'' \rrbracket)) \leq k$. Le lemme 6.23 précise alors la structure de $\llbracket t \rrbracket$.

1. En fait, \leq est un cas particulier de la relation «est un mineur de» qui satisfait la même propriété.

Lemme 6.23 *Soit \mathcal{G} un graphe coloré et k un entier tel que pour tout n*

$$LA(\square^n(\mathcal{G})) \leq k ,$$

alors tous les arcs de \mathcal{G} sont des boucles.

Preuve. Supposons que \mathcal{G} possède un arc qui ne soit pas une boucle. Alors $\square^n(\mathcal{G})$ contient un hypercube de dimension n . Le nombre d'arcs dans un hypercube de dimension n est $n2^{n-1}$, le nombre de sommets est 2^n . Donc le ratio nombre d'arcs/nombre de sommets de l'hypercube de dimension n vaut $n/2$ et n'est donc pas borné. Ainsi, le graphe union de tous les hypercubes n'est pas uniformément creux et donc pas de largeur arborescente bornée. Contradiction. \square

Peu de graphes satisfont la contrainte de ne posséder que des boucles. En fait, ces graphes sont entièrement décrits par le nombre (éventuellement infini) de sommets de chaque type qu'ils contiennent, le type étant formé de la couleur et de l'ensemble des étiquettes sur une boucle de ce sommet. Soit \mathcal{G}_0 un graphe satisfaisant cette contrainte. Il s'agit de remplacer une équation de la forme $X = Y \square \mathcal{G}_0$ par une équation équivalente ne faisant pas intervenir l'opérateur \square . L'opérateur *boucle* permet cette opération. Étant donné un ensemble d'étiquettes $E \subseteq A$, l'opérateur *boucle*[E] ajoute à tous les sommets d'un graphe et toute étiquette $a \in E$ une boucle étiquetée par a sur ce sommet. Les opérateurs VRA augmentés de cet opérateur sont notés (VRA + boucle).

Lemme 6.24 *Soit \mathcal{G} un graphe coloré ne contenant que des boucles et η une application de C^2 dans C alors il existe une application (VRA+boucle)-définissable $f_{\mathcal{G}}$ (resp. $r_{\mathcal{G}}$) telle que pour tout graphe coloré \mathcal{H} , on a continuellement :*

$$\mathcal{G} \square_{\eta} \mathcal{H} \approx f_{\mathcal{G}}(\mathcal{H}) \quad (\text{resp. } \mathcal{H} \square_{\eta} \mathcal{G} \approx g_{\mathcal{G}}(\mathcal{H})).$$

Preuve. (a) Supposons que \mathcal{G} ne soit formé que d'un sommet de couleur c_0 . Soit E l'ensemble des étiquettes apparaissant sur ce sommet. On pose $f_{\mathcal{G}} = \text{boucle}[E] \circ \text{recol}[\alpha]$ avec pour tout $c \in C$, $\alpha(c) = \eta(c_0, c)$. On a alors pour tout graphe coloré \mathcal{H} , $\mathcal{G}_0 \square_{\eta} \mathcal{H} \approx f_{\mathcal{G}}(\mathcal{H})$.

(b) Si maintenant \mathcal{G} est formé de l'union disjointe d'une infinité de fois un graphe \mathcal{G}_0 ne contenant qu'un seul sommet, alors on définit $f_{\mathcal{G}}$ comme plus petite solution de $f_{\mathcal{G}}(\mathcal{H}) = f_{\mathcal{G}_0}(\mathcal{H}) \oplus f_{\mathcal{G}}(\mathcal{H})$. On a alors pour tout graphe coloré \mathcal{H} $\mathcal{G}_0 \square_{\eta} \mathcal{H} \approx f_{\mathcal{G}}(\mathcal{H})$.

Tout graphe ne contenant que des boucles s'écrit comme une union disjointe finie de graphes des types (a) et (b). On en déduit $f_{\mathcal{G}}$ convenable comme une union finie des applications précédemment définies.

L'application g_G s'obtient similairement. \square

Lemme 6.25 *Pour tout ensemble de couleurs C , tout ensemble d'étiquettes A et tout $\text{VRA}[C, A]$ -terme régulier t tel que $\llbracket t \rrbracket$ est de largeur arborescente bornée, il existe un $(\text{VRA} + \text{boucle})[C, A]$ -terme régulier t' tel que*

- $\llbracket t \rrbracket \approx \llbracket t' \rrbracket$,
- aucune branche de t' ne contient une infinité d'opérateurs \square ,
- t est normalisé.

Preuve. On suppose t normalisé.

Soient $\mathcal{G}_1, \dots, \mathcal{G}_n$ les graphes ne contenant que des boucles apparaissant valeur d'un sous-terme de t . Ceux-ci sont en nombre fini car t est régulier. Bien entendu, et pour tout $i \in [n]$, les graphes isomorphes à \mathcal{G}_i sont $\text{VRA}[C, A]$ -reconnaissables (appliquer par exemple le corollaire 6.17, les graphes \mathcal{G}_i étant axiomatisables en logique du premier ordre augmentée par les modalités de comptage).

On construit alors le transducteur déterministe \mathbf{t} à un seul état q suivant.

$$\begin{array}{ll}
 q(x \square_{\eta} y) \rightarrow f_{\mathcal{G}_i}(q(y)) & \text{si } \llbracket x \rrbracket \approx \mathcal{G}_i \\
 q(x \square_{\eta} y) \rightarrow g_{\mathcal{G}_i}(q(y)) & \text{si } \llbracket y \rrbracket \approx \mathcal{G}_i \\
 q(x \square_{\eta} y) \rightarrow q(x) \square_{\eta} q(y) & \text{sinon} \\
 q(x \oplus y) \rightarrow q(x) \oplus q(y) & \\
 \text{recol}[\alpha](q(x)) \rightarrow \text{recol}[\alpha](q(x)) & \\
 q(\text{arc}[c, a, c'](x)) \rightarrow \text{arc}[c, a, c'](q(x)) & \\
 q(\text{sommet}[c]) \rightarrow \text{sommet}[c] &
 \end{array}$$

Le fait que ce transducteur préserve la valeur ne pose pas de problème (on applique pour cela le lemme 6.24 pour les deux premières règles). Il est aussi évident que \mathbf{t} préserve la normalisation.

Soit $t' = \mathbf{t}(t)$. Supposons que t contienne une branche contenant une infinité d'opérateurs \square . Comme $\llbracket t \rrbracket$ est de largeur arborescente bornée et d'après le lemme 6.23, l'un de ses paramètres est un graphe ne possédant que des boucles. Or ce graphe était déjà présent dans t , donc l'une des deux premières règles du transducteur aurait dû être appliquée. Contradiction. \square

Après application du lemme 6.25, il ne nous reste plus qu'à éliminer les opérateurs de boucle.

Lemme 6.26 *Pour tout ensemble de couleurs C et tout ensemble d'étiquettes A il existe un transducteur déterministe \mathbf{t} des $(\text{VRA} + \text{boucle})[C, A]$ -termes dans les*

VRA-termes tel que pour tout (VRA + boucle)[C, A]-terme t ,

$$\llbracket t \rrbracket = \llbracket \mathbf{t}(t) \rrbracket .$$

De plus, si t ne contient pas de branche contenant une infinité d'opérateurs \square , alors il en est de même pour $\mathbf{t}(t)$. Enfin, si t est normalisé, $\mathbf{t}(t)$ est aussi normalisé.

Construction. On considère le transducteur déterministe possédant $\langle E \rangle$ pour tout $E \subseteq A$ comme états. Ce transducteur collecte les opérateurs **boucle** et les propage jusqu'aux feuilles.

$$\begin{aligned} \langle E \rangle (x \square y) &\rightarrow \langle E \rangle (x) \square \langle \emptyset \rangle (y) \\ \langle E \rangle (x \oplus y) &\rightarrow \langle E \rangle (x) \oplus \langle E \rangle (y) \\ \langle E \rangle (\text{boucle}[F](x)) &\rightarrow \text{recol}[id](\langle E \cup F \rangle (x)) \\ \langle E \rangle (\text{recol}[\alpha](x)) &\rightarrow \text{recol}[\alpha](\langle E \rangle (x)) \\ \langle E \rangle (\text{arc}[c, a, c'](x)) &\rightarrow \text{arc}[c, a, c'](\langle E \rangle (x)) \\ \langle E \rangle (\text{sommet}[c]) &\rightarrow \text{arc}[\{c\} \times D \times \{c'\}](\text{sommet}[c]) \end{aligned}$$

Soit $\langle \emptyset \rangle$ l'état initial. On montre sans difficulté que ce transducteur satisfait les conclusions du lemme. \square

À ce stade de la transformation, on a obtenu à partir d'un VRA-terme régulier dont la valeur est de largeur arborescente bornée un VRA-terme régulier normalisé dont aucune branche ne contient une infinité d'opérateurs \square . L'étape suivante permet d'éliminer les opérateurs \square restants.

Étape 2 : élimination des produits restants

Lemme 6.27 Pour deux graphes connexes \mathcal{G} et \mathcal{H} , on a,

$$LA(\mathcal{G} \square \mathcal{H}) \geq \min(|V_{\mathcal{G}}|, |V_{\mathcal{H}}|) - 1 .$$

Preuve. Chaque sommet du graphe est un couple $(v, v') \in V_{\mathcal{G}} \times V_{\mathcal{H}}$. Appelons ligne (resp. colonne) tout ensemble maximal de sommets qui possèdent la même première (resp. seconde) projection. Soit N tel que $|V_{\mathcal{G}}| > N$ et $|V_{\mathcal{H}}| > N$. Considérons un jeu du voleur et des policiers à N policiers. À tout instant du jeu, une ligne et une colonne au moins ne contiennent aucun policier. C'est à l'intersection de cette ligne et de cette colonne que se place le voleur. On vérifie aisément qu'à chaque étape du jeu, le voleur peut se déplacer jusqu'à ce sommet sans rencontrer

de policier. Il gagne. □

Corollaire 6.28 *Si le produit asynchrone de deux graphes est de largeur arborescente bornée alors l'un au moins est une union disjointe de composantes connexes de tailles bornées.*

Lemme 6.29 *Les graphes préfixes reconnaissables sont clos par produit asynchrone avec les graphes possédant des composantes connexes de taille bornée.*

Preuve. Standard. □

Lemme 6.30 *Pour tout VRA-terme régulier normalisé t dont aucune branche ne contient une infinité d'opérateurs \square et tel que $\llbracket t \rrbracket$ est de largeur arborescente bornée, il existe un VR-terme régulier t' tel que $\llbracket t \rrbracket \approx \llbracket t' \rrbracket$.*

Preuve. Pour t un terme, notons $S(t)$ l'ensemble des sous-termes de t possédant un opérateur \square à la racine.

On procède alors par récurrence sur $|S(t)|$.

Si $|S(t)| = 0$ alors t est un VR-terme régulier.

Si $|S(t)| > 0$ alors soit t' un sous-terme de t de la forme $t' = t'_1 \square_\eta t'_2$. On a pour $i \in [2]$, $S(t'_i) \subseteq S(t)$ car t'_i est un sous-terme de t . On a de plus $t' \notin S(t'_i)$ car $t' \notin S(t')$ par hypothèse sur les branches infinies.

Par hypothèse de récurrence, $\llbracket t'_1 \rrbracket$ et $\llbracket t'_2 \rrbracket$ sont préfixes reconnaissables. En appliquant le corollaire 6.28 et le lemme 6.29, on en déduit que $\llbracket t' \rrbracket$ est préfixe reconnaissable. Il existe donc un VR-terme t'' tel que $\llbracket t' \rrbracket \approx \llbracket t'' \rrbracket$. On remplace alors dans t toutes les occurrences du terme t' par t'' , obtenant ainsi le terme t''' . On a alors $\llbracket t''' \rrbracket \approx \llbracket t \rrbracket$. De plus, $S(t''') \subseteq S(t) - \{t'\}$, donc, par hypothèse de récurrence, $\llbracket t''' \rrbracket$ est préfixe reconnaissable. □

Conclusions

L'application successive des étapes 1 et 2 nous amène alors au lemme ci-dessous.

Lemme 6.31 *Les graphes solutions des VRA-systèmes d'équations finis et de largeur arborescente bornée sont VR-équationnels.*

Il ne reste plus alors qu'à appliquer le résultat de Barthelmann (théorème 25) pour conclure que les graphes solutions de VRA-systèmes finis d'équations de largeur arborescente bornée sont HR-équationnels. Réciproquement, les graphes

HR-équationnels sont de largeur arborescente bornée (Courcelle, propriété 4.12). Ceci achève la preuve du théorème 34.

Remarquons que cette preuve est fondamentalement différente de celle de Barthelmann. En effet, l'hypothèse de finitude du système d'équations est utilisée à plusieurs reprises, et donc cette transformation ne peut pas être effectuée (*a priori*) par un transducteur déterministe. De l'avis de l'auteur, il est fort improbable qu'une telle généralisation soit vraie. Néanmoins, il est possible que la preuve présentée ici puisse être généralisée aux VRA-systèmes d'équations appartenant à la hiérarchie de Caucal: le HR-système d'équations résultant étant éventuellement à un autre niveau de cette hiérarchie.

6.5.3 Décidabilité et effectivité

Il s'agit maintenant de montrer que la transformation présentée précédemment permet à la fois de décider si la solution d'un VRA-système fini d'équations est de largeur arborescente bornée, et, le cas échéant, de construire effectivement un HR-système d'équations fini équivalent. Pour établir cela, il s'agit d'observer plus attentivement la preuve précédente, étape par étape.

Étape 1 : pour que l'étape 1 puisse être effectuée, il suffit de garantir qu'aucune branche ne contient une infinité de produits avec des graphes possédant des arcs qui ne sont pas des boucles. Les graphes ne possédant que des boucles sont VRA-reconnaissables (car cette propriété est exprimable par une formule au premier ordre). L'existence d'une branche contenant une infinité de produits avec des graphes possédant un arc qui n'est pas une boucle est alors reconnaissable. Pour pouvoir effectuer la transformation du lemme 6.25, il reste à montrer que l'on peut savoir pour tout ensemble d'étiquettes $E \subseteq A$ le nombre de sommets d'un graphe possédant exactement E comme étiquettes de ses boucles. Pour cela, on commence par décider pour un ensemble E s'il concerne une infinité de sommets, ce qui est décidable d'après le corollaire 6.17. Sinon, on compte ces sommets. Le processus se termine car les sommets sont en nombre fini.

Étape 2 : pour assurer la validité de l'étape 2, il suffit de pouvoir décider si un graphe préfixe reconnaissable a ses composantes connexes de taille bornée. On sait [Cau96] que les composantes connexes d'un graphe préfixe reconnaissable sont en nombre fini à isomorphisme près. Il suffit donc de décider si les composantes du graphe sont finies. C'est une propriété décidable. Pour reconstruire le graphe correspondant, on utilise un argument similaire au cas précédent.

Enfin, Barthelmann [Bar98] nous enseigne que le problème de savoir si un graphe VR-équationnel est de largeur arborescente bornée est décidable, et la transformation en un HR-système fini d'équations est effective.

Il est alors possible de conclure.

Propriété 6.32 *Le problème de savoir si la solution d'un VRA-système d'équa-*

tions fini est de largeur arborescente bornée est décidable, et, si c'est le cas, il est effectivement possible de produire un HR-système fini d'équations de solution isomorphe.

Notons que ce résultat n'est possible que grâce à un renforcement des hypothèses par rapport au résultat de Barthelmann (théorème 25). En effet, borner les sous-graphes bipartis complets ou restreindre aux graphes uniformément creux n'est ici pas suffisant (considérer la grille infinie par exemple).

Il est aussi intéressant de comparer ce résultat de décidabilité avec le cas des graphes de réécriture suffixe de termes restreints par accessibilité : «*décider si un graphe de réécriture suffixe avec restriction par accessibilité est de largeur arborescente bornée est une question ouverte (cf [LÖ2b]).*» Ce problème de décidabilité s'avère beaucoup plus délicat que celui qui vient d'être présenté.

6.6 Largeur de clique

Il s'agit dans ce paragraphe d'étudier une extension possible du théorème 34 : les graphes solutions de VRA-systèmes finis d'équations de «largeur de clique bornée» sont-ils solution de VR-systèmes finis d'équations ? Nous donnons à cette question une réponse négative, et pour cela nous construisons un contre-exemple.

Nous présentons d'abord la problématique et les travaux connexes (paragraphe 6.6.1). Le paragraphe 6.6.2 présente comment construire un arbre possédant une théorie monadique indécidable servant ultérieurement. Le paragraphe 6.6.3 présente un opérateur de retrait et montre comment il est relié aux opérateurs VRA. La construction proprement dite est alors abordée au paragraphe 6.6.4.

6.6.1 Contexte

La largeur arborescente correspond aux solutions des HR-systèmes d'équations : un graphe est de largeur arborescente bornée si et seulement s'il est valeur d'un HR[C, A]-terme où C est un ensemble fini de couleurs.

Ainsi, le théorème de Barthelmann 34 peut-il s'énoncer : «si un graphe est à la fois valeur d'un VRA-terme régulier et valeur d'un HR-terme, alors il est valeur d'un HR-terme régulier».

À la lumière de cette équivalence, il est naturel de considérer une mesure des graphes correspondant aux opérateurs VR. Courcelle la nomme largeur de clique [Cou00] (Wanke et Blumensath définissent des mesures comparables). On dit qu'un graphe est de *largeur de clique bornée* si il existe un VR-terme dont il est solution. Une autre caractérisation des graphes de largeur de clique bornée nous intéresse.

Propriété 6.33 *Un graphe est largeur de clique bornée si et seulement si il*

s'obtient par interprétation monadique d'un arbre.

Ce résultat classique suit des résultats du chapitre 4.

La question naturelle est alors la suivante : les graphes valeur des VRA-termes réguliers et de largeur de clique bornée sont-ils les graphes VR-équationnels ?

Cette question est ouverte dans la thèse de Löding [LÖ2b]. Löding a montré que les graphes de réécriture suffixe restreint par accessibilité de largeur de clique bornée étaient HR-équationnels [LÖ2a]. Cette preuve consiste juste à remarquer que dans ce cas, les graphes ayant des sous-graphes bipartis complets bornés, la largeur arborescente bornée et la largeur de clique bornée sont deux notions équivalentes. La véritable question se pose en fait quand les graphes peuvent être de degré infini et contenir «de grands» graphes bipartis complets.

Nous montrons ici que cette conjecture est fautive. Ce résultat est issu d'une collaboration entre Löding et l'auteur.

Le contre-exemple que nous allons construire (cf. théorème 35) ne possède pas une théorie monadique décidable.

6.6.2 Un arbre de théorie monadique indécidable

Dans ce paragraphe, nous expliquons brièvement comment construire un arbre dont la théorie monadique est indécidable : le dépliage de la grille infinie avec retour.

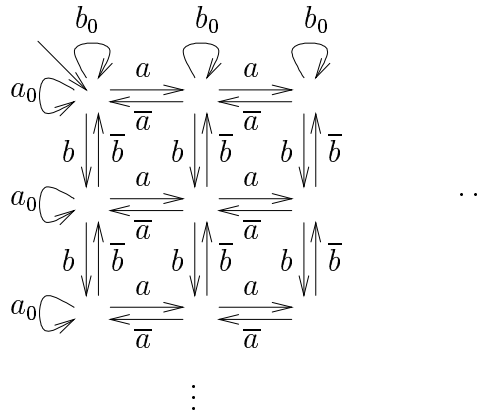


FIG. 6.4 – Grille infinie avec retour et tests à zéro.

Pour construire un arbre dont la théorie monadique est indécidable, on peut par exemple chercher à réduire la décidabilité de la logique monadique à un problème d'arrêt de machine de Minsky [Min67]. Une machine de Minsky est une machine à états finis disposant de deux compteurs (à valeur dans les entiers naturels), et pouvant les incrémenter, les décrémenter et tester leur éventuelle

égalité à 0. L'arrêt d'une telle machine à partir d'une configuration donnée est un problème indécidable.

Le domaine de calcul d'une telle machine se représente comme une grille infinie avec retour et test à zéro. La figure 6.4 présente une telle grille K . Les étiquettes a, \bar{a} permettent d'incrémenter et de décrémenter le premier compteur, l'étiquette a_0 de tester son égalité à 0. Les étiquettes b, \bar{b} et b_0 jouent le même rôle pour le second compteur. Le problème de l'arrêt d'une machine de Minsky se ramène directement au problème suivant : étant donné un langage rationnel L sur l'alphabet $\{a, \bar{a}, a_0, b, \bar{b}, b_0\}$, existe-t-il un chemin dans K d'origine $(0, 0)$ étiqueté par un mot de L ?

Ce problème ne change en fait pas quand on déplie K depuis le sommet $(0, 0)$. L'accessibilité rationnelle étant exprimable en logique monadique, on en déduit que la théorie monadique du dépliage depuis $(0, 0)$ de K est indécidable.

En fait, il n'est pas nécessaire pour obtenir ce résultat de disposer des arcs étiquetés par a_0 et b_0 . En effet, les sommets de $Depliage(K)$ correspondant à une valeur du premier (resp. second) compteur égale à zéro sont exactement ceux qui ne sont origine d'aucun arc étiqueté par \bar{a} (resp. par \bar{b}). Ceci est exprimable en logique monadique. Ainsi, le dépliage de la grille infinie avec retour sans test à zéro a une théorie monadique indécidable.

6.6.3 Opérateur de retrait

Nous voyons dans ce paragraphe, vis à vis des propriétés qui nous intéressent (c'est-à-dire la décidabilité de la théorie monadique et la largeur de clique), comment ajouter des arcs «équivalent» à en enlever (corollaire 6.35).

On introduit pour cela un nouvel opérateur $\text{retrait}[c_1, a, c_2]$ où c_1 et c_2 sont des couleurs et a est une étiquette. Cet opérateur appliqué à un graphe coloré retire tous les arcs étiquetés par la lettre a reliant un sommet de couleur c_1 à un sommet de couleur c_2 . Formellement, on pose

$$\text{retrait}[c_1, a, c_2] = \left[\begin{array}{l} x \xrightarrow{a} y : x \xrightarrow{a} y \wedge \left(\bigvee_{c \neq c_1} c(x) \vee \bigvee_{c \neq c_2} c(y) \right) \\ x \xrightarrow{b} y : x \xrightarrow{b} y \\ c(x) : c(x) \end{array} \quad b \neq a \right].$$

Comme pour l'opérateur arc et pour $D \subseteq C \times \mathbf{A} \times C$, $\text{retrait}[D]$ est un raccourci pour $\text{retrait}[c_1, a_1, c'_1] \circ \dots \circ \text{retrait}[c_n, a_n, c'_n]$ où $D = \{(c_1, a_1, c'_1), \dots, (c_n, a_n, c'_n)\}$.

Propriété 6.34 *Pour $D \subseteq C \times A \times C$, on a*

- $\text{retrait}[D] = \text{retrait}[D] \circ \text{arc}[D]$ et
- $\text{arc}[D] = \text{arc}[D] \circ \text{retrait}[D]$.

Preuve. Évident. □

Corollaire 6.35 *Pour tout $D \subseteq C \times A \times C$ et tout graphe coloré \mathcal{G} ,*

- *retrait $[D](\mathcal{G})$ est de largeur de clique finie si et seulement si $\text{arc}[D](\mathcal{G})$ est de largeur de clique finie,*
- *retrait $[D](\mathcal{G})$ a une théorie monadique décidable si et seulement si $\text{arc}[D](\mathcal{G})$ a une théorie monadique décidable.*

Preuve. Si $\text{retrait}[D](\mathcal{G})$ est de largeur de clique finie, alors $\text{arc}[D](\text{retrait}[D](\mathcal{G})) = \text{arc}[D](\mathcal{G})$ l'est aussi car $\text{arc}[D]$ est une interprétation monadique. Symétriquement, si $\text{arc}[D]$ est de largeur de clique finie, alors $\text{retrait}[D](\mathcal{G})$ l'est aussi. Le même principe s'applique pour la décidabilité de la théorie monadique. □

6.6.4 La construction

Le principe de la preuve est de construire un graphe de la forme

$$\mathcal{H}' = \text{retrait}[D](\mathcal{G}_1 \square_{\eta} \mathcal{G}_2 \square_{\eta'} \mathcal{G}_3)$$

où \mathcal{G}_1 , \mathcal{G}_2 et \mathcal{G}_3 sont des graphes préfixes et \mathcal{H}' est un arbre de théorie monadique indécidable.

Bien entendu, ce graphe n'est pas solution d'un VRA-système fini d'équations (car il serait de largeur arborescente bornée, donc HR-équationnel, donc de théorie monadique décidable). En revanche, $\text{arc}[D](\mathcal{G}_1 \square_{\eta} \mathcal{G}_2 \square_{\eta'} \mathcal{G}_3)$ est un graphe solution d'un VRA-système fini d'équations qui d'après le corollaire 6.35 est de largeur de clique finie et de théorie monadique indécidable.

Il nous reste donc à construire \mathcal{G}_1 , \mathcal{G}_2 , \mathcal{G}_3 et D . On pose $A = \{a, \bar{a}, b, \bar{b}, \bullet\}$. Les graphes sont présentés graphiquement à la figure 6.5. Les graphes \mathcal{G}_1 , \mathcal{G}_2 et \mathcal{G}_3 correspondent respectivement à la demi-droite des entiers avec retour étiquetée par a et \bar{a} , à la demi-droite des entiers avec retour étiquetée par b et \bar{b} et à l'arbre quaternaire complet étiqueté par a , \bar{a} , b et \bar{b} . Chacun des arcs a été scindé en deux parties en insérant un sommet intermédiaire. Cette opération préserve les graphes préfixes et les graphes HR-équationnels mais pas les graphes VR-équationnels. Dans le cas présent, \mathcal{G}_1 , \mathcal{G}_2 et \mathcal{G}_3 sont des graphes préfixes. Plus précisément, l'arc nouvellement ajouté est étiqueté par \bullet et le nouveau sommet se voit attribuer une couleur qui permet d'identifier l'étiquette de l'arc de la scission duquel il provient.

Il s'agit de considérer le produit $\mathcal{H} = \mathcal{G}_1 \square_{\eta} \mathcal{G}_2 \square_{\eta'} \mathcal{G}_3$. Les fonctions de coloriage η et η' sont choisies de manière à ce que la couleur d'un sommet soit le triplet des couleurs qu'ont ses trois composantes dans \mathcal{G}_1 , \mathcal{G}_2 et \mathcal{G}_3 . Ainsi, le graphe \mathcal{H} est-il étiqueté par $C = [3] \times [3] \times [5]$.

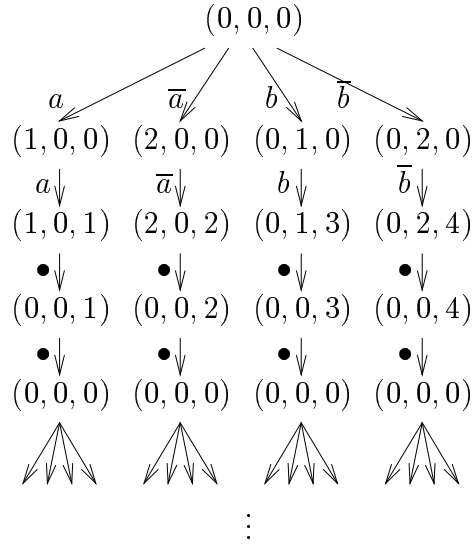


FIG. 6.6 – Construction du théorème 35.

$D = (C \times A \times C) - D_c$. On considère le graphe $\mathcal{H}' = \text{retrait}[D](\mathcal{H})$.

Lemme 6.36 *Tous les sommets de \mathcal{H}' de couleur appartenant à*

$$C' = C - \{(0, 0, 0), (1, 0, 0), (1, 0, 1), (0, 0, 1), \\ (2, 0, 0), (2, 0, 2), (0, 0, 2), \\ (0, 1, 0), (0, 1, 3), (0, 0, 3), \\ (0, 2, 0), (0, 2, 4), (0, 0, 4)\},$$

sont isolés (c.a.d. origine et destination d'aucun arc).

Preuve. Ces couleurs n'apparaissent pas dans D_c . □

Lemme 6.37 *Tous les sommets de \mathcal{H}' de couleur dans C' sont destination d'au plus un arc.*

Preuve. Au cas par cas. □

Corollaire 6.38 *Le graphe \mathcal{H}' est de largeur arborescente bornée (et donc de largeur de clique).*

Preuve. On peut montrer que tout graphe fini connexe tel que tout sommet est destination d'au plus un arc est de l'une des deux formes suivantes :

- il s'agit d'un arbre,
- il s'agit d'un graphe tel que, si on lui enlève un arc approprié, il devient un arbre.

Ces deux types de graphes ont une largeur arborescente inférieure à 2. L'union disjointe de tels graphes a donc aussi une largeur arborescente inférieure à 2. \square

Remarque 6.39 *En fait, \mathcal{H}' est une forêt d'arbres (de largeur arborescente 1) bien qu'il ne soit pas nécessaire de le montrer pour conclure.*

On ne s'intéresse plus maintenant qu'aux chemins de longueur 4 reliant ses sommets de couleur $(0, 0, 0)$. On suppose que l'ensemble des sommets de couleur 0 de \mathcal{G}_1 (resp. \mathcal{G}_2) est \mathbb{N} , et que ce codage est naturel, *c.a.d.* que le premier sommet est numéroté 0 et que l'on passe de n à $n+1$ en empruntant un arc a (resp. b) puis un arc \bullet . De même, on identifie les sommets de \mathcal{G}_3 de couleur 0 avec $\{a, b, \bar{a}, \bar{b}\}^*$ de manière naturelle.

Un sommet de \mathcal{H}' de couleur $(0, 0, 0)$ est alors un triplet (n_1, n_2, u) où $n_1, n_2 \in \mathbb{N}$ et $u \in \{a, b, \bar{a}, \bar{b}\}^*$.

Lemme 6.40 *Tout sommet (n_1, n_2, u) de couleur 0 de \mathcal{H}' est origine d'un unique chemin étiqueté par $aa \bullet \bullet$ (resp. $bb \bullet \bullet$) et la destination de ce chemin est $(n_1 + 1, n_2, ua)$ (resp. $(n_1, n_2 + 1, ub)$).*

Tout sommet (n_1, n_2, u) de couleur 0 de \mathcal{H}' est origine d'un arc étiqueté par \bar{a} (resp. \bar{b}) si et seulement si $n_1 > 0$ (resp. $n_2 > 0$) et dans ce cas la destination de chemin est $(n_1 - 1, n_2, u\bar{a})$ (resp. $(n_1, n_2 - 1, u\bar{b})$).

Preuve. Au cas par cas, arcs par arcs. \square

Lemme 6.41 *Il existe une interprétation monadique \mathcal{I} telle que $\mathcal{I}(\mathcal{H}')$ est le dépliage de la grille avec retour.*

Preuve. Appliquer d'abord la substitution finie inverse $aa \bullet \bullet \mapsto a$, $bb \bullet \bullet \mapsto b$, $\bar{a}\bar{a} \bullet \bullet \mapsto \bar{a}$, $\bar{b}\bar{b} \bullet \bullet \mapsto \bar{b}$ puis restreindre par accessibilité à partir du sommet $(0, 0, \varepsilon)$ (il s'identifie de manière unique comme le seul sommet de couleur $(0, 0, 0)$ destination d'aucun arc et origine d'aucun arc étiqueté par une lettre barrée).

La validité de cette construction repose bien sûr sur le lemme 6.40 \square

Corollaire 6.42 *Le graphe \mathcal{H}' a une théorie monadique indécidable.*

Il ne reste plus qu'à conclure.

Théorème 35 *Il existe un graphe \mathcal{G} solution d'un VRA-système fini d'équations tel que*

- \mathcal{G} est de largeur de clique bornée,*
- \mathcal{G} a une théorie monadique indécidable.*

Preuve. Le graphe \mathcal{H}' est de largeur de clique bornée et a une théorie monadique indécidable. D'après le corollaire 6.35, il en est de même pour $\text{arc}[D](\mathcal{G}_1 \square \mathcal{G}_2 \square \mathcal{G}_3)$ qui est solution d'un VRA-système fini d'équations. \square

Chapitre 7

Conclusion et perspectives

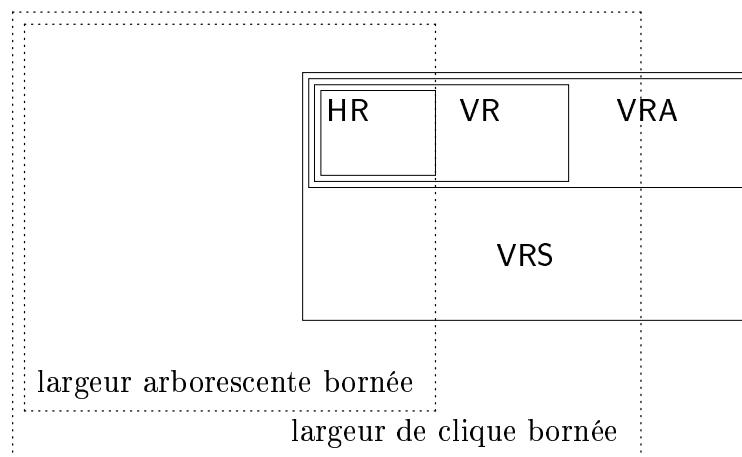


FIG. 7.1 – Familles de graphes solutions de systèmes d'équations finis.

Notre étude a concerné les structures infinies admettant une description finie.

Nous avons montré que l'approche équationnelle, qui est classiquement utilisée pour les opérateurs **HR** ou **VR**, pouvait être étendue à d'autres opérateurs. En particulier, l'ajout d'opérateurs de produit (synchrone ou asynchrone) permet de décrire plusieurs classes de familles pour lesquelles aucune représentation de ce type n'était jusqu'alors connue. Nous nous sommes également intéressés dans un cas à étudier l'ajout d'un opérateur supplémentaire de fusion.

Nous avons étudié les différentes relations qu'entretiennent ces familles de structures. La figure 7.1 récapitule une partie de ces résultats. Chaque zone Z symbolise l'ensemble des graphes solutions de Z -systèmes finis d'équations. Ces inclusions sont aussi valables pour les systèmes infinis sauf en ce qui concerne le vide de l'intersection de la zone **VRA** et de la largeur arborescente bornée (cf. conclusions du paragraphe 6.5.2).

Nous nous sommes aussi attachés à établir les différentes représentations possibles pour ces structures. Celles-ci peuvent être classées en trois catégories : les représentations équationnelles, les représentations internes et les représentations par transformations. Les familles de structures les plus «robustes» admettent ces trois représentations (il s’agit des structures préfixes reconnaissables et des structures à présentation terme-automatique).

Enfin, nous avons montré comment certains résultats se généralisent naturellement aux systèmes infinis d’équations. En particulier, la preuve de Barthelmann reliant les graphes préfixes reconnaissables aux graphes HR-équationnels reste valide dans le contexte des systèmes infinis d’équations. Il en est de même pour la décidabilité du premier ordre (avec modalité de comptage) pour les structures admettant une présentation terme-automatique, ainsi que pour la décidabilité du premier ordre avec accessibilité pour les graphes de réécriture suffixe.

Pour mener à bien ces démonstrations, nous avons dû développer les transducteurs descendants déterministes avec anticipation. Nous pensons que l’étude des structures infinies illustre pleinement l’utilité de cet outil.

Problèmes ouverts

Pour finir, voici une liste de problèmes directement reliés au travail présenté et qui sont pour l’instant sans réponses.

Aussi étonnant que cela puisse paraître, on ne connaît pas les inclusions respectives de toutes les familles de graphes présentées jusqu’à présent. En particulier, la question ci-dessous est ouverte.

Question 7.1 (C. Löding [L02b]) *Les graphes VRA-équationnels admettent-ils une présentation automatique ?*

Löding conjecture que la réponse à cette question est négative. Un graphe solution d’un VRA-système fini d’équations candidat à ne pas admettre une présentation automatique se décrit par réécriture suffixe de termes comme suit. L’ensemble de sommets est l’ensemble des termes sur l’alphabet $\{c:0, p:2\}$ et l’unique règle de réécriture remplace le sous-terme c par $p(c, c)$. Ce graphe est décomposé en couches (correspondant au nombre de symboles c contenus dans le terme), et tout arc issu d’un sommet de la couche de niveau n a pour destination un sommet de la couche de niveau $(n+1)$. Le nombre de sommets de la couche de niveau n est $\frac{1}{n+1} \binom{2n}{n}$. Ce nombre est connu sous le nom de nombre de Catalan, et pour cela, on appelle ce graphe *graphe de Catalan*.

Plusieurs questions concernent l’opérateur fusion. La première correspond à la remarque 4.19 page 110.

Question 7.2 *La famille des structures solutions de $VR' + \text{fusion-systèmes finis}$*

d'équations est-elle égale à la famille des structures solutions de VR'-systèmes finis d'équations si l'opérateur fusion peut prendre comme paramètre n'importe quel symbole unaire (contrairement au théorème 16 page 115 dans lequel seules les fusions de couleurs sont autorisées)? Si la réponse est positive, la transformation du système d'équations peut-elle être effectuée par un transducteur déterministe?

L'impact de l'opérateur de fusion reste à étudier pour les autres systèmes d'équations.

Question 7.3 *Les structures solutions de VRS'-systèmes finis d'équations utilisant des opérateurs fusion supplémentaires coïncident-elles avec les structures solutions de VRS'-systèmes finis d'équations? Si la réponse est positive, la transformation des systèmes d'équations peut-elle être effectuée par un transducteur déterministe ?*

À ces deux questions, il est probable que la réponse est positive et qu'il ne s'agit que d'une adaptation de la preuve du théorème 16 traitant le cas du produit synchrone. Notons toutefois qu'il est nécessaire, pour donner un sens à cette question, de modifier les opérateurs VRS' afin qu'ils puissent gérer les couleurs.

Enfin, une dernière question concerne les graphes de réécriture suffixe.

Question 7.4 *Quelle est la nature de la famille de graphes solutions de VRA-systèmes finis d'équations augmentés de l'opérateur fusion ? Est-elle égale à la famille des graphes solutions des VRA-systèmes finis d'équations ?*

La réponse à cette question est, d'après l'auteur, négative. Un candidat contre-exemple est le graphe solution en x des équations sur les couleurs $\{0, 1\}$ et l'unique étiquette a ci-dessous.

$$\begin{aligned} x &= \text{fusion}[0](\text{sommet}[0] \oplus (x \square_{\eta} A)) \\ A &= \text{arc}[0, a, 1](\text{sommet}[0] \oplus \text{sommet}[1]) \\ \eta(c, c') &= \begin{cases} 0 & \text{si } c = c' = 0 \\ 1 & \text{sinon} \end{cases} \end{aligned}$$

Il s'agit d'un hypercube de dimension dénombrable (dont tous les sommets sont de couleur 1 sauf un de couleur 0).

Il est toutefois simple de montrer que la famille des graphes solutions des VRA-systèmes finis d'équations augmentés d'un opérateur de fusion a une théorie au premier ordre avec accessibilité décidable. Ceci rend cette famille attractive.

Bibliographie

- [AN01] A. Arnold and D. Niwiński. *Rudiments of μ -Calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2001.
- [Bar97] K. Barthelmann. On equational simple graphs. Technical Report 9, Universität Mainz, Institut für Informatik, 1997.
- [Bar98] K. Barthelmann. When can equational simple graphs be generated by hyperedge replacement? In L. Brim, J. Gruska, and J. Zlatuska, editors, *MFCS'98*, LNCS, pages 543–552. Springer Verlag, 1998.
- [Büc60] J. Büchi. On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University press, 1960.
- [Büc64] J. Büchi. Regular canonical systems. In *Archiv für Mathematische Grundlagenforschung*, volume 6, pages 91–111. , 1964.
- [Bek67] H. Bekič. Definable operations in general algebras, and the theory of automata and flowcharts. Technical report, IBM Laboratory, Wien, 1967.
- [Blu99] A. Blumensath. Automatic structures. Diploma thesis, RWTH-Aachen, 1999.
- [Blu01] A. Blumensath. Prefix-recognisable graphs and monadic second-order logic. Technical Report AIB-06-2001, RWTH Aachen, May 2001.
- [Cac03] T. Cachat. Higher order pushdown automata, the Caucal hierarchy of graphs and parity games. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming*, volume 2719 of *LNCS*, pages 556–569. Springer-Verlag, 2003.
- [Cau90] D. Caucal. On the regular structure of prefix rewriting. In *Caap 90*, volume 431 of *LNCS*, pages 87–102, 1990. Paru dans TCS 106, pp. 61–86, 1992.
- [Cau96] D. Caucal. On infinite transition graphs having a decidable monadic theory. In *ICALP'96*, volume 1099 of *LNCS*, pages 194–205, 1996.
- [Cau98] D. Caucal. *Sur des graphes infinis réguliers*. Habilitation à diriger les recherches, Université de Rennes I, 1998.

- [Cau02] D. Caucal. On infinite terms having a decidable monadic theory. In *MFCS'02*, pages 79–115. Springer Verlag, 2002.
- [CC03] A. Carayol and T. Colcombet. On equivalent representations of infinite structures. In *ICALP'03*, LNCS. Springer Verlag, July 2003.
- [CDG⁺97] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 1997. release October, 1st 2002.
- [CF00] T. Colcombet and P. Fradet. Enforcing trace properties by program transformation. In *27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, January 2000.
- [CK02] B. Courcelle and T. Knapik. The evaluation of first-order substitution is monadic second-order compatible. *Theoretical Computer Science*, 281:177–206, 2002.
- [CL04] T. Colcombet and C. Löding. On the expressive power of deterministic transducers over infinite trees. In *STACS'04*, LNCS. Springer Verlag, 2004.
- [CM02] B. Courcelle and J.A Makowsky. Fusion in relational structures and the verification of mso logic. In *MSCS*, volume 12, pages 203–235, 2002.
- [Col02a] T. Colcombet. On families of graphs having a decidable first order theory with reachability. In *ICALP'02*, volume 2380 of *LNCS*. Springer, July 2002.
- [Col02b] T. Colcombet. Rewriting in the partial algebra of typed terms modulo AC. *Electronic Notes in Theoretical Computer Science*, 68(6), 2002.
- [Cou89] B. Courcelle. The monadic second-order logic of graphs ii: infinite graphs of bounded tree width. *Math. Systems Theory*, 21:187–221, 1989.
- [Cou90] B. Courcelle. Graph rewriting: an algebraic and logic approach. *Handbook of TCS*, B:193–242, 1990.
- [Cou92] B. Courcelle. The monadic second-order logic of graphs iii: Tree-decompositions, minors and complexity issues. 26(3):257–286, 1992.
- [Cou94] B. Courcelle. Monadic second-order graph transductions: A survey. In *Theoretical Computer Science*, volume 126, pages 53–75. Elsevier, 1994.
- [Cou95] B. Courcelle. The monadic second order logic of graphs ix: Machines and their behaviours. In *Theoretical Computer Science*, volume 151, pages 125–162, 1995.
- [Cou97] B. Courcelle. On the expression of graph properties in some fragment of monadic second-order logic. In N. Immerman and P. Kolaitis,

- editors, *Descriptive Complexity and Finite Models*, volume 31 of *DMACS Series in Discrete Mathematics*, pages 33–62. AMS, 1997.
- [Cou00] B. Courcelle. Clique-Width of Countable Graphs: A Compactness Property. unpublished, 2000.
- [Cou03] B. Courcelle. The monadic second-order logic of graphs xiv: Uniformly sparse graphs and edge set quantifications. *Theoretical Computer Science*, 299(1-36), 2003.
- [CW98] B. Courcelle and I. Walukiewicz. Monadic second-order logic, graph coverings and unfoldings of transition systems. In *Annals of Pure and Applied Logic*, 1998.
- [CW03] A. Carayol and S. Wöhrle. The causal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In Springer-Verlag, editor, *FSTTCS'03, Int. Conf. on Foundations of Software Technology and Theoretical Computer Science*, LNCS, 2003.
- [Die00] R. Diestel. *Graph Theory, second edition*. Springer Verlag, 2000.
- [DT90] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *LICS'90*, pages 242–248. IEEE, 1990.
- [EGO00] C. Hirsch E. Grädel and M. Otto. Back and forth between guarded and modal logics. In *15th IEEE Symposium on Logic in Computer Science*, pages 217–228, 2000.
- [Eil74] S. Eilenberg. *Automata, Languages and Machines*, volume A. Academic Press, New York, 1974.
- [Eng77] J. Engelfriet. Top-down tree transducers with regular lookahead. *Mathematical System Theory*, 10:289–303, 1977.
- [Eng94] J. Engelfriet. Graph grammars and tree transducers. In S. Tison, editor, *CAAP'94*, volume 787 of *LNCS*, pages 15–36. Springer Verlag, 1994.
- [Eng99] J. Engelfriet. Derivation trees of ground term rewriting systems. *Information and Computation*, 152(1):1–15, 1999.
- [FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Computer and System Sciences*, 2(18):194–211, 1979.
- [FV59] S. Feferman and R. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicæ*, 47:57–103, 1959.
- [HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages and computation*. Addison-Wesley, 1979.
- [KN94] B. Khousainov and A. Nerode. Automatic presentations of structures. In D. Leivant, editor, *International Workshop in Logic and Computational Complexity*, volume 960 of *LNCS*, pages 367–392, 1994.
- [KNU02] T. Knapik, D. Niwinski, and P. Urzyczyn. Higher-order pushdown trees are easy. In M. Nielsen, editor, *FOSSACS'02*, LNCS, 2002.

- [Koz83] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [Lö2a] C. Löding. Ground tree rewriting graphs of bounded tree width. In *STACS'02*, volume 2285 of *LNCS*, pages 559–570. Springer Verlag, 2002.
- [Lö2b] C. Löding. *Infinite Graphs Generated by Tree Rewriting*. PhD thesis, RWTH Aachen, 2002.
- [Lö2c] C. Löding. Model-checking infinite systems generated by ground tree rewriting. In *FoSSaCS 2002*, volume 2303 of *LNCS*, pages 280–294. Springer Verlag, 2002.
- [Min67] M. Minsky. *Computation: Finite and Infinite Machines*. Prentice hall, 1967.
- [MS85] D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.
- [MS97] A. Mateescu and A. Salomaa. *Aspects of classical language theory*, volume 1 of *Handbook of formal languages*. G. Rozenberg, springer verlag edition, 1997.
- [MW67] J. Mezei and J. B. Wright. Algebraic automata and context-free sets. *Information and Control*, 11(2-3):3–29, 1967.
- [NT01] M. Napoli and S. La Torre. Automata-based representations for infinite graphs. *Theoretical Informatics and Applications*, 35(4):311–330, 2001.
- [Rab69] M. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. soc.*, 141:1–35, 1969.
- [RS83] N. Robertson and P. Seymour. Graph minors i, excluding a forest. *Journal of Combinatorial Theory, B*, 35:39–61, 1983.
- [RS86] N. Robertson and P. Seymour. Graph minors v, excluding a planar graph. *Journal of Combinatorial Theory, B*, 41:92–114, 1986.
- [RS90] N. Robertson and P. Seymour. Graph minors x, obstructions to tree-decomposition. *Journal of Combinatorial Theory, B*, 52:153–190, 1990.
- [Sem84] A. Semenov. Decidability of monadic theories. In *MFCS'84*, volume 176 of *LNCS*, pages 162–175. Springer-Verlag, 1984.
- [She75] S. Shelah. The monadic second order theory of order. *Annals of Mathematics*, 102:379–419, 1975.
- [Tar55] A. Tarski. A lattice-theoretical fixpoint theorem and its application. *Pacific Journal of Mathematics*, 5:285–309, 1955.
- [Tho88] R. Thomas. The tree-width compactness theorem for hypergraphs. manuscript, 1988.

- [Tho89] C. Thomassen. Configurations in graphs of large minimum degree, connectivity or chromatic number. In G. S. Bloom, editor, *Proceedings of 3rd International Conference on Combinatorial Mathematics*, volume 555, pages 402–411. New-York Academy of Science, 1989.
- [Tho90] W. Thomas. Automata on infinite objects. In J. Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier, 1990.
- [Tho97] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 389–455. Springer Verlag, 1997.
- [Urv02] T. Urvoy. On abstract families of graphs. In *DLT 02*, 2002.
- [Wal96] I. Walukiewicz. Monadic second order logic on tree-like structures. In *STACS'96*, volume 1046 of *LNCS*, pages 401–414, 1996.

Annexe A

Annexes

A.1 Complément sur les transducteurs déterministes

Arbre d'exécution et états inévitables

Afin de pouvoir énoncer le lemme de correction dans toute sa généralité, il faut pouvoir faire référence à l'exécution d'un transducteur lorsqu'il est appliqué à un terme. Cette exécution est représentée comme un terme, appelé arbre d'exécution, dont chaque symbole représente une transition du transducteur. La totalité du terme représente la totalité de l'exécution du transducteur sur un terme donné. L'arbre d'exécution permet alors de reconstruire par évaluation le résultat de ce transducteur lors de cette exécution.

Il convient tout d'abord de préciser cette notion d'arbre d'exécution. Considérons un transducteur déterministe $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$. Soit $T \subseteq \delta$ un ensemble de transitions, on définit l'alphabet gradué $\langle T \rangle$ suivant :

$$\langle T \rangle = \{ \langle \gamma \rangle \mid \gamma \in T \},$$

$$\text{avec pour tout } \gamma \in T, \quad | \langle \gamma \rangle | = \begin{cases} |f| & \text{si } \gamma = q(x) \rightarrow f(r_1(x), \dots, r_{|f|}(x)), \\ 1 & \text{sinon.} \end{cases}$$

Soit t^0 un \mathcal{F} -terme. On construit par récurrence sur l'entier n et pour tout

état $q \in Q$, le terme $exec_n^t[q](t^0) \in \mathcal{T}_\perp^\omega(\langle \delta \rangle)$ comme suit.

$$\begin{aligned}
& exec_n^t[q](t^0) \\
= & \begin{cases} \perp & \text{si } n = 0, \\
\langle \gamma \rangle (exec_{n-1}^t[r_1](t^0), \dots, exec_{n-1}^t[r_{|g|}](t^0)) & \text{si } \gamma = q(x) \rightarrow g(r_1(x), \dots, r_{|g|}(x)) \in \delta, \\
\langle \gamma \rangle (exec_{n-1}^t[r](t_i^0)) & \text{si } \gamma = q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i) \in \delta, \\
& \text{et } t^0 = f(t_1^0, \dots, t_{|f|}^0), \\
\langle \gamma \rangle (exec_{n-1}^t[r](t^0)) & \text{si } \gamma = q(x \in L) \rightarrow r(x) \in \delta \\
& \text{et } t^0 \in L \\
\text{non défini} & \text{sinon.} \end{cases}
\end{aligned}$$

Pour tout état $q \in Q$, le $\langle \delta \rangle$ -terme $exec^t[q](t^0)$ se définit alors par passage à la limite : $exec^t[q](t^0) = \sqcup_{n \in \mathbb{N}} exec_n^t[q](t^0)$. Cette définition a bien un sens (cf. définition de l'évaluation d'un transducteur). Dans la pratique, on omet de préciser le transducteur déterministe \mathbf{t} de la notation $exec^t$ quand il n'y a pas d'ambiguïté.

À partir d'un terme sur l'alphabet $\langle \delta \rangle$, on peut recomposer un \mathcal{F}' -terme en l'évaluant dans la $\langle \delta \rangle$ -algèbre \mathbf{A}^t de domaine $\mathcal{T}_\perp^\omega(\mathcal{F}')$ et dont l'application sémantique est définie comme suit :

$$\begin{aligned}
\text{sem}[\mathbf{A}^t](\langle q(x) \rightarrow g(q_1(x), \dots, q_{|g|}(x)) \rangle)(t_1, \dots, t_{|g|}) &= g(\text{sem}[\mathbf{A}^t](t_1), \dots, \text{sem}[\mathbf{A}^t](t_{|g|})) , \\
\text{sem}[\mathbf{A}^t](\langle q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i) \rangle)(t) &= \text{sem}[\mathbf{A}^t](t) , \\
\text{sem}[\mathbf{A}^t](\langle q(x \in L) \rightarrow r(x) \rangle)(t) &= \text{sem}[\mathbf{A}^t](t) .
\end{aligned}$$

Énonçons un résultat bien naturel.

Lemme A.1 *Soit \mathbf{t} un transducteur déterministe des \mathcal{F} -termes dans les \mathcal{F}' -termes, q un de ses états et t^0 un \mathcal{F} -terme, alors $\mathbf{t}_q(t^0)$ est défini si et seulement si $exec^t[q](t^0)$ est défini, et dans ce cas,*

$$\mathbf{t}_q(t^0) = \llbracket exec^t[q](t^0) \rrbracket_{\mathbf{A}^t} .$$

On peut aussi remarquer que la définition de $exec$ est très similaire à la définition de l'exécution d'un transducteur déterministe. Ce lien est explicité par le lemme A.2.

Lemme A.2 *Pour \mathbf{t} un transducteur déterministe et q un de ses états, il existe un transducteur déterministe \mathbf{t}^e des \mathcal{F} -termes dans les \mathcal{F}^t -termes tel que $\mathbf{t}^e = exec^t[q]$.*

Construction. Le transducteur \mathbf{t}^e s'obtient de \mathbf{t} en :

- remplaçant le symbole g produit par une transition γ par $\langle \gamma \rangle$,

- faisant suivre chaque transition de consommation ou d'anticipation γ par une transition produisant $\langle \gamma \rangle$ (cela nécessite d'insérer un état intermédiaire).

□

Soit $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ un transducteur déterministe et $T \subseteq \delta$ un ensemble de ses transitions. On dit que T est *inévitabile* (dans \mathbf{t}) si pour tout \mathcal{F} -terme t^0 et tout état $q \in Q$ tels que $\mathbf{t}_q(t^0)$ est défini, alors toutes les branches infinies de $exec[q](t^0)$ contiennent une infinité d'occurrences d'un symbole $\langle \gamma \rangle \in \langle T \rangle$. La définition s'étend aux états. Soit $Q' \subseteq Q$ un ensemble d'états, Q' est *inévitabile* (dans \mathbf{t}) si l'ensemble des transitions de δ dont l'état de tête est dans Q' est lui-même inévitable.

Remarque A.3 *Soit $T \subseteq \delta$ un ensemble de transitions tel que tout «cycle» de \mathbf{t} contienne une transition de T , alors T est inévitable.*

La réciproque est fautive, et ce en particulier car le transducteur déterministe peut en utilisant une anticipation s'assurer qu'un état sera atteint au bout d'un nombre (non borné mais) fini d'itérations. Considérons par exemple le transducteur déterministe sur l'alphabet $\{a : 1, b : 0\}$ qui n'est défini que sur les termes finis, et dans ce cas renvoie le terme lui même :

$$\begin{aligned} q(x \in (\mu y.a(y).b)[]) &\rightarrow q', \\ q'(a(x)) &\rightarrow a(q(x)), \\ q'(c) &\rightarrow c. \end{aligned}$$

Ce transducteur déterministe contient un cycle, et pourtant \emptyset est inévitable car toute exécution de ce transducteur est finie.

Propriété A.4 *Déterminer si un ensemble de transitions d'un transducteur déterministe est inévitable est décidable.*

Preuve. Soient $\mathbf{t} = (\mathcal{F}, \mathcal{F}', Q, q_{init}, \delta)$ un transducteur et $T \subseteq \delta$ un ensemble de transitions dont on cherche à déterminer l'inévitabilité. L'ensemble des $\langle \delta \rangle$ -termes possédant une branche ne contenant pas une infinité de symboles de $\langle T \rangle$ est reconnaissable. Son image inverse par \mathbf{t}^e (cf. lemme A.2) l'est aussi. Cet ensemble est vide si et seulement si T est inévitable, et le vide d'une partie reconnaissable est décidable. □

Preuve du lemme 3.59

Preuve. La preuve consiste à évaluer le terme d'exécution associé à t^0 en utilisant les applications h comme sémantique. Le cœur de la preuve établit la correc-

tion de cette approche. Formalisons tout d'abord ce que l'on entend par «évaluer» un terme d'exécution. On définit pour tout terme $t^0 \in \mathcal{T}^\omega(\mathcal{F})$, tout état q et tout terme fini $t^e \sqsubseteq \text{exec}[q](t^0)$ l'élément $\llbracket t^e \rrbracket_{q,t^0} \in \mathcal{R}[q, t^0]$ par récurrence comme suit

- si $t^e = \perp$, alors $\llbracket t^e \rrbracket_{q,t^0} = \perp_{q,t^0}$,
- si $t^e = \gamma(u_1^e, \dots, u_{|g|}^e)$ pour $\gamma = q(x) \rightarrow g(r_1(x), \dots, r_{|g|}(x)) \in \delta$ alors,

$$\llbracket t^e \rrbracket_{q,t^0} = h_{\gamma,t^0}(\llbracket u_1^e \rrbracket_{r_1,t^0}, \dots, \llbracket u_{|g|}^e \rrbracket_{r_{|g|},t^0}),$$

- si $t^e = \gamma(u^e)$ pour $\gamma = q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i)$, et $t^0 = f(t_1^0, \dots, t_{|f|}^0)$ alors,

$$\llbracket t^e \rrbracket_{q,t^0} = h_{\gamma,t^0}(\llbracket u^e \rrbracket_{r,t_i^0}),$$

- si $t^e = \gamma(u^e)$ pour $\gamma = q(x \in L) \rightarrow r(x)$,

$$\llbracket t^e \rrbracket_{q,t^0} = h_{\gamma,t^0}(\llbracket t^e \rrbracket_{r,t^0}).$$

La définition s'étend par passage à la limite à $\text{exec}[q](t^0)$. Ainsi, il s'agit de la définition standard dans une algèbre si ce n'est que les informations supplémentaires q et t^0 sont maintenues simultanément.

Le reste de la preuve consiste à montrer que $\llbracket \text{exec}[q](t^0) \rrbracket_{q,t^0} = (t^0, \mathbf{t}(t^0))$, *c.a.d.* d'une part $\rho(\llbracket \text{exec}[q](t^0) \rrbracket_{q,t^0}) = t^0$ et d'autre part $\rho'(\llbracket \text{exec}[q](t^0) \rrbracket_{q,t^0}) = \mathbf{t}(t^0)$.

Preuve de $\rho'(\llbracket \text{exec}[q](t^0) \rrbracket_{q,t^0}) = \mathbf{t}(t^0)$. D'après le lemme A.1 et par passage à la limite, il est suffisant d'établir que $\text{exec}[q](t^0)$ est limite d'une chaîne de termes finis $(t_n^e)_{n \in \mathbb{N}}$ telle que pour tout $n \in \mathbb{N}$ on a $\llbracket t_n^e \rrbracket_{\mathbf{A}\mathbf{t}} = \rho'(\llbracket t_n^e \rrbracket_{q,t^0})$. Pour cela, on construit t_n^e en coupant toutes les branches du terme $\text{exec}[q](t^0)$ au niveau de la n -ième occurrence d'un symbole de $\langle \delta_\perp \rangle$ ou δ_\perp est l'ensemble des transitions ayant leur état de tête dans Q_\perp . On remplace le sous-arbre coupé par \perp . D'après l'hypothèse 3 les termes ainsi construits sont bien finis. Ils ont aussi naturellement t^e pour limite. Les hypothèses 3, *a*, *b* et *c* permettent alors par récurrence d'établir que $\llbracket t_n^e \rrbracket_{\mathbf{A}\mathbf{t}} = \rho'(\llbracket t_n^e \rrbracket_{q,t^0})$.

Preuve de $\rho(\llbracket \text{exec}[q](t^0) \rrbracket_{q,t^0}) = t^0$. Remarquons tout d'abord que par définition, $\rho(\llbracket \text{exec}[q](t^0) \rrbracket_{q,t^0}) \sqsubseteq t^0$. Il s'agit de montrer l'autre inégalité.

Soit t^e un terme d'exécution. On note $\text{prof}_c(t^e)$ la hauteur de t^e après que l'on a remplacé tous les sous-arbres ayant une transition de consommation à la racine par \perp (avec par convention la hauteur de \perp égale à 0).

On montre par récurrence sur t la propriété suivante.

Hyp. réc. : pour tout terme $t^0 \in \mathcal{T}^\omega(\mathcal{F})$, tout terme fini $t \in \overleftarrow{t^0}$ et tout état q , il existe un terme $t^e \sqsubseteq \text{exec}[q](t^0)$ tel que $t \sqsubseteq \rho(\llbracket t^e \rrbracket_{q,t^0})$.

Remarquons tout d'abord qu'une fois cette hypothèse établie, par passage à la limite, on obtient $t^0 \sqsubseteq \rho(\llbracket exec[q](t^0) \rrbracket)$ et la preuve est terminée.

La récurrence s'effectue sur la hauteur de t et $prof_c(exec[q](t^0))$.

- Supposons $t = \perp$, alors $t^e = \perp$ convient.
- Supposons $t = f(t_1, \dots, t_{|f|})$ et $prof_c(exec[q](t^0)) = 0$. La deuxième hypothèse implique que le symbole à la racine de $exec[q](t^0)$ est une transition de consommation γ . Soit $t^0 = f(t_1^0, \dots, t_{|f|}^0)$. La transition γ consomme un symbole f car $exec[q](t^0)$ est défini et que f est le symbole à la racine de t^0 . Soit donc $\gamma = q(f(x_1, \dots, x_{|f|})) \rightarrow r(x_i)$. Par hypothèse de récurrence, il existe un terme fini $u^e \sqsubseteq exec[r](t_i^0)$ tel que $t_i \sqsubseteq \rho(\llbracket u^e \rrbracket_{r, t_i^0})$. On a alors,

$$\begin{aligned}
t &\sqsubseteq f(t_1^0, \dots, t_{i-1}^0, t_i, t_{i+1}^0, \dots, t_{|f|}^0) && (\text{car } \forall j. t_j \sqsubseteq t_j^0) \\
&\sqsubseteq f(t_1^0, \dots, t_{i-1}^0, \rho(\llbracket u^e \rrbracket_{r, t_i^0}), t_{i+1}^0, \dots, t_{|f|}^0) && (\text{hyp. rec.}) \\
&\sqsubseteq \rho(h_{\gamma, t^0}(\llbracket u^e \rrbracket_{r, t_i^0})) && (\text{hyp. b}) \\
&\sqsubseteq \rho(\llbracket \langle \gamma \rangle (u^e) \rrbracket_{q, t^0}) && (\text{def } \llbracket \cdot \rrbracket)
\end{aligned}$$

Donc $t^e = \langle \gamma \rangle (u^e)$ satisfait l'hypothèse de récurrence.

- Supposons $t \neq \perp$ et $prof_c(exec[q](t^0)) > 0$. Soit γ la transition à la racine de $exec[q](t^0)$. Cette transition ne peut pas être une transition de consommation par hypothèse et donc seuls deux cas sont possibles.

1. si $\gamma = q(x) \rightarrow g(r_1(x), \dots, r_{|g|}(x))$, alors pour tout $i \in [|g|]$, par hypothèse de récurrence, pour tout $i \in [|g|]$, il existe $t_i^e \sqsubseteq exec[r_i](t^0)$ tel que $t \sqsubseteq \rho(\llbracket t_i^e \rrbracket_{r_i, t^0})$. On a alors,

$$\begin{aligned}
t &\sqsubseteq \rho(\llbracket t_1^e \rrbracket_{r_1, t^0}, \dots, \llbracket t_{|g|}^e \rrbracket_{r_{|g|}, t^0}) \\
&\sqsubseteq \rho(h_{\gamma, t^0}(\llbracket t_1^e \rrbracket_{r_1, t^0}, \dots, \llbracket t_{|g|}^e \rrbracket_{r_{|g|}, t^0})) && (\text{hyp. a}) \\
&\sqsubseteq \rho(\llbracket \langle \gamma \rangle (t_1^e, \dots, t_{|g|}^e) \rrbracket_{q, t^0}) && (\text{def. } \llbracket \cdot \rrbracket)
\end{aligned}$$

Donc $t^e = \langle \gamma \rangle (t_1^e, \dots, t_{|g|}^e)$ satisfait l'hypothèse de récurrence.

2. si $\gamma = q(x \in L) \rightarrow r(x)$ avec $t^0 \in L$, alors par hypothèse de récurrence il existe $u^e \sqsubseteq exec[r](t^0)$ tel que $t \sqsubseteq \llbracket u^e \rrbracket_{r, t^0}$. On a alors,

$$\begin{aligned}
t &\sqsubseteq \rho(h_{\gamma, t^0}(\llbracket u^e \rrbracket_{r, t^0})) && (\text{hyp. c}) \\
&\sqsubseteq \rho(\llbracket \langle \gamma \rangle (u^e) \rrbracket_{q, t^0}) && (\text{def. } \llbracket \cdot \rrbracket)
\end{aligned}$$

Donc $t^e = \langle \gamma \rangle (u^e)$ satisfait l'hypothèse de récurrence.

□

A.2 Complément sur le théorème 16

Lemme A.5 (opérateur sommet) Soient $E \subseteq C$ et $c \in C$. Il existe une structure VR'-définissable $f^{\text{sommet}[c],E}$ telle que :

$$\mathcal{R}[E](\text{sommet}[c], f^{\text{sommet}[c]}) .$$

Preuve. Si $c \in E$, on pose $f^{\text{sommet}[c],E} = \mathcal{I}_0(\text{sommet}[c])$, sinon $f^{\text{sommet}[c],E} = \mathcal{I}_0(\perp)$. Dans tous les cas, \mathcal{S} est finie et donc VR'-définissable et satisfait de plus la relation $\mathcal{R}[E](\text{sommet}[c], f^{\text{sommet}[c],E})$. \square

Lemme A.6 (opérateur interprétation) Soient $E \subseteq C$ et \mathcal{I} une interprétation booléenne positive des Σ -structures dans les Σ -structures. Alors il existe une application VR'-définissable $f^{\mathcal{I},E}$ telle que pour toute Σ -structure \mathcal{S} et pour toute Σ' -structure \mathcal{S}' , on a continuellement

$$\text{si } \mathcal{R}[E](\mathcal{S}, \mathcal{S}') \text{ alors } \mathcal{R}[E](\mathcal{I}(\mathcal{S}), f^{\mathcal{I},E}(\mathcal{S}')) .$$

Preuve. On pose :

$$\mathcal{I}_0^{\mathcal{I}} = \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) \\ c(x) \end{array} : \begin{array}{l} \langle X, g, \phi^{\mathcal{I}} \rangle (\overline{X}) \\ c(x) \end{array} \right| .$$

Supposons $\mathcal{R}[E](\mathcal{S}, \mathcal{S}')$. On a alors :

$$\begin{aligned} \mathcal{I}_0(\mathcal{I}(\mathcal{S}))|_E &= \left(\left| \begin{array}{l} \langle X, g, \phi \rangle \overline{X} \\ c(x) \end{array} : \begin{array}{l} \overline{g} \wedge \phi^{\mathcal{I}} \\ c(x) \end{array} \right| (\mathcal{S}) \right) \Big|_E && \text{(lemme 4.21)} \\ &= \mathcal{I}_0^{\mathcal{I}}(\mathcal{I}_0(\mathcal{S}))|_E && \text{(def } \mathcal{I}_0^{\mathcal{I}}) \\ &= \mathcal{I}_0^{\mathcal{I}}(\mathcal{I}_0(\mathcal{S})|_E) && \text{(lemme 4.26)} \\ &\approx \mathcal{I}_0^{\mathcal{I}}(\mathcal{I}'_0(\mathcal{S}')) && \text{(car } \mathcal{R}[E](\mathcal{S}, \mathcal{S}')) \end{aligned}$$

Donc $f^{\mathcal{I},E}(x) = \mathcal{I}_0^{\mathcal{I}}(\mathcal{I}'_0(x))$ satisfait les conclusions du lemme. \square

Lemme A.7 (opérateur \oplus) Soit $E \subseteq C$. Il existe une application binaire VR'-définissable $f^{\oplus,E}$ telle que pour toutes Σ -structures \mathcal{S}_0 et \mathcal{S}_1 et pour toutes Σ' -structures \mathcal{S}'_0 et \mathcal{S}'_1 , on a continuellement

$$\text{si } \mathcal{R}[E](\mathcal{S}_0, \mathcal{S}'_0) \text{ et } \mathcal{R}[E](\mathcal{S}_1, \mathcal{S}'_1) , \quad \text{alors } \mathcal{R}[E](\mathcal{S}_0 \oplus \mathcal{S}_1, f^{\oplus,E}(\mathcal{S}'_0, \mathcal{S}'_1)) .$$

Preuve. On pose :

$$\mathcal{I}^\oplus = \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) : \bigvee_{(\psi_1, \psi_2) \in \phi_g^\oplus} \begin{array}{c} \langle FV(\psi_1), g|_{FV(\psi_1)}, \psi_1 \rangle (\overline{FV(\psi_1)}) \\ \wedge \\ \langle FV(\psi_2), g|_{FV(\psi_2)}, \psi_2 \rangle (\overline{FV(\psi_2)}) \end{array} \end{array} \right|$$

On pose alors $f^{\oplus, E}(\mathcal{S}'_1, \mathcal{S}'_2) = \mathcal{I}^\oplus(\mathcal{I}'_0(\mathcal{S}'_1) \oplus \mathcal{I}'_0(\mathcal{S}'_2))$. Cette application satisfait les conclusions du lemme. \square

Lemme A.8 (opérateur fusion) *Soient $E \subseteq C$ et $c_0 \in C$. Il existe une application VR'-définissable $f^{\text{fusion}[c_0], E}$ telle que pour toute Σ -structure \mathcal{S} contenant un sommet de couleur c_0 et pour toute Σ' -structure \mathcal{S}' , on a continuellement*

$$\text{si } \mathcal{R}[E - \{c_0\}](\mathcal{S}, \mathcal{S}') \text{ alors } \mathcal{R}[E](\text{fusion}[c_0](\mathcal{S}), f^{\text{fusion}[c_0], E}(\mathcal{S}')) .$$

Preuve. Supposons $c_0 \in E$ (le cas $c_0 \notin E$ est similaire).

$$\mathcal{I}^{\text{fusion}} = \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) : \begin{array}{l} \langle X - X_0, g|_{X-X_0}, \phi_{g, c_0} \rangle (\overline{X - X_0}) \\ \text{avec } X_0 = g^{-1}(c_0) \end{array} \\ c(x) : c(x) \end{array} \right| .$$

On pose de plus

$$\text{net} = \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) : \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) \text{ si } g^{-1}(c_0) = \emptyset \\ \text{faux si } g^{-1}(c_0) \neq \emptyset \end{array} \\ c(x) : c(x) \end{array} \right| ,$$

$$\text{et } eq = \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) : \langle X, g, \phi \rangle (\overline{X}) \\ c(x) : c(x) \\ x \sim y : x \sim y \vee (c_0(x) \wedge c_0(y)) \end{array} \right| .$$

L'interprétation *net* élimine les hyperarcs impliquant un sommet de couleur c_0 et l'interprétation *eq* effectue la fusion par c_0 sans pour autant modifier les autres hyperarcs (cela ne conserve pas en général les structures avec quotient). L'interprétation $\mathcal{I}^{\text{fusion}}$ ne dépend pas des hyperarcs impliquant des sommets de couleur c_0 ,

$$\text{donc } \mathcal{I}^{\text{fusion}} \circ \text{net} = \mathcal{I}^{\text{fusion}} , \quad (\text{A})$$

$$\text{et } \mathcal{I}^{\text{fusion}} \circ eq = eq \circ \mathcal{I}^{\text{fusion}} . \quad (\text{B})$$

On a aussi pour toute structure \mathcal{S} contenant un sommet de couleur c_0 les équivalences suivantes :

$$\text{net}(\mathcal{S})|_E \approx \mathcal{S}|_{E - \{c_0\}} \oplus \text{net}(\mathcal{S}|_{\{c_0\}}) , \quad (\text{C})$$

$$eq(\mathcal{S})|_E = eq(\mathcal{S}|_E) , \quad (\text{D})$$

$$\text{et } eq(\text{net}(\mathcal{S})|_{\{c_0\}}) \approx \text{sommet}[c_0] . \quad (\text{E})$$

Supposons $\mathcal{R}[E - \{c_0\}](\mathcal{S}, \mathcal{S}')$ où \mathcal{S} contient un sommet de couleur c_0 . D'après les lemmes A.7 et A.5, il existe une application VR'-définissable h telle que $\mathcal{R}[E - \{c_0\}](\mathcal{S} \oplus \text{sommet}, h(\mathcal{S}'))$ (F). On a alors :

$$\begin{aligned}
& \mathcal{I}_0(\text{fusion}[c_0](\mathcal{S}))|_E \\
&= \left| \begin{array}{l} \langle X, g, \phi \rangle (\overline{X}) \\ c(x) \\ x \sim y \end{array} : \begin{array}{l} \overline{g} \wedge \phi_{g, c_0} \\ c(x) \\ x \sim y \\ \vee c_0(x) \wedge c_0(y) \end{array} \right| (\mathcal{S})|_E \quad (\text{lemme 4.22}) \\
&= eq(\mathcal{I}^{\text{fusion}}(\mathcal{I}_0(\mathcal{S})))|_E \quad (\text{def } eq \text{ et } \mathcal{I}^{\text{fusion}}) \\
&= eq(\mathcal{I}^{\text{fusion}}(\mathcal{I}_0(\mathcal{S}))|_E) \quad (\text{D}) \\
&= eq(\mathcal{I}^{\text{fusion}}(\mathcal{I}_0(\mathcal{S})|_E)) \quad (\text{lemme 4.26}) \\
&= eq(\mathcal{I}^{\text{fusion}}(\text{net}(\mathcal{I}_0(\mathcal{S})|_E))) \quad (\text{A}) \\
&\approx eq(\mathcal{I}^{\text{fusion}}(\text{net}(\mathcal{I}_0(\mathcal{S})|_{E-\{c_0\}} \oplus \text{net}(\mathcal{I}_0(\mathcal{S})|_{\{c_0\}}))) \quad (\text{C}) \\
&\approx eq(\mathcal{I}^{\text{fusion}}(\mathcal{I}_0(\mathcal{S})|_{E-\{c_0\}} \oplus \text{net}(\mathcal{S}|_{\{c_0\}}))) \quad (\text{F}) \\
&\approx \mathcal{I}^{\text{fusion}}(eq(\mathcal{I}_0(\mathcal{S})|_{E-\{c_0\}} \oplus \text{net}(\mathcal{S}|_{\{c_0\}}))) \quad (\text{B}) \\
&\approx \mathcal{I}^{\text{fusion}}(\mathcal{I}_0(\mathcal{S})|_{E-\{c_0\}} \oplus eq(\text{net}(\mathcal{S}|_{\{c_0\}}))) \\
&\approx \mathcal{I}^{\text{fusion}}(\mathcal{I}'_0(h(\mathcal{S}')) \oplus \text{sommet}[c_0]) \quad (\text{E})
\end{aligned}$$

Donc $f^{\text{fusion}[c_0, E]}(x) = \mathcal{I}^{\text{fusion}}(\mathcal{I}'_0(h(x)) \oplus \text{sommet}[c_0])$ satisfait les conclusions du lemme. \square

Lemme A.9 (opérateur recol) Soient $E \subseteq C$ et α une application de C dans C . Il existe $E' \subseteq C$ tel que pour toute Σ -structure \mathcal{S} et pour toute Σ' -structure \mathcal{S}' , on a continuellement

$$si \ \mathcal{R}[E'](\mathcal{S}, \mathcal{S}') \ \text{alors} \ \mathcal{R}[E](\text{recol}[\alpha](\mathcal{S}), \text{recol}[\alpha](\mathcal{I}'_0(\mathcal{S}'))).$$

Preuve. Preuve similaire à celle du lemme A.6. \square

A.3 Complément sur le théorème 26

Preuve du lemme 4.37

Preuve. On pose $C' = C \times [k]$ et l'application β est définie par $\beta((c, n)) = c$ pour toute couleur $c \in C$ et tout entier $n \in [k]$. Notons σ l'application $\text{recol}[\beta]$. Le transducteur que l'on construit attribue les couleurs respectives $(c, 1), \dots, (c, n)$ aux sommets de couleur originale c si ils sont en nombre inférieur à k (le choix de leur numérotation étant arbitraire). Si ils sont en nombre supérieur à k , alors tous les sommets se voient attribuer (par convention) la couleur $(c, 1)$. Les transitions maintiennent ce codage.

Formellement, étant donné un graphe coloré \mathcal{G}^∞ , la relation $\mathcal{R}[\mathcal{G}^\infty](\mathcal{G}, \mathcal{H})$ pour \mathcal{G} un graphe coloré par C et \mathcal{H} un graphe coloré par C' est satisfaite si

- $\mathcal{G} \subseteq \mathcal{G}^\infty$,
- $\mathcal{G} = \sigma(\mathcal{H})$,
- pour tout $c \in C$, si $|\mathcal{G}^\infty|_c > k$ alors
 - pour tout $i \in \{2, \dots, k\}$, $|\mathcal{H}|_{(c,i)} = 0$,
- pour tout $c \in C$, si $|\mathcal{G}^\infty|_c \leq k$,
 - pour tout $i \in \{1, \dots, |\mathcal{G}^\infty|_c\}$, $|\mathcal{H}|_{(c,i)} \leq 1$,
 - pour tout $i \in \{|\mathcal{G}^\infty|_c + 1, \dots, k\}$, $|\mathcal{H}|_{(c,i)} = 0$.

Si pour un graphe \mathcal{G} , on construit un graphe \mathcal{H} tel que $\mathcal{R}[\mathcal{G}^\infty](\mathcal{G}, \mathcal{H})$, alors il satisfait (*) et $\mathcal{G} = \sigma(\mathcal{H})$ ce qui conclut ce lemme.

Les constructions sont alors les suivantes.

1. Pour toute couleur c ,

$$\mathcal{R}[\text{sommet}[c]](\text{sommet}[c], \text{sommet}[(c, 1)]) .$$

2. Supposons $\mathcal{R}[\mathcal{G}_1^\infty](\sigma(\mathcal{H}_1), \mathcal{H}_1)$ et $\mathcal{R}[\mathcal{G}_2^\infty](\sigma(\mathcal{H}_2), \mathcal{H}_2)$. Alors,

$$\sigma(\mathcal{H}_1) \oplus \sigma(\mathcal{H}_2) = \sigma(\text{recol}[\gamma_1](\mathcal{H}_1) \oplus \text{recol}[\gamma_2](\mathcal{H}_2))$$

$$\text{avec } \gamma_1((c, n)) = \begin{cases} (c, n) & \text{si } |\mathcal{G}_1^\infty \oplus \mathcal{G}_2^\infty|_c \leq k , \\ (c, 1) & \text{sinon,} \end{cases}$$

$$\text{et } \gamma_2((c, n)) = \begin{cases} (c, n + |\mathcal{G}_1^\infty|_c) & \text{si } |\mathcal{G}_1^\infty \oplus \mathcal{G}_2^\infty|_c \leq k , \\ (c, 1) & \text{sinon.} \end{cases}$$

3. Soient c_1 et c_2 des couleurs de C et supposons que $\mathcal{R}[\mathcal{G}^\infty](\sigma(\mathcal{H}), \mathcal{H})$. Alors

$$\text{recol}[c_1 \rightarrow c_2](\sigma(\mathcal{H})) = \sigma(\text{recol}[\gamma](\mathcal{H}))$$

$$\text{avec } \gamma((c_1, n)) = \begin{cases} (c_2, 1) & \text{si } \|\text{recol}[c_1 \rightarrow c_2](\mathcal{G}^\infty)\|_{c_2} > k, \\ (c_2, n) & \text{sinon,} \end{cases}$$

$$\gamma((c_2, n)) = \begin{cases} (c_2, 1) & \text{si } \|\text{recol}[c_1 \rightarrow c_2](\mathcal{G}^\infty)\|_{c_2} > k, \\ (c_2, n + |\mathcal{G}^\infty|_{c_1}) & \text{sinon,} \end{cases}$$

$$\gamma((c, n)) = (c, n) \text{ si } c \notin \{c_1, c_2\} .$$

4. Soient c_1 et c_2 des couleurs de C et supposons $\mathcal{R}[\mathcal{G}^\infty](\sigma(\mathcal{H}), \mathcal{H})$. Alors

$$\text{arc}[c_1, a, c_2](\sigma(\mathcal{H})) = \sigma(\text{arc}[D](\mathcal{H})) \quad \text{avec } D = \{(c_1, i), a, (c_2, j) \mid i, j \in [k]\} .$$

Ainsi, d'après le lemme 3.59, ces égalités peuvent être vues comme les transitions d'un transducteur déterministe \mathbf{t} satisfaisant les conclusions du lemme. \square

Récurrence

Les lemmes A.10, A.11, A.12, A.13 et A.14 correspondent aux transitions d'un transducteur déterministe. Ces lemmes sont tous organisés de manière identique et mettent en évidence leur lien avec le transducteur. Ainsi, leurs conclusions se divisent explicitement en quatre catégories :

- [v]** assure la **v**alidité, *c.a.d.* que les hypothèses (A-C) ou (A-E) sont bien satisfaites,
- [a]** assure que les différents cas peuvent être séparés par **a**nticipation reconnaissable,
- [d]** garantit que la construction est bien **H**R-définissable,
- [r]** explicite l'étape de **r**écurrence proprement dite.

Le premier lemme permet de se ramener de la relation \mathcal{R} à la relation \mathcal{R}' . Les autres traitent un par un les opérateurs **VR**.

Lemme A.10 *Pour tout \mathcal{G}_∞ , E et D satisfaisant (A-C), il existe E' et D' et f tels que :*

- [v]** \mathcal{G}^∞ , E' et D' satisfont (A-E)
- [a]** E' , D' et f ne dépendent que de E , D et $C_1(\mathcal{G}^\infty)$,
- [d]** f est **H**R-définissable,
- [r]** et pour tout \mathcal{G} et tout \mathcal{H} ,

$$\text{si } \mathcal{R}'[\mathcal{G}^\infty, E', D'](\mathcal{G}, \mathcal{H}) \quad \text{alors } \mathcal{R}[\mathcal{G}^\infty, E, D](\mathcal{G}, f(\mathcal{H})) .$$

Preuve. Soit $C_1 = C_1(\mathcal{G}_\infty)$ et $C_+ = C_+(\mathcal{G}^\infty)$. Comme $\text{arc}[D](\mathcal{G}^\infty)$ ne contient pas de graphe biparti complet de taille $k/2$ (d'après (C)), on en déduit que $D \cap D(C_+, C_+) = \emptyset$ (a).

Soient $D_0 = D \cap D(C_1 \cup E, C_1 \cup E)$, $D' = D - D_0$ et $E' = C(D') - C(\mathcal{G})$. Par construction et (a), E' , D' et \mathcal{G}^∞ satisfont (A-E).

Supposons $\mathcal{R}'[\mathcal{G}^\infty, E', D'](\mathcal{G}, \mathcal{H})$. Alors,

$$\begin{aligned}
& \text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}) \\
& \approx \text{arc}[D_0 \cup D'](\text{fusion}[E \cup C_1](\text{sommet}[E \cup C_1] \oplus \text{sommet}[E'] \oplus \mathcal{G})) \\
& \quad (\text{car } C_1(\mathcal{G}) = C_1 \text{ et (A)}) \\
& \approx \text{fusion}[E \cup C_1](\text{arc}[D_0](\text{sommet}[E \cup C_1]) \oplus \text{arc}[D'](\text{sommet}[E'] \oplus \mathcal{G})) \\
& \quad (\text{lemme 4.39 deux fois}) \\
& \approx \text{fusion}[E \cup C_1](\text{arc}[D_0](\text{sommet}[E \cup C_1]) \oplus \mathcal{H}) \\
& \quad (\text{car } \mathcal{R}'[\mathcal{G}^\infty, E', D'](\mathcal{G}, \mathcal{H}))
\end{aligned}$$

Soit donc $\mathcal{H}_0 = \text{arc}[D_0](\text{sommet}[E \cup C_1])$, alors on a

$$\mathcal{R}[\mathcal{G}^\infty, E, D](\mathcal{G}, \text{fusion}[C_1](\mathcal{H}_0 \oplus \mathcal{H})) .$$

□

Lemme A.11 *Pour tout, E et D satisfaisant (A-E) et toute couleur $c \in C$, il existe un graphe fini \mathcal{H}_0 tel que*

- [d] \mathcal{H}_0 est fini (et donc HR-définissable),
- [r] et $\mathcal{R}'[\text{sommet}[c], E, D](\text{sommet}[c], \mathcal{H}_0)$.

Preuve. Il suffit de prendre $\mathcal{H}_0 = \text{arc}[D](\text{sommet}[E \cup \{c\}])$. □

Lemme A.12 *Pour tout \mathcal{G}^∞ , E , D , c , c' et a tels que $\text{arc}[c, a, c'](\mathcal{G}^\infty)$, E , D satisfont (A-E), il existe D' et f tels que*

- [a] D' et f ne dépendent que de D , c , c' et a ,
- [v] \mathcal{G}^∞ , E' , D' satisfont (A-C),
- [d] f est HR-définissable,
- [r] pour tout graphes \mathcal{G} et \mathcal{H} , on a continuellement

$$\begin{aligned}
& \text{si } \mathcal{R}[\mathcal{G}^\infty, E', D'](\mathcal{G}, \mathcal{H}) \\
& \quad \text{alors } \mathcal{R}'[\text{arc}[c, a, c'](\mathcal{G}^\infty), E, D](\text{arc}[c, a, c'](\mathcal{G}), f(\mathcal{H})) .
\end{aligned}$$

Preuve. Il suffit de prendre $D' = D \cup \{(c, a, c')\}$ et $f(\mathcal{H}) = \text{arc}[\{(c, a, c')\}](\mathcal{H})$. \square

Lemme A.13 *Pour tout \mathcal{G}^∞, E, D et α tels que $\text{recol}[\alpha](\mathcal{G}^\infty), E, D$ satisfont (A-E), il existe D' et f tels que*

[a] D' et f ne dépendent que de E, D et α ,

[d] f est HR-définissable,

[r] pour tout graphes \mathcal{G} et \mathcal{H} , on a continuellement

$$\text{si } \mathcal{R}[\mathcal{G}^\infty, E, D'](\mathcal{G}, \mathcal{H}) \text{ alors } \mathcal{R}'[\text{recol}[\alpha](\mathcal{G}^\infty), E, D](\text{recol}[\alpha](\mathcal{G}), f(\mathcal{H})) .$$

Preuve.

$$\text{On étend } \alpha \text{ en } \alpha' \text{ sur } E \cup C \text{ par } \alpha'(c) = \begin{cases} \alpha(c) & \text{si } c \in C, \\ c & \text{sinon.} \end{cases}$$

$$\text{et on pose } D' = \{(c, a, c') \mid (\alpha'(c), a, \alpha'(c')) \in D\} .$$

On a alors la propriété suivante:

$$\text{arc}[D] \circ \text{recol}[\alpha'] = \text{recol}[\alpha'] \circ \text{arc}[D'] \quad (\#)$$

Supposons maintenant $\mathcal{R}[\mathcal{G}^\infty, E, D'](\mathcal{G}, \mathcal{H})$, alors,

$$\begin{aligned} & \text{arc}[D](\text{sommet}[E] \oplus \text{recol}[\alpha](\mathcal{G})) \\ &= \text{arc}[D](\text{sommet}[E] \oplus \text{recol}[\alpha'](\mathcal{G})) \\ &= \text{arc}[D](\text{recol}[\alpha'](\text{sommet}[E] \oplus \mathcal{G})) \\ &= \text{recol}[\alpha'](\text{arc}[D'](\text{sommet}[E] \oplus \mathcal{G})) && \text{(d'après (\#))} \\ &= \text{recol}[\alpha'](\mathcal{H}) && \text{(car } \mathcal{R}[\mathcal{G}^\infty, E, D'](\mathcal{G}, \mathcal{H})) \end{aligned}$$

Donc $f(\mathcal{H}) = \text{recol}[\alpha'](\mathcal{H})$ satisfait les conclusions du lemme. \square

Lemme A.14 *Pour tous $\mathcal{G}_1^\infty, \mathcal{G}_2^\infty, E, D$ tels que $\mathcal{G}_1^\infty \oplus \mathcal{G}_2^\infty, E$ et D satisfont (A-E), il existe E_1, D_1, E_2, D_2 et f tels que*

[a] E_1, E_2, D_1, D_2 et f ne dépendent que de $E, D, C_1(\mathcal{G}_1^\infty), C_1(\mathcal{G}_2^\infty)$,

[v] $\mathcal{G}_1^\infty, E_1, D_1$ satisfont (A-C),

[v] $\mathcal{G}_2^\infty, E_2, D_2$ satisfont (A-C),

[d] f est HR-définissable,

[r] pour tous graphes $\mathcal{G}_1, \mathcal{G}_2, \mathcal{H}_1$ et \mathcal{H}_2 , on a continuellement

$$\begin{aligned} & \text{si } \mathcal{R}[\mathcal{G}_1^\infty, E_1, D_1](\mathcal{G}_1, \mathcal{H}_1) \text{ et } \mathcal{R}[\mathcal{G}_2^\infty, E_2, D_2](\mathcal{G}_2, \mathcal{H}_2) \\ & \text{alors } \mathcal{R}'[\mathcal{G}_1^\infty \oplus \mathcal{G}_2^\infty, E, D](\mathcal{G}_1 \oplus \mathcal{G}_2, f(\mathcal{H}_1, \mathcal{H}_2)) . \end{aligned}$$

Preuve. Soit, pour $i \in [2]$, $C_1^i = C_1(\mathcal{G}_i^\infty) = C_1(\mathcal{G}_i)$ et $C^i = C(\mathcal{G}_i^\infty) = C(\mathcal{G}_i)$. On peut supposer (ce qui est le cas après normalisation pour $k > 2$) que $C_1^1 \cap C^2 = \emptyset$ et $C_1^2 \cap C^1 = \emptyset$.

Idée : L'intention est alors de poser $E_1 = E \cup C^2$, $E_2 = E \cup C^1$ et $D = D_1 = D_2$ ce qui ne satisfait pas l'hypothèse (A). Supposons néanmoins que $\mathcal{R}[\mathcal{G}_1^\infty, E_1, D_1](\mathcal{G}_1, \mathcal{H}_1)$ et $\mathcal{R}[\mathcal{G}_2^\infty, E_2, D_2](\mathcal{G}_2, \mathcal{H}_2)$ sauf en ce qui concerne l'hypothèse (A). On a alors,

$$\begin{aligned}
& \text{arc}[D](\text{sommet}[E] \oplus \mathcal{G}_1 \oplus \mathcal{G}_2) \\
& \approx \text{arc}[D](\text{fusion}[E \cup C^1 \cup C^2](\text{sommet}[E \cup C^2] \oplus \mathcal{G}_1) \oplus (\text{sommet}[E \cup C^1] \oplus \mathcal{G}_2)) \\
& \approx \text{fusion}[E \cup C^1 \cup C^2](\text{arc}[D](\text{sommet}[E \cup C^2] \oplus \mathcal{G}_1) \oplus (\text{sommet}[E \cup C^1] \oplus \mathcal{G}_2)) \\
& \approx \text{fusion}[E \cup C^1 \cup C^2](\text{arc}[D](\text{sommet}[E \cup C^2] \oplus \mathcal{G}_1) \\
& \quad \oplus \text{arc}[D](\text{sommet}[E \cup C^1] \oplus \mathcal{G}_2)) \\
& \quad \quad \quad (\text{lemme 4.40}) \\
& \approx \text{fusion}[E \cup C^1 \cup C^2](\mathcal{H}_1 \oplus \mathcal{H}_2) \\
& \quad (\text{car } \mathcal{R}[\mathcal{G}_1^\infty, E_1, D_1](\mathcal{G}_1, \mathcal{H}_1) \text{ et } \mathcal{R}[\mathcal{G}_2^\infty, E_2, D_2](\mathcal{G}_2, \mathcal{H}_2))
\end{aligned}$$

Donc $f(\mathcal{H}_1, \mathcal{H}_2) = \text{fusion}[E \cup C^1 \cup C^2](\mathcal{H}_1 \oplus \mathcal{H}_2)$ conviendrait.

Il manque l'hypothèse (A). La solution complète combine donc cette technique avec une duplication de C^1 et C^2 afin de les rendre disjoints de C . L'opération de recoloriage permet alors d'obtenir la même équivalence. \square

Index

– Symboles –

μ -définissabilité	68
E_{\perp} (cpo pointé)	49
Δ_1 (demi-droite infinie)	11
Δ_2 (arbre binaire complet)	12
$ R $ (arité de R)	9
$ f $ (arité de f)	13
$\sqrt[t]{}$	85
\perp (plus petit élément cpo)	48
\xrightarrow{w}_g (étiquettes chemin)	11
$t \downarrow_n$ (coupure profondeur n)	53
\sim	9
\xrightarrow{a} (arc étiqueté par a)	10
\approx (isomorphe)	9
\models (modèle)	16
$\llbracket \rrbracket$ (morphisme canonique)	55
\square (produit asynchrone)	32
\otimes_g (produit généralisé)	32
\otimes (produit synchrone)	31
\rightarrow_{Δ}	152
\rightarrow_{Δ}^*	152
\sqcup (borne supérieure)	48
$\mathcal{S}_{/\sim}$	10
\oplus (union disjointe)	29, 97, 104
t^v	14
$\mathbb{1}$ (plus grand élément treillis)	64
$\mathbb{0}$ (plus petit élément treillis)	64
$\text{Auto}_{\text{dd}}^{\mathcal{T}}$	131
Auto_{d}	90
$\text{Auto}^{\mathcal{T}}$	131
$\mathcal{LB}(t)$ (langage des branches)	13
$Cl(\phi)$	65
$dest$	10

d, d^+, d^- (degrés)	10
<i>final</i>	11
fix (plus petit point fixe)	50
FV (variables libres)	16, 63
HR	96
LA (largeur arborescente)	42
\mathcal{L} (langage accepté)	11
$\mathcal{L}_{I,F}$ (trace)	11
<i>orig</i>	10
R^S (interprétation de R)	9
<i>racine</i>	11
$\text{sem}[\mathbf{A}](f)$ (sémantique de f dans \mathbf{A})	54
$\text{Solution}[\mathbf{A}](S)$ (solution de S)	56
sommet	104
$\mathcal{T}_{\perp}^{\omega}(\mathcal{F})$ (cpo termes infinis)	53
$\mathcal{T}^{\omega}(\mathcal{F})$ (termes infinis)	13
$\mathcal{T}(\mathcal{F})$ (termes finis)	13
$\mathcal{U}_{\mathcal{S}}$ (univers de \mathcal{S})	9
$V_{\mathcal{G}}$ (sommets de \mathcal{G})	10
VR'	105
VRA	155
VRS'	135
VR	103
arc	104
fusion	97
recol	97, 104

– A –

accessibilité	
rationnelle	18
additivité	61
ajout fini	26
algèbre	54

continue	55	demi-droite des entiers	11
continue sortée	57	destination	10
libre	54		
alphabet gradué	13	– E –	
arbre	12	élimination des sommets isolés ...	28
binaire complet	12	équation	56
arc	10	escabeau	11
arité	9, 13	étiquette	10
automate	11	expression	
automate d'arbre	130	booléenne	17
complet	130	booléenne positive	17
déterministe ascendant	130		
déterministe descendant	131	– F –	
		feuille	12
– B –		formule	
borne inférieure	60	booléenne	17
borne supérieure	48, 60	booléenne positive	17
boucle	10	close	16, 63
branche	12	du premier ordre	15
		en forme linéaire [complète]	106
– C –		existentielle	17
μ -calcul	63	monadique	17
chaîne ascendante	48	fusion	97
chemin	10		
conjonction linéaire	106	– G –	
continuité	50	graphe	10
couleur	96	HR-équationnel	97
cpo	48	VR-équationnel	104
des applications	49	biparti complet	45
des graphes	53	co-déterministe	10
des langages	52	coloré	96
des parties	49	déterministe	10
des structures	52	de largeur arborescente bornée	44
des structures avec quotient	53	de récriture suffixe	154
des termes infinis	53	enraciné	11
fini	49	préfixe	91
pointé	49	préfixe reconnaissable	116
produit	49	uniformément creux	44
croissante	61		
		– H –	
– D –		hyperarc	9
degré	10	hypergraphe	8

– I –	
inévitable	190
interprétation	9, 26
monadique des parties finies	142
isomorphisme	
de structure avec quotient	9
– J –	
jeu du voleur et des policiers	42
– L –	
langage	
accepté par un automate	11
de mots	52
de termes	52
des branches	13
rationnel (de termes)	130
reconnaisable (de termes) ..	130
largeur arborescente	42
largeur de clique	172
limite	48
logique	
du premier ordre	15
du second-ordre gardée	20
monadique	17
– M –	
marquage rationnel	28
modèle	16
modification finie	26
morphisme	
canonique	55
d'algèbre	54
d'algèbre continue	55
inverse	28
multiplicativité	61
– N –	
nœud	12
– O –	
opérateur	54
ordre partiel complet	48
origine	10
– P –	
père	12
point fixe	61
(plus petit) point fixe	50
présentation	
automatique	149
terme-automatique	134
produit	
asynchrone	32
généralisé	32
synchrone	31
projection	51
– R –	
racine	11
recoloriage	97
réécriture suffixe	152
relation	
continue	84
PR	119
terme-synchronisée	133
renommage	27
restriction	
par accessibilité	28
rationnelle	28
– S –	
satisfaction	16
sémantique	54
signature	9
solution (système d'équations) ...	56
sommet	10
accessible	11
final	11
principal	13
sous-terme	14
structure	9
HR-équationnelle	97
avec quotient	9
colorée	96

quotient	10
substitution	
finie inverse	28
rationnelle inverse	28
symbole	
fonctionnel	13
relationnel	9
système d'équations	56
normalisé	57

– **T** –

terme, \mathcal{F} -terme	13
théorie	
au premier ordre	16
trace (d'un graphe)	11
transducteur déterministe	72
transduction monadique	35
treillis complet	60

– **U** –

union disjointe	29
union disjointe explicite	30

– **V** –

valeur	55
valuation	16, 56, 64
variable	
(du second ordre) monadique	17
du μ -calcul	63
du premier ordre	15
libre	16

Résumé

Ce travail est dédié à l'étude de structures et de graphes infinis admettant une représentation finie, aux équivalences entre ces représentations et aux propriétés géométriques et de décidabilité les concernant. En suivant l'approche de Courcelle, nous nous attachons à leur définition en terme de solution de systèmes d'équations.

Nous introduisons tout d'abord la notion de transducteur déterministe (descendant avec anticipation de termes infinis). Cet outil permet de manipuler des systèmes infinis d'équations, et en cela, son usage permet d'étendre nombre de résultats connus antérieurement, établis dans le cas de systèmes finis.

Nous étudions ensuite les familles de structures à base de pile en nous concentrons sur les structures préfixes reconnaissables. Nous établissons diverses équivalences de représentations et étudions la restriction à une largeur arborescente bornée. Nous considérons enfin l'ajout d'un opérateur supplémentaire de fusion aux systèmes d'équations.

La contribution suivante concerne les structures terme-automatiques. Nous montrons qu'elles admettent aussi diverses représentations équivalentes.

Pour finir, nous étudions les graphes décrits par réécriture suffixe de termes et montrons diverses équivalences de représentations les concernant. Nous finissons par l'étude de ces graphes quand restreints à une largeur arborescente ou une largeur de clique bornée.

Summary

This work is dedicated to the study of infinite structures and graphs which admit a finite representation, to their geometrical properties as well as decidability properties concerning them. Following Courcelle's work, we focus our study on their representation as solution of equational systems.

We first introduce deterministic transducers (top-down with lookahead for infinite trees). This tool allows us to deal with infinite equational systems, thus extending some previous results, originally stated for finite equational systems.

We then study the stack based families of structures and concentrate ourselves on prefix recognizable ones. We establish various equivalent representations for them, and study their restriction to bounded tree-width. We finally consider the enrichment of those equational systems by an extra operator of fusion.

We then study structures admitting term-automatic presentations. We show once more that those structures admit various equivalent representations.

Finally, we study the family of graphs definable by ground term rewriting and establish some equivalences of representations. We conclude by a study of the nature of those graphs when restricted to bounded tree or clique width.