N° d'ordre : ????

UNIVERSITÉ CLERMONT-FERRAND 1
ÉCOLE DOCTORALE SCIENCES POUR L'INGENIEUR
Laboratoire de Logique, Algorithmique et Informatique de Clermont 1

Thèse

présentée par

Annie CHATEAU

pour obtenir le grade de

Docteur spécialité SCIENCES

Utilisation des destinées pour la décision et sa complexité dans le cas de formules à profondeur de quantification bornée sur des structures logiques finies et infinies.

Soutenance prévue le : 1er juillet 2003

Après avis des rapporteurs :

Patrick Cegielski ..	Professeur
Etienne Grandjean	Professeur
Michal Krynicki ..	Professeur

Devant la commission d'examen composée de :

Patrick Cegielski ..	Professeur	Examinateur
Etienne Grandjean	Professeur	Examinateur
Michal Krynicki ..	Professeur	Examinateur
Francis Nézondet .	Docteur	Examinateur
Malika More	Maîtresse de Conférence	Co-directrice de thèse
Denis Richard	Professeur	Directeur de thèse

– 2003 –

Utilisation des destinées pour la décision et sa complexité dans le cas de formules à profondeur de quantification bornée sur des structures logiques finies et infinies.

Annie Chateau

Résumé

Nous étudions les structures logiques finies ou infinies à travers les énoncés de profondeur de quantification donnée qui sont vrais dans ces structures.

Cette étude porte principalement sur un nouvel outil logique appelé k-destinées de Nézondet. Une k-destinée d'une structure consiste en une présentation arborescente des types de k-isomorphisme de Fraïssé de la structure. Nous analysons ce nouvel outil, en particulier nous montrons que les destinées s'intègrent parfaitement au contexte des k-isomorphismes de Fraïssé et des jeux d'Ehrenfeucht.

Nous détaillons un algorithme de décision utilisant les destinées. Nous comparons en détail les structures dont on peut construire récursivement les destinées et les structures H-bornées. Enfin, nous donnons quelques résultats intermédiaires du problème NE ?= CoNE et de sa version logique, la conjecture du Spectre.

Using Destinies for the Decision Problem of Sets of Bounded Quantifier Depth Sentences in Finite or Infinite Structures.

Abstract

We study finite or infinite logical structures, by considering the sets of sentences with quantifier depth k that are true in these structures. More spacifically, we use a new logic tool called Nézondet destinies. A k-destiny of a structure is a tree containing all the k-isomorphism types of the structure. We show that destinies are a very relevant notion, equivalent to Fraïssé k-isomorphism and Ehrenfeucht games.

We present a decision algorithm using destinies. We carefully compare the class of structures for which there exists an algorithm to construct destinies, and the class of H-bounded structures. Finally, we give some partial results on the NE vs. CoNE problem and its logic equivalent, the Spectrum conjecture.

Table des matières

Table des figures

Introduction générale

Cette thèse porte sur l'étude de structures de taille finie ou infinie, sur des langages relationnels finis. On s'intéresse particulièrement aux énoncés de profondeur de quantification bornée qui sont vrais dans ces structures.

Les outils traditionnels pour étudier les énoncés de profondeur de quantification fixée k sont les k-isomorphismes de Fraïssé ([Fra72]) ainsi que les jeux d'Ehrenfeucht du premier ordre en k coups ([EF99]). Nous présentons un troisième outil, les destinées de Nézondet ([Néz97]), dont la nature est différente. Les k-isomorphismes de Fraïssé ainsi que les jeux d'Ehrenfeucht sont des moyens de *comparaison* entre les structures. Les destinées de Nézondet sont des outils de *description* d'une structures : elles représentent, de manière arborescente, tous les types de k-isomorphisme satisfaisables dans la structure.

La première partie de la thèse contient une présentation des destinées. Nous y décrivons un algorithme de décision des énoncés de profondeur de quantification donnée dans une structure. Cet algorithme utilise les destinées de la structure. Nous comparons également les destinées de Nézondet avec les deux autres outils déjà cités, les k-isomorphismes de Fraïssé et les jeux d'Ehrenfeucht.

Le fil conducteur de la deuxième partie de la thèse est la construction des destinées. A travers deux exemples, nous constatons que l'automatisation de cette construction n'est pas toujours réalisable, et que cette automatisation est en partie liée à la décidabilité de la théorie complète de la structure choisie. Nous essayons ensuite de cerner dans quels cas l'automatisation de la construction des destinées est possible. Nous étudions en détail le cas des structures H-bornées, pour lesquelles on connaît une élimination des quantificateurs. Dans ce cas, nous comparons la complexité de la décision des énoncés par l'algorithme utilisant les destinées et par la méthode classique due à Ferrante et Rackoff ([FR79]).

La troisième partie consiste en la présentation de résultats obtenus autour du problème $\mathrm{NE} \stackrel{?}{=} \mathrm{CoNE}$ (le complémentaire d'un ensemble reconnaissable en temps exponentiel non déterministe est-il également un ensemble reconnaissable en temps exponentiel non déterministe ?). Plus précisément, nous étudions l'équivalent logique de ce problème, la conjecture du Spectre ([Ass55]), ainsi qu'une conjecture plus forte, proposée par Ash ([Ash94]). Les résultats développés font appel à l'étude des classes de k-isomorphisme de structures finies. Nous utilisons abondamment les destinées et les jeux d'Ehrenfeucht afin de caractériser ces classes de k-isomorphisme.

Première partie : Destinées et décision

Les destinées ont été introduites par Francis Nézondet dans sa thèse ([Néz97]), soutenue au LLAIC1 (Laboratoire de Logique, Algorithmique et Informatique de Clermont-Ferrand 1) en 1997. Nézondet définit les destinées et expose un algorithme de décision des énoncés de profondeur de quantification fixée utilisant les destinées. Cette première partie reprend les travaux de Nézondet. Nous adoptons une terminologie légèrement différente et nous complétons certaines notions. La Partie I se décompose en cinq chapitres.

Chapitre 1 : Préliminaires

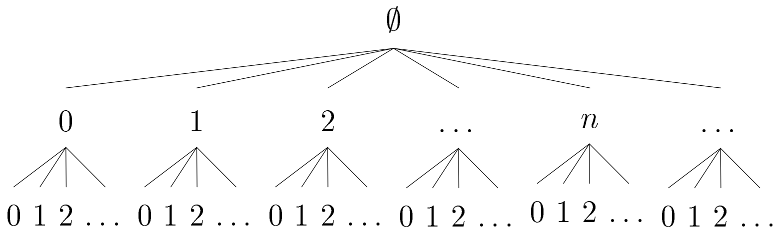
Le Chapitre 1 est un ensemble de rappels de logique, dont le lecteur averti peut se dispenser.

Chapitre 2 : Une définition des destinées d'une structure

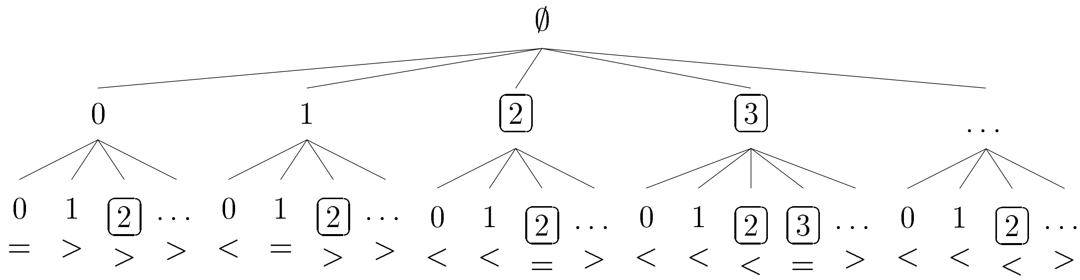
Le Chapitre 2 présente quelques rappels sur les arbres, et donne une définition des destinées en tant que structures sur un langage donné. En particulier, on s'attache à la notion de destinée d'une structure sur un langage relationnel fini. En voici un exemple informel : on se donne la structure de domaine $\mathbb{N}$, les entiers naturels, sur le langage $\{\leq, P\}$, où le prédicat $\leq$ est interprété comme l'ordre naturel sur les entiers, et le prédicat unaire P comme la relation "être premier".

Ce qu'on appelle 2-destinée de $\langle \mathbb{N}, \leq, P \rangle$ est une présentation arborescente des types de 2-isomorphisme sur cette structure, et plus précisément un arbre complet de hauteur 3, dont la racine est l'élément vide $\emptyset$, et les autres nœuds sont des éléments de la structure (donc ici des entiers).

Pour que la destinée contienne tous les types de 2-isomorphisme (dans ce cas on dira qu'elle est **exhaustive**), on commence par considérer un arbre complet de hauteur 3 avec comme racine $\emptyset$ et tel que chaque nœud a pour fils tous les éléments de la structure :



On a donc un arbre de degré infini, de hauteur 3. Pour tenir compte des relations du langage telles qu'elles sont interprétées dans la structure, on fait ensuite apparaître les relations satisfaites par un nœud, avec lui-même et tous ses ascendants (sauf la racine). Cela apparaît dans notre exemple, pour l'ordre : sous la forme d'un signe "<" si le fils est inférieur au père, un signe ">" si le fils est supérieur au père et du signe "=" s'ils sont égaux ; pour le caractère premier : sous la forme d'un cadre autour du nœud s'il est premier :

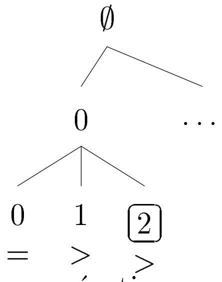


Chapitre 3 : Isomorphismes dans les destinées

Le Chapitre 3 formalise deux notions fondamentales lorsque l'on utilise les destinées d'une structure :
- la notion de destinée exhaustive, suggérée dans l'exemple ci-dessus,
- la notion de sous-arbres isomorphes dans une destinée.

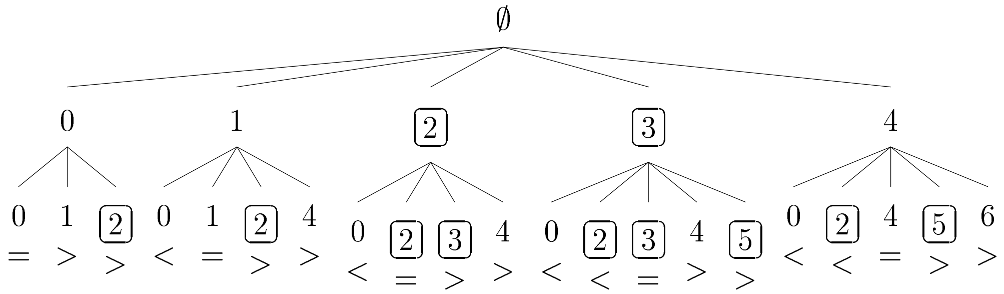
Reprenons l'exemple de la 2-destinée de la structure $\langle \mathbb{N}, \leq, P \rangle$. Il s'agit maintenant de réduire cet objet infini à une 2-destinée de taille finie que l'on qualifiera d'**essentielle** (il n'y a qu'un nombre fini de types de 2-isomorphisme), en procédant de la façon suivante :
- Au niveau des feuilles, si deux feuilles d'un même sous-arbre présentent les mêmes relations avec leurs ascendant (feuille comprise), alors on garde la plus petite et on élimine l'autre. Par exemple, sur le sous-arbre de 0, on ne garde que les trois feuilles 0, 1 et 2 car tout nombre n supérieur à 2 est tel que :
  - s'il est premier, le couple $(0,n)$ est isomorphe au couple $(0,2)$;
  - s'il n'est pas premier, le couple $(0, n)$ est isomorphe au couple $(0,1)$.

  Cela nous donne, pour ce sous-arbre :



- Une fois que l'on a réalisé la même opération sur tous les sous-arbres, on regarde quels sous-arbres sont isomorphes, et on garde le plus petit représentant de chaque classe. Il reste les 5 sous-arbres de 0, 1, 2, 3 et 4. En effet si n est un nombre entier supérieur à 4 :
  - s'il est premier, le sous-arbre de n est isomorphe au sous-arbre de 3 ;
  - s'il n'est pas premier, le sous-arbre de n est isomorphe au sous-arbre de 4.

  Cela nous donne, finalement :



Chapitre 4 : Satisfaction des énoncés et algorithme de décision utilisant les destinées

Le Chapitre 4 explique comment transporter la satisfaction d'un énoncé de profondeur de quantification k dans une structure vers la satisfaction d'un énoncé modifié, appelé forme destinale, dans une k-destinée exhaustive de la structure de départ. L'algorithme de décision des ensembles d'énoncés de profondeur de quantification inférieure ou égale à k qui est associé à cette transformation est également présenté.

Informellement, sur notre exemple, l'algorithme de décision est le suivant : une fois la destinée construite, on peut décider simplement les énoncés du premier ordre sur le langage choisi (ici $\{P, <\}$) de profondeur de quantification inférieure à la hauteur de la destinée (ici ≤ 2), en "vérifiant" si l'énoncé est vrai sur la destinée, c'est-à-dire en instanciant ses variables de la façon suivante : prenons l'exemple de l'énoncé $\forall x \exists y (P(y) \wedge (P(x) \vee (x < y)))$. Imaginons quelques minutes que nous ne savons pas s'il s'agit d'un théorème, et notons $F[x, y] = P(y) \wedge (P(x) \vee (x < y))$.
Décomposons cet énoncé :

- le premier quantificateur est un $\forall$, il va donc falloir donner à x successivement toutes les valeurs des nœuds fils de $\emptyset$ de la destinée et faire tourner l'algorithme sur chacun des sous-arbres : si tous les résultats sont positifs (la sous-formule $\exists y F[x, y]$ est vraie quand on attribue à x la valeur du nœud), c'est que l'énoncé est un théorème ; si l'un des résultats est négatif, c'est que l'énoncé n'est pas un théorème, et on a un contre-exemple explicite.
- le deuxième quantificateur est un $\exists$, on va donc encore une fois, pour une valeur donnée de x parmi les valeurs des nœuds fils de $\emptyset$, donner successivement à y toutes les valeurs des fils de x dans la destinée. Cette fois les critères de "réussite" sont différents : si l'un des résultats est positif, c'est que la sous-formule $\exists y F[x, y]$ est vraie pour cette valeur de x, et on a un y qui convient explicitement ; si tous les résultats sont négatifs, alors c'est que la sous-formule $\exists y F[x, y]$ est fausse pour cette valeur de x.

Faisons tourner cet algorithme sur l'énoncé choisi : on donne à x les valeurs 0, 1, 2, 3, 4 successivement, et pour chacune de ces valeurs on donne à y les valeurs des fils du nœud correspondant à la valeur de x (0, 1, 2 pour $x = 0$; 0, 1, 2, 4 pour $x = 1$; 0, 2, 3, 4 pour $x = 2$; 0, 2, 3, 4, 5 pour $x = 3$ et 0, 2, 4, 5, 6 pour $x = 4$), et on évalue $F[x, y]$. Pour chaque valeur de x, on a une valeur de y pour laquelle $F[x, y]$ est vérifiée, donc l'énoncé initial est vrai.

La complexité de cet algorithme est en $\mathcal{O}(N_p \times n^2)$, où N_p est le nombre de nœuds de la destinée et n est la taille de la formule. Une présentation détaillée de l'algorithme peut être trouvée dans [Cha00], [Cha01] et [Cha02].

L'interêt de cette méthode de décision réside dans le fait qu'une fois la destinée construite, et stockée, on peut utiliser cet algorithme très simple pour décider un très grand nombre d'énoncés. Cela soulève la question de la construction des destinées, qui est l'objet de la deuxième partie.

Chapitre 5 : Comparaison des destinées avec les k-isomorphismes et les jeux

On peut trouver une présentation détaillée des k-isomorphismes de Fraïssé dans divers ouvrages, tels que [Fra72], [Hod93] ou [Poi85], et des jeux d'Erhenfeucht-Fraïssé dans [EF99] ou [Hod93].

Le principal résultat du Chapitre 5 est le théorème suivant :

Theorème 1
Soient $\mathcal{A}$ et $\mathcal{B}$ deux σ-structures, $(a_1, \ldots, a_n)$ et $(b_1, \ldots, b_n)$ deux n-uples d'éléments de $\mathcal{A}$ et $\mathcal{B}$ respectivement. Soit $k \geq 0$. Les assertions suivantes sont équivalentes :

1. *Les n-uples $(a_1, \ldots, a_n)$ et $(b_1, \ldots, b_n)$ sont k-isomorphes (au sens de Fraïssé).*
2. Joueur II *a une stratégie gagnante pour le jeu d'Ehrenfeucht en k coups $G_k((\mathcal{A}, a_1, \ldots, a_n), (\mathcal{B}, b_1, \ldots, b_n))$.*
3. $\mathcal{B} \models \varphi^k_{(a_1,\ldots,a_n)}(b_1, \ldots, b_n)$.
4. *Les n-uples $(a_1, \ldots, a_n)$ dans $\mathcal{A}$ et $(b_1, \ldots, b_n)$ dans $\mathcal{B}$ satisfont respectivement les mêmes formules de profondeur de quantification k à n variables libres.*
5. *Deux k-destinées exhaustives des structures $(\mathcal{A}, a_1, \ldots, a_n)$ et $(\mathcal{B}, b_1, \ldots, b_n)$ sont isomorphes.*

Grâce à ce théorème, on dispose d'un éventail de techniques équivalentes pour étudier les énoncés à profondeur de quantification bornée.

Deuxième partie : Structures infinies

L'algorithme de décision décrit au Chapitre 4 fait appel à une k-destinée de la structure, où k est la profondeur de quantification des énoncés à décider. Il faut donc, pour avoir un processus de décision complet, disposer d'un algorithme de calcul de cette k-destinée.

Dans le cas d'une structure finie, les destinées sont faciles à construire à partir de l'arbre complet, car celui-ci est fini. Mais dans le cas d'une structure infinie, bien qu'une destinée exhaustive et essentielle de la structure soit un objet fini, l'arbre complet permettant d'y aboutir est infini. Il faut donc pouvoir déterminer les "limites" d'une destinée exhaustive et essentielle afin de construire un arbre ne contenant qu'un nombre fini mais suffisant de nœuds.

La Partie II s'intéresse au processus de calcul des destinées d'une structure. On commence par considérer deux exemples, un cas où la construction n'est pas facile, et un cas où cette construction est automatisable. Puis nous généralisons le processus automatique de calcul des destinées à toute une classe de structures. La Partie II se compose de quatre chapitres.

Chapitre 6 : Construction de la 3-destinée réduite de $\langle \mathbb{N}, S, \perp \rangle$

Le Chapitre 6 présente un premier exemple de construction de destinée d'une structure. La structure considérée est la structure arithmétique $\langle \mathbb{N}, S, \perp \rangle$ (les entiers naturels munis du successeur et de la coprimarité), dont la théorie complète est indécidable.

Cette tentative est inachevée, et aboutit à cinq problèmes ouverts en arithmétique qui correspondent à cinq sous-arbres pour lesquels on ne sait pas s'ils apparaissent dans la destinée (on n'a ni trouvé de témoin, ni prouvé qu'il n'y avait pas de témoin pour ces sous-arbres).

On dispose alors d'un éventail de 32 algorithmes de décision possibles des énoncés de profondeur de quantification inférieure ou égale à 3 sur cette structure, mais on ne sait pas lequel est le bon.

Chapitre 7 : Un algorithme de construction des destinées de $\langle \mathbb{N}, \leq \rangle$

Le Chapitre 7 présente un deuxième exemple de construction de destinées d'une structure. Contrairement au chapitre précédent, cette construction se passe bien, dans le sens où elle est toujours possible, et automatisable.

La structure considérée est la structure $\langle \mathbb{N}, < \rangle$, dont la théorie complète est décidable. Nous calculons les bornes indiquant à quel nœud s'arrêter lorsque l'on construit l'arbre pour que la destinée soit exhaustive (à chaque niveau, ces bornes déterminent le nombre de fils du nœud que l'on est en train d'examiner pour que l'ensemble de l'arbre soit exhaustif).

Nous présentons l'algorithme de construction d'une k-destinée réduite de la structure pour tout k, à partir de ces bornes.

Chapitre 8 : Comparaison entre structures à destinées récursives et structures H-bornées

Le Chapitre 8 consiste en une généralisation du processus de construction décrit au Chapitre 7. Nous rappelons la notion de structure H-bornée ([FR79] ou [Dub95]). Les structures H-bornées admettent une élimination des quantificateurs et possèdent donc des théories complètes décidables.

Le résultat de la comparaison entre les structures H-bornées et les structures dont on peut construire algorithmiquement les destinées est le suivant : si la structure est H-bornée, avec H calculable, alors on peut, pour tout $k \geq 1$, construire une k-destinée algorithmiquement. En revanche, il existe une structure, que l'on présente dans cette partie, pour laquelle on peut construire les destinées, mais qui n'est H-bornée pour aucune fonction H calculable.

Chapitre 9 : Comparaison entre l'algorithme de Nézondet et l'algorithme de Ferrante et Rackoff

Dans le cas où la structure est H-bornée, nous disposons, d'après le Chapitre 8, de deux algorithmes de décision :

- l'algorithme de décision dû à Ferrante et Rackoff ([FR79]), utilisant une élimination des quantificateurs,
- l'algorithme de décision de Nézondet, utilisant les destinées de la structure.

Le Chapitre 9 présente une comparaison des complexités de ces deux algorithmes, et les avantages et inconvénients respectifs des deux méthodes.

Troisième partie : Structures finies

Outre l'application à la décision, les destinées sont utilisées, au même titre que les autres outils que sont les jeux et les k-isomorphismes, dans un problème relevant du domaine de la théorie des modèles finis. Le problème initial est le suivant : "est-ce que la classe de complexité NE et la classe de complexité CoNE coïncident ?", autrement dit, "le complémentaire d'un langage reconnaissable en temps exponentiel non déterministe est-il également un langage reconnaissable en temps exponentiel non déterministe ?". La Partie III présente ce problème de complexité sous un angle particulier qui est celui de la théorie des modèles finis : on se ramène à un problème équivalent en théorie des modèles finis, la conjecture du Spectre ([Ass55]). Cette partie se décompose en cinq chapitres. Les deux premiers décrivent respectivement le contexte et les méthodes utilisées, les trois derniers présentent les résultats obtenus au cours de ce travail de thèse.

Chapitre 10 : le problème NE $\stackrel{?}{=}$ CoNE et la conjecture du Spectre

Le Chapitre 10 présente le contexte général du problème NE $\stackrel{?}{=}$ CoNE. Ce problème possède une formulation équivalente en théorie des modèles finis, intitulée "conjecture du Spectre" ([Ass55]). Le spectre d'un énoncé du premier ordre est l'ensemble des cardinaux des modèles finis de cet énoncé. La conjecture du Spectre est la suivante : "Le complémentaire d'un spectre est-il un spectre ?". La classe des parties de $\mathbb{N}$ qui sont des spectres coïncide exactement avec la classe des langages reconnaissables en temps exponentiel non déterministe ([JS74]), d'où l'équivalence entre les deux formulations.

La conjecture du Spectre est impliquée par une autre conjecture de théorie des modèles finis, la conjecture de Ash ([Ash94]). Cette conjecture porte sur le comportement d'une fonction de comptage des classes de k-isomorphismes en fonction de la taille des structures sur un langage relationnel fini σ :

$$n \mapsto N_{\sigma,k}(n),$$

où $N_{\sigma,k}(n)$ est le nombre de σ-structures de taille n qui ne sont pas k-isomorphes.

L'énoncé de la conjecture est le suivant :

"Pour tout k et tout σ, la fonction $n \mapsto N_{\sigma,k}(n)$ est ultimement constante."

La vérité de cette conjecture implique un résultat positif pour la conjecture du spectre et par conséquent, implique NE = CoNE. Ash en suggère une version légèrement affaiblie (la fonction est ultimement périodique), qui est également suffisante (mais non nécessaire) pour démontrer la conjecture du Spectre.

Ce chapitre présente un affaiblissement de cette conjecture qui donne une condition non seulement suffisante mais aussi nécessaire sur le comportement de la fonction $N_{\sigma,k}$ pour que la conjecture du Spectre soit vraie. Nous appelons cette version affaiblie "la conjecture de Ash ultrafaible".

Chapitre 11 : Angles d'attaque de la conjecture de Ash et de la conjecture du Spectre

Le Chapitre 11 présente les méthodes employées pour tenter de résoudre la conjecture du Spectre, notamment en réalisant des restrictions :
- syntaxiques (on fixe l'arité des relations, par exemple, en étudiant le cas où σ est composé uniquement de relations unaires, ou uniquement de relations binaires) ;
- sémantiques (on associe une théorie au langage, et on compte non plus les classes de k-isomorphisme de structures, mais les classes de k-isomorphisme de modèles de cette théorie).

Chapitre 12 : Le cas $k = 2$

Le Chapitre 12 présente un résultat partiel de la conjecture de Ash et de la conjecture du Spectre. Nous montrons que pour des énoncés de profondeur de quantification 2, la conjecture de Ash est vérifiée, et par conséquent la conjecture du Spectre l'est également.

Chapitre 13 : Résultats sur le cas binaire

Le Chapitre 13 présente une série de résultats concernant des théories sur des langages d'une relation binaire et l'égalité. Le cas général se ramène au cas d'une relation de graphe, via un rembourrage polynomial, c'est pourquoi nous nous intéressons, dans ce chapitre, à des cas particuliers de relations de graphe. Nous montrons notamment que la conjecture de Ash périodique est vraie dans le cas d'un graphe de bijection, d'un graphe de fonction, d'un graphe orienté ou non orienté de degré total 2. Nous montrons également que dans le cas d'un graphe de bijection, la conjecture de Ash constante est fausse. Dans tous ces cas, la conjecture du Spectre est vraie, et l'on sait préciser la forme du complémentaire des spectres.

Chapitre 14 : Résultats sur les équivalences

Le Chapitre 14 présente des résultats de même nature concernant la théorie d'une relation d'équivalence et l'égalité : dans ce cas, la conjecture de Ash périodique est vraie. On présente également le cas de deux relations d'équivalence imbriquées, pour lequel la conjecture de Ash périodique est vraie. On montre enfin que le cas général peut se ramener au cas de deux relations d'équivalence et l'égalité via un rembourrage polynomial.

Les différents exemples présentés permettent de dresser un état de l'art et de fournir des pistes pour poursuivre l'étude de la conjecture du spectre.

Première partie

Destinées et décision

Cette première partie est dédiée à des rappels de logique et des définitions concernant les destinées de Nézondet. On a repris certains travaux que Francis Nézondet a exposé dans sa thèse ([Néz97]), où les destinées ont fait pour la première fois leur apparition sur la scène publique. Au-delà du caractère pittoresque de leur nom, les destinées de Nézondet apportent beaucoup à la compréhension globale d'une structure, et de son expressivité à profondeur de quantification fixée.

Francis Nézondet a défini, décrit et utilisé les destinées, mais il reste encore beaucoup à dire sur ces objets mathématiques. Cette première partie correspond tout d'abord à un travail de "lissage" et de simplification des notions présentées par F. Nézondet. Pour cette raison, les lecteurs des deux thèses ne retrouveront pas exactement les mêmes termes ni les mêmes définitions, mais les notions les plus importantes sont bien sûr conservées.

Le terme de "destinée" n'a pas été choisi innocemment. En effet la problématique dans laquelle les destinées ont été introduites est celle de la décidabilité d'ensemble d'énoncés de profondeur de quantification fixée (ces notions sont rappelées dans le Chapitre 1). Les destinées sont des arbres dont la hauteur correspond à la profondeur de quantification choisie, qui illustrent tout ce qui est vrai, dans une structure donnée, à cette profondeur de quantification. Une p-destinée porte donc en elle le "destin" à la profondeur de quantification p des éléments de la structure. Nous verrons qu'il s'agit en fait de représentants des classes de p-isomorphisme au sens de Fraïssé, et que ces deux notions, destinées et p-isomorphismes, sont intimement liées.

Cette première partie se décompose en cinq chapitres.

Dans le Chapitre 1, on rappelle quelques notions essentielles de logique : langages, formules, théories, modèles, problème de décision.

Dans le Chapitre 2, on présente une définition des destinées ainsi que le vocabulaire nécessaire sur les arbres.

Dans le Chapitre 3, nous définissons deux propriétés des destinées, à savoir l'essentialité (signifiant qu'il n'y a pas de redondance entre plusieurs sous-arbres dans une destinée) et l'exhaustivité (signifiant que toute configuration à la profondeur p est bien représentée dans la destinée). Nous introduisons également une notion de réduction qui permet de normaliser pour certaines structures les destinées qui leur sont associées.

On décrit ensuite au Chapitre 4 l'algorithme de décision de l'ensemble des énoncés de profondeur de quantification inférieure ou égale à p qui sont vrais dans une structure donnée. Cet algorithme s'obtient en utilisant une destinée exhaustive de rang p de cette structure. Il s'appuie sur le transport des énoncés du langage initial vers un énoncé du langage des destinées, tel que l'énoncé initial est vrai dans la structure si et seulement si l'énoncé transformé est vrai dans la destinée.

Enfin, on présente dans le Chapitre 5 la comparaison entre les destinées de Nézondet, les isomorphismes de Fraïssé et les jeux d'Ehrenfeucht, à savoir que ces trois outils permettent d'étudier les mêmes propriétés, tout en gardant chacun leur domaine d'application privilégié. Il s'agit en fait de trois points de vue différents sur la même notion.

Chapitre 1

Préliminaires

On rappelle quelques notions élémentaires de logique, plus amplement détaillées dans [CL94] ou [End72].

1.1 Langages, formules et structures

1.1.1 Langages et formules

Définition 1 (Langage du premier ordre)
Un **langage** *est un ensemble de symboles qui peuvent être :*
- *des symboles de constante c ;*
- *des symboles de relation R ;*
- *des symboles de fonction f.*

On ne considère dans cette thèse que des langages qui ne comportent pas de symbole de fonction. Les symboles de constante pouvant être assimilés à des symboles de relation d'arité 0, on appelle un tel langage sans symbole de fonction un **langage relationnel**. On exige de plus que le langage soit **fini** (c'est-à-dire qu'il ne comporte qu'un nombre fini de symboles). Ces langages ne sont pas nécessairement **égalitaires** (un langage égalitaire est un langage qui comporte le symbole d'égalité =).

On se fixe désormais un langage σ.

Définition 2 (Variables et termes)
Soit V un ensemble dénombrable de symboles, disjoint de σ. L'ensemble V est appelé ensemble de **variables***, et on définit les* **termes du premier ordre** *sur le langage σ par induction :*
- *les symboles de constante et les variables sont des termes ;*
- *pour tout symbole de fonction f d'arité k dans σ, et pour tous termes du premier ordre $t_1, \ldots, t_k$ du langage σ, l'expression $f(t_1, \ldots, t_k)$ est un terme du premier ordre.*

Définition 3 (Formules)
Les **formules du premier ordre** *sur le langage* σ *(ou* σ-**formules**)*, sont définies comme suit :*
- *pour tout symbole de relation* r *d'arité* k *du langage* σ*, et tous termes* $t_1, \ldots, t_k$ *du premier ordre sur* σ*, l'expression* $r(t_1, \ldots, t_k)$ *est une formule du premier ordre sur* σ *;*
- *pour toutes formules* F *et* G *du premier ordre sur le langage* σ*, les expressions* $(\neg F)$*,* $F \vee G$ *et* $F \wedge G$ *(respectivement "non* F*", "*F *ou* G*" et "*F *et* G*") sont des formules du premier ordre sur le langage* σ *;*
- *une variable est dite* **libre** *si elle n'est pas quantifiée par un quantificateur existentiel ou universel. Elle est* **liée** *sinon. Pour toute formule* F *du premier ordre sur* σ *et toute variable* x *libre dans* F*, les expressions* $\exists x(F)$ *et* $\forall x(F)$ *(respectivement, "pour tout* x*,* F*" et "il existe* x *tel que* F*") sont des formules du premier ordre sur le langage* σ*.*

Définition 4 (Énoncé)
Une formule sans variable libre est appelée un **énoncé***, ou encore* **formule close***.*

1.1.2 Théories et modèles

Définition 5 (Structure)
Une **structure** $\mathcal{X}$ *sur un langage* σ *(ou* σ*-structure) est un couple* $\langle X, \sigma^{\mathcal{X}} \rangle$*, où :*
- X *est un ensemble, appelé le* **domaine** *de la structure* $\mathcal{X}$ *;*
- $\sigma^{\mathcal{X}}$ *est un ensemble de constantes, relations et fonctions sur le domaine* X*, chacune associée à un symbole de même nature et de même arité du langage* σ*, appelées* **interprétations** *des symboles de* σ *dans la structure* $\mathcal{X}$*.*

Définition 6 (Théorie)
Une **théorie** *du premier ordre* $\mathcal{T}$ *sur un langage* σ *est un ensemble d'énoncés du premier ordre sur le langage* σ*.*

Définition 7 (Satisfaction des formules)
Soit $\mathcal{X}$ une σ-structure et F une formule du premier ordre sur σ à k variables libres. Soient $a_1, \ldots, a_k$ des éléments du domaine X. On dit que $(\mathcal{X}, a_1, \ldots, a_k)$ **satisfait** *F, ou que F est* **vraie** *dans $(\mathcal{X}, a_1, \ldots, a_k)$, et on note*

$$(\mathcal{X}, a_1, \ldots, a_k) \models F(a_1, \ldots, a_k),$$

lorsque la formule $F(a_1, \ldots, a_k)$ est vraie quand ses variables liées parcourent X.

Définition 8 (Modèle)
Une σ-structure $\mathcal{X}$ est **modèle** *d'une théorie $\mathcal{T}$ sur le langage σ si tous les énoncés de $\mathcal{T}$ sont vrais dans $\mathcal{X}$.*

Définition 9 (Théorie complète d'une structure)
Soit une σ-structure $\mathcal{X}$. On appelle **théorie complète** *de la structure $\mathcal{X}$ l'ensemble des énoncés du langage σ qui sont vrais dans $\mathcal{X}$.*

1.1.3 Profondeur de quantification

La notion de profondeur de quantification est fondamentale pour l'utilisation des destinées. Elle se définit par induction sur la structure de la formule, comme suit :

Définition 10 (Profondeur de quantification d'une formule)
Soit σ un langage. La **profondeur de quantification** *d'une formule F du premier ordre sur le langage σ, notée $q(F)$, est définie par :*
- *si F est atomique, $q(F) = 0$;*
- *si F est combinaison booléenne de formules $G_1, \ldots, G_k$, alors*

$$q(F) = \max(q(G_1), \ldots, q(G_k)) \ ;$$

- *si F est de la forme $\forall x G$ ou $\exists x G$, alors $q(F) = q(G) + 1$.*

1.2 Problème de décision des énoncés dans une structure

Nous allons nous intéresser à un problème très étudié en logique mathématique : celui de la décision des énoncés dans une structure.

Définition 11 (Théorie décidable)
Une théorie $\mathcal{T}$ est **décidable** *si l'ensemble des énoncés qui sont conséquences logiques de $\mathcal{T}$ est récursif.*

Dans les exemples qui sont présentés par la suite, on s'intéresse aux théories complètes de structures sur un langage relationnel fini.
Dans ce cadre précis, une telle théorie est décidable lorsque l'on dispose d'un algorithme qui prend en entrée un énoncé et répond à la question "Cet énoncé est-il vrai dans la structure ?". Certaines des structures que nous considérons ont des théories complètes décidables, d'autres non. Dans tous les cas, on s'intéresse à un sous-ensemble de ces théories complètes qui est toujours décidable : l'ensemble des énoncés de profondeur de quantification inférieure ou égale à un entier donné $p \geq 1$ vrais dans la structure.

Proposition 1
Soit $\mathcal{X}$ une structure sur un langage σ relationnel fini. Soit $p \geq 1$ et $\mathcal{T}_p$ l'ensemble des énoncés de profondeur de quantification inférieure ou égale à p qui sont vrais dans $\mathcal{X}$. Cet ensemble d'énoncés est décidable.

Preuve : La preuve repose sur le fait suivant : à équivalence logique près, il n'existe qu'un nombre fini d'énoncés de profondeur de quantification inférieure ou égale à p, et donc un nombre fini d'entre eux seulement sont vrais dans $\mathcal{X}$.

En effet, pour une profondeur de quantification donnée, disons p, il n'y a qu'un nombre fini de formules atomiques faisant intervenir les prédicats du langage et les variables libres $x_1, \ldots, x_p$. Il n'y a également qu'un nombre fini de façons de combiner ces formules avec les opérateurs $\neg, \wedge, \vee$, et de les quantifier sans répétition.

Pour comparer un énoncé quelconque aux énoncés de la liste finie d'énoncés vrais dans $\mathcal{X}$ qui se présentent sous la forme ci-dessus (variables nommées $x_1, \ldots, x_p$), on fait subir quelques transformations récursives aux énoncés pris en entrée de l'algorithme :

- on renomme les variables,
- on ordonne les formules atomiques,
- on élimine les répétitions.

L'algorithme de décision (qui dépend bien évidemment de la structure) consiste à comparer l'énoncé pris en entrée, après transformation, aux énoncés de la liste finie des énoncés vrais dans $\mathcal{X}$ établie auparavant.

$\square$

Il reste le problème suivant : on sait que cet algorithme existe, mais on ne sait pas forcément l'exhiber, c'est-à-dire décrire la liste des énoncés qui sont vrais dans $\mathcal{X}$. C'est ici que les destinées de Nézondet interviennent, car elles fournissent explicitement un autre algorithme de décision pour de tels ensembles d'énoncés, comme nous le verrons dans le Chapitre 4.

Chapitre 2

Une définition des destinées d'une structure

Une destinée est une structure arborescente, il est donc indispensable de procéder à quelques rappels de vocabulaire sur les arbres. En effet, nous utiliserons abondamment les notions décrites au Paragraphe 2.1, dont on peut trouver une présentation plus détaillée dans [Ber85] ou [GM79].

On définit ensuite ce qu'est une destinée (notion introduite par Francis Nézondet dans [Néz97]) et notamment une destinée liée à une structure. Enfin, nous présentons un exemple simple de destinée d'une structure arithmétique.

2.1 Vocabulaire sur les arbres

2.1.1 Arbre, nœud, branche

Définition 12 (Arbre)
Un **arbre** *est un triplet* (U, P, r) *où :*
– U *est un ensemble ;*
– P *est une relation binaire antiréflexive et antisymmétrique sur* U*, appelée* **relation de paternité** *telle que :*

1. *si* $P(x, y)$*, on dit que* x *est le* **père** *de* y*, ou que* y *est le* **fils** *de* x *;*
2. *il n'y a pas de cycle pour cette relation ;*
3. *tout élément de* U *sauf* r *a un unique père ;*

– *l'élément* $r \in U$*, appelé* **racine** *de l'arbre, est tel que pour tout* x *dans* U*, on a :* $\neg P(x, r)$ *(la racine n'a pas de père).*

Définition 13 (Nœud, feuille)
Un **nœud** *x dans un arbre (U, P, r) est un élément de U. Si x est tel que pour tout $y \in U$, on a $\neg P(x, y)$, le nœud x est appelé une* **feuille** *de l'arbre (une feuille n'a pas de fils).*

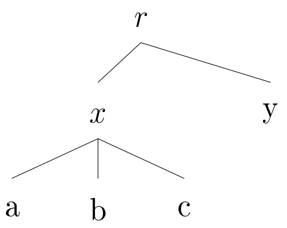


FIG. 2.1 – Un exemple d'arbre.

Définition 14 (Étiquetage)
Soit (U, P, r) un arbre et D un ensemble, on appelle **étiquetage** *de cet arbre une application l qui à chaque nœud de l'arbre associe un élement de D. Si x est un nœud, on appelle $l(x)$ l'***étiquette** *de x.*

Définition 15 (Branche)
Soit (U, P, r) un arbre. On appelle **branche** *de cet arbre tout ensemble de nœuds de l'arbre $\{x_0, \ldots, x_k\}$ tel que :*
- *$x_0 = r$;*
- *Pour tout $i \in \{0, \ldots, k-1\}$ on a : $P(x_i, x_{i+1})$;*
- *x_k est une feuille.*

La **longueur** *de la branche est le nombre de nœuds constituant la branche, c'est-à-dire $k + 1$.*

Sur l'exemple de la Figure 2.2, les éléments encadrés constituent une branche.

Définition 16 (Ascendants d'un nœud)
Soit (U, P, r) un arbre et $x \in U$ un nœud de l'arbre. On définit les **ascendants** *de x comme étant l'ensemble des nœuds qui sont entre x et la racine de l'arbre (nœud x exclu mais racine comprise).*

Sur la Figure 2.2, les ascendants du nœud b sont x et r.

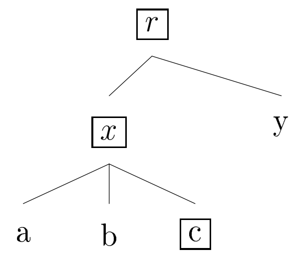


FIG. 2.2 – Branche de r à c.

2.1.2 Sous-arbre d'un nœud

Définition 17 (Sous-arbre d'un nœud)
Soit (U, P, r) un arbre et $x \in U$ un nœud de l'arbre. On définit le **sous-arbre** *de x comme étant la réunion des branches qui passent par x, privée des ascendants de x. On note $ST(x)$ le sous-arbre de x. C'est un arbre de racine x pour la relation de paternité induite.*

Sur l'exemple de la Figure 2.3, les éléments encadrés constituent le sous-arbre de x.

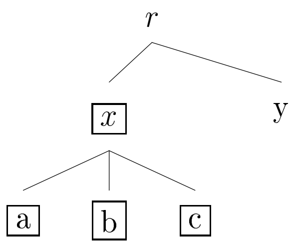


FIG. 2.3 – Sous-arbre de x.

2.1.3 Rang et degré d'un nœud

Définition 18 (Rang d'un nœud)
Soit (U, P, r) un arbre et $x \in U$ un nœud de l'arbre. On définit le **rang** *de x comme étant le nombre de nœuds entre x et la racine. La racine est de rang 0.*

Définition 19 (Hauteur d'un arbre)
Un arbre est dit de **hauteur finie** *s'il existe un entier K tel que tous les nœuds de l'arbre sont de rang inférieur à K. Dans ce cas, la* **hauteur** *de l'arbre est le maximum des longueurs des branches de l'arbre.*

Définition 20 (Degré d'un nœud)
Soit (U, P, r) un arbre et $x \in U$ un nœud de l'arbre. Le **degré** *de x, noté $Deg(x)$, est le nombre de fils de x (ce nombre est éventuellement infini). Le nœud x est dit de* **degré fini** *si $Deg(x)$ est fini.*

Définition 21 (Degré d'un arbre)
Un arbre est dit de **degré fini** *si tous ses nœuds sont de degré fini. Le degré de l'arbre est alors le maximum des degrés des nœuds.*

Sur la Figure 2.4, le nœud x est de rang 1 et de degré 3. L'arbre est de hauteur finie 3 et de degré fini 3.

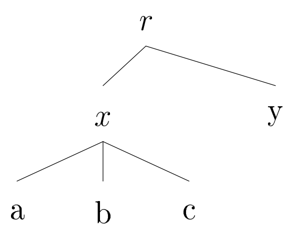


FIG. 2.4 – Rang et degré dans un arbre de hauteur 3 : x est de rang 1 et de degré 3.

Définition 22 (Rang d'un sous-arbre)
On appelera **rang d'un sous-arbre** *le rang de sa racine.*

2.2 Définition d'une destinée

2.2.1 Langage et interprétation étendus

Soit σ un langage relationnel fini quelconque. Au langage σ on associe un langage σ' étendu tel que $\sigma' = \sigma \cup \{P, \emptyset\}$, où P est une relation binaire et $\emptyset$ une constante.

Définition 23 (Destinée)
Soit $p \geq 1$. On appelle **destinée de hauteur** p *, ou* p**-destinée***, sur le langage σ, une σ'-structure telle que :*
- *si $c_1, \ldots, c_n$ sont les symboles de constante du langage σ, le domaine de la p-destinée est de la forme $U \cup \{a_1 \ldots, a_n\}$;*
- *la constante $\emptyset$ du langage σ' est interprétée par un élément de U que nous noterons $\emptyset$ également ;*
- *$(U, P, \emptyset)$ est un arbre dont toutes les branches sont de longueur $p + 1$;*
- *les constantes c_i sont interprétées respectivement par les éléments a_i du domaine.*

Une destinée est donc une structure sur un langage contenant la relation de paternité, et dont le domaine (privé des constantes) constitue, pour cette relation de paternité, un arbre dont les branches sont toutes d'une même longueur $p + 1$. Dans cette définition très générale, les interprétations des relations du langage sont quelconques. La notion de destinée devient particulièrement intéressante lorsque U est étiqueté par les éléments du domaine d'une σ-structure.

Définition 24 (Sous-arbre de destinée)
Soit T une p-destinée sur le langage σ. Soit x un nœud de T. Le sous-arbre de destinée de racine x est la sous-structure de T (sur le langage σ') qui a pour domaine $ST(x) \cup \{a_1, \ldots, a_n\}$. On notera également $ST(x)$ cette sous-structure.

2.2.2 Destinée d'une structure

Soient σ un langage relationnel fini, et $\mathcal{X}$ une σ-structure de domaine X.

Définition 25 (Destinée d'une structure $\mathcal{X}$)
Soit $p \geq 1$. Une p-destinée T sur le domaine $U \cup \{a_1, \ldots a_n\}$ et le langage σ (comportant n symboles de constante) est une **destinée de hauteur** p **de** $\mathcal{X}$*, ou* p**-destinée de** $\mathcal{X}$*, si :*
- *l'arbre $(U, P, \emptyset)$ est étiqueté par X ;*
- *on étend la fonction d'étiquetage aux constantes en posant $l(c^T) = c^{\mathcal{X}}$;*
- *pour toute branche B de T, et pour toute relation R de σ d'arité k et tout ensemble $\{x_1, \ldots, x_k\}$ constitué de nœuds de B de rang au moins 1 (on ne considère pas la racine $\emptyset$) et de symboles de constantes, on a :*

$$T \models R^T(x_1, \ldots, x_k) \text{ si et seulement si } \mathcal{X} \models R^{\mathcal{X}}(l(x_1), \ldots, l(x_k)),$$

où l est la fonction d'étiquetage.

Cette définition fixe les interprétations des relations du langage σ dans la destinée (les symboles de constantes et le prédicat P sont déjà fixés dans la définition d'une destinée).

Remarque 1 *Si on considère une p-destinée T quelconque sur le langage $\sigma' = \sigma \cup \{P, \emptyset\}$, où σ est un langage relationnel fini quelconque, il n'existe pas forcément de σ-structure $\mathcal{X}$ dont T soit une p-destinée. En effet, si R est une relation d'arité k, on peut avoir deux branches B et B' de T telles qu'il existe sur B et B' respectivement des k-uples de nœuds $x_1, \ldots, x_k$ et $x'_1, \ldots, x'_k$ tels que :*
- *pour tout $i \in \{1, \ldots, k\}$, on a $l(x_i) = l(x'_i)$;*
- *$T \models R^T(x_1, \ldots, x_k)$;*
- *$T \models \neg R^T(x'_1, \ldots, x'_k)$.*

Définition 26 (Destinée d'un n-uple de la structure $\mathcal{X}$)
Soit $p \geq 1$ et $n \geq 1$. On appelle p-destinée du n-uple d'éléments $(a_1, \ldots, a_n)$ de $\mathcal{X}$ toute p-destinée de la structure enrichie $(\mathcal{X}, a_1, \ldots, a_n)$ (les éléments a_i apparaissent comme des constantes ajoutées au langage σ).

2.2.3 Un exemple

On considère la structure $\langle \mathbb{N}, \perp, = \rangle$, où $=$ est l'égalité et $\perp$ la relation binaire "être premiers entre eux". On voit sur la Figure 2.5 une 2-destinée de cette structure. La relation d'égalité n'est pas représentée parmi les relations d'un nœud avec lui-même, en revanche, figure à côté du nœud la relation $\perp$ ou $\not\perp$ du nœud avec lui-même.

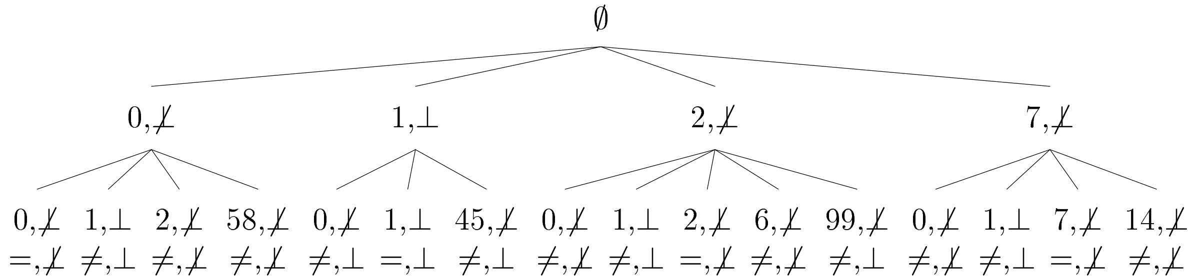


FIG. 2.5 – Une 2-destinée de la structure $\langle \mathbb{N}, \perp, = \rangle$.

On peut remarquer les choses suivantes sur cette figure :
- certaines feuilles ont, vis-à-vis de leur père, les mêmes relations qui sont vérifiées, comme c'est le cas de la feuille 0 et la feuille 14 dans le sous-arbre de 7, ou bien de la feuille 0 et la feuille 6 dans le sous-arbre de 2 ;
- il y a des situations non représentées dans la 2-destinée qui sont présentes dans la structure, par exemple, on pourrait rajouter la feuille 2 au sous-arbre de 7, pour représenter un élément premier avec 7 mais non premier avec lui-même.

On veut que les destinées sur lesquelles on va travailler soient à la fois significatives de tout ce qu'il se passe dans la structure pour une profondeur de quantification donnée, et également que ce soit des objets simplifiés au maximum. Nous allons par conséquent préciser dans le chapitre suivant ces deux notions de destinée *exhaustive* (il ne "manque" pas de branche) et de destinée *essentielle* (il n'y a pas de redondance entre plusieurs sous-arbres).

Chapitre 3

Isomorphismes dans les destinées

Maintenant que l'on dispose de l'objet "destinée", il est nécessaire de pouvoir comparer deux destinées entre elles, et de privilégier parmi les destinées d'une structures celles qui ont de bonnes propriétés (pas de redondance entre les sous-arbres, et pas de branche manquante).

On commence, dans le Paragraphe 3.1.1, par définir sur les destinées une notion d'isomorphisme entre sous-arbres de destinées. Cette notion correspond en fait, comme nous le verrons au Chapitre 5, à la notion de p-isomorphisme de Fraïssé entre structures.

Cette notion d'isomorphisme entre sous-arbres de destinées nous permet de définir au Paragraphe 3.1.3 l'*essentialité* d'une destinée (pas de redondances entre les sous-arbres).

Cette possibilité de comparer deux destinées fonde également la notion d'*exhaustivité* d'une destinée liée à une structure (il n'y a pas de branche manquante). C'est l'objet du Paragraphe 3.2.

On définit ensuite, au Paragraphe 3.3, deux mesures locales des tailles des étiquettes d'une destinée, à savoir : un majorant des étiquettes des fils d'un nœud et un majorant des étiquettes des descendants d'un nœud.

Enfin, on s'intéresse au Paragraphe 3.4 à la notion de réduction d'une destinée de certaines structures ayant de bonnes propriétés (structures normées, structures de domaine bien ordonné).

3.1 Isomorphisme entre sous-arbres de destinées

3.1.1 Définition

On définit la notion d'isomorphisme entre deux sous-arbres de même rang dans des destinées de même hauteur, par récurrence descendante sur le rang des sous-arbres.

Définition 27 (Isomorphisme entre sous-arbres)
Soit σ un langage relationnel fini. On fixe $p \geq 1$ et soient T et T' deux p-destinées sur le langage étendu σ'.

- *Deux feuilles u_p et u'_p de T et T' respectivement, et d'ascendants $u_{p-1}, \ldots, u_1, \emptyset$ et $u'_{p-1}, \ldots, u'_1, \emptyset$ respectivement sont isomorphes si les p-uples $(u_p, \ldots, u_1)$ et $(u'_p, \ldots u'_1)$ vérifient les mêmes formules atomiques et négatomiques sur le langage σ' ;*
- *Deux sous-arbres de T et T' de racines respectives u_k et u'_k de rang $k < p$, et d'ascendants respectifs $u_{k-1}, \ldots, u_1, \emptyset$ et $u'_{k-1}, \ldots, u'_1, \emptyset$ sont isomorphes si :*
  1. *Les k-uples $(u_k, \ldots, u_1)$ et $(u'_k, \ldots u'_1)$ vérifient les mêmes formules atomiques et négatomiques sur le langage σ' ;*
  2. *Si l'ensemble des fils de u_k est la famille $(v_i)_{i \in I}$ et l'ensemble des fils de u'_k est la famille $(v'_j)_{j \in J}$, on a :*
     - *Pour tout $i \in I$, il existe un $j \in J$ tel que $ST(v_i)$ et $ST(v'_j)$ sont isomorphes ;*
     - *Pour tout $j \in J$, il existe un $i \in I$ tel que $ST(v_i)$ et $ST(v'_j)$ sont isomorphes ;*

3.1.2 Exemple

Dans la Figure 3.1, les feuilles 0 et 6 dans le sous-arbre de 2 sont isomorphes. Les sous-arbres de 0 et 7 sont isomorphes, on peut faire correspondre la feuille 0 du sous-arbre de 0 à la feuille 7 du sous-arbre de 7, la feuille 1 à la feuille 1, la feuille 2 du sous-arbre de 0 à la feuille 0 du sous-arbre de 2, et la feuille 58 à la feuille 14.

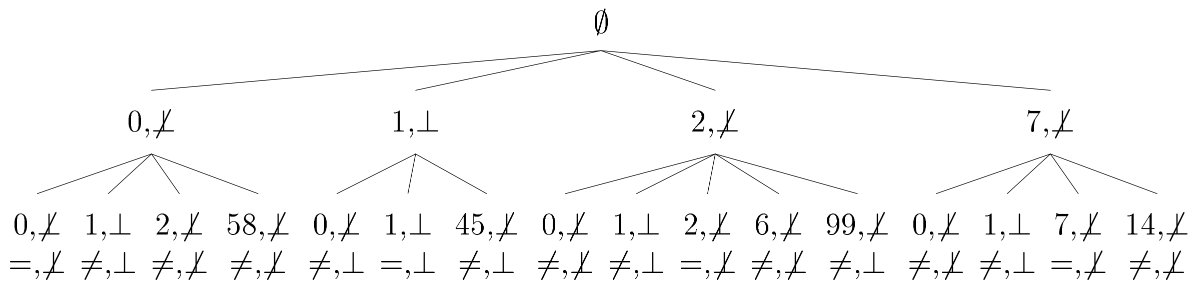


FIG. 3.1 – Une 2-destinée de la structure $\langle \mathbb{N}, \perp, = \rangle$.

Remarque 2 *Deux sous-arbres peuvent être isomorphes sans que leurs racines aient le même nombre de fils. Par exemple, si on supprime la feuille 14 du sous-arbre de 7, la feuille 58 du sous-arbre de 0 peut se voir associée la feuille 0 du sous-arbre de 7, et les deux sous-arbres de 0 et 7 restent isomorphes. En revanche, il est nécessaire que les deux racines aient pour fils des représentants des mêmes classes d'isomorphisme.*

3.1.3 Notion d'essentialité

Grâce à la notion d'isomorphisme entre les sous-arbres que l'on vient de définir, on peut simplifier une destinée en ne sélectionnant, pour chaque classe d'isomorphisme, qu'un seul représentant. Cette simplification, détaillée ci-dessous, produit une destinée sans redondance entre ses sous-arbres, que nous appelerons une destinée *essentielle*.

Le processus de simplification d'une destinée s'exécute par induction sur la structure de la destinée, à commencer par les feuilles. Soit $p \geq 1$ et T une p-destinée.
- Pour chaque sous-arbre de rang $p-1$ (les fils de la racine sont tous des feuilles), on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les feuilles.
- Pour chaque sous-arbre de rang $p-2$ (les fils sont racines de sous-arbres de rang $p-1$), on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les sous-arbres de rang $p-1$.
- ...
- Pour chaque sous-arbre de rang $k < p$ (les fils sont racines de sous-arbres de rang $k+1$), on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les sous-arbres de rang $k+1$.
- ...
- Dans la p-destinée, on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les sous-arbres de rang 1.

Définition 28 (Destinée essentielle)
On appelle **destinée essentielle** *toute destinée qui est invariante par la simplification décrite ci-dessus.*

Sur l'exemple du Paragraphe 3.1.2, le processus de simplification est décrit, étape par étape, dans les Figures 3.2, 3.3 et 3.4.

On commence par simplifier les feuilles dans les sous-arbres de rang 1 (ici $p = 2$). Dans le sous-arbre de 0, les feuilles 2 et 58 sont isomorphes, on n'en garde qu'une seule, par exemple 2. Dans le sous-arbre de 1, les feuilles 0 et 45 sont isomorphes, on ne garde que la feuille 0. Dans le sous-arbre de 2, les feuilles 0 et 6 sont isomorphes, on ne garde que la feuille 0. Dans le sous-arbre de 7, les feuilles 0, et 14 sont isomorphes, on ne garde que la feuille 0.

On simplifie ensuite la 2-destinée en ne conservant qu'un seul représentant par classe d'isomorphisme de sous-arbres de rang 1. Les sous-arbres de 0 et de 7 sont isomorphes, on ne garde donc que le sous-arbre de 0.

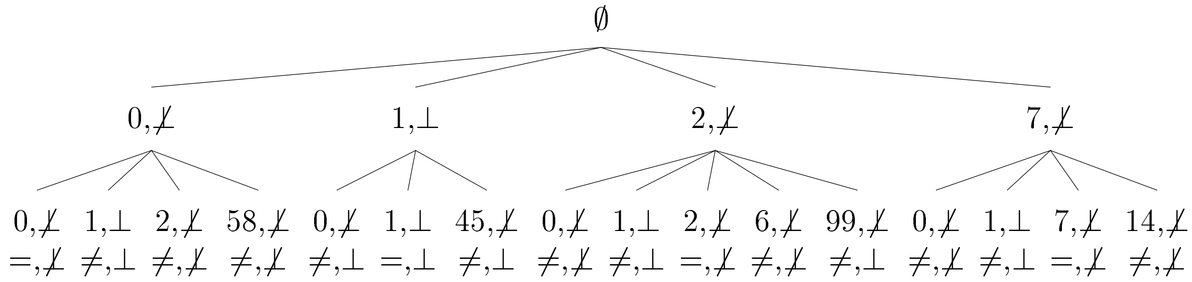


Fig. 3.2 – 2-destinée à simplifier.

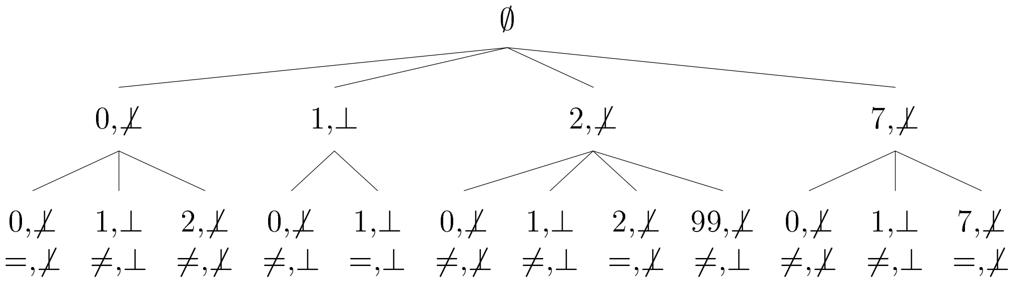


Fig. 3.3 – Simplification des feuilles.

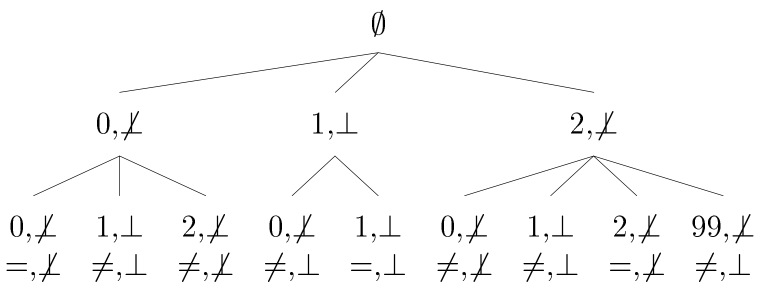


Fig. 3.4 – Simplification des sous-arbres de rang 1.

Remarque 3 *Si la feuille "manquante" de 7 (la feuille qui dit qu'il y a un élément premier avec 7 et non premier avec lui-même) avait été présente, le sous-arbre de 7 aurait été isomorphe au sous-arbre de 2 et non pas au sous-arbre de 0, ce qui aurait changé considérablement la signification du sous-arbre de 7.*

Ce problème est résolu si on s'arrange pour que la destinée à simplifier ne comporte pas de branche manquante, c'est le sujet du Paragraphe 3.2.

3.2 Exhaustivité d'une destinée liée à une structure

On formalise à présent la notion de branche "manquante". C'est une notion qui n'intervient que lorsque l'on a affaire à une destinée d'une structure, et que l'on compare les situations présentes dans la structure et celles qui sont présentes dans la destinée. Pour pouvoir dire qu'une branche "manque", il faut en effet pouvoir dire qu'il manque quelque chose par rapport à ce qu'il se passe dans la structure. Ce n'était pas le cas de la notion d'essentialité, qui est indépendante de la structure sous-jacente. Nous verrons au Chapitre 5 que les "situations présentes" que l'on veut voir apparaître dans les p-destinées d'une structure correspondent aux types de p-isomorphisme (au sens Fraïsséen).

3.2.1 Définition

Pour définir l'absence de "manque", on utilise une destinée de référence, dont il est sûr qu'elle comporte toutes les situations présentes dans la structure. Cette destinée est appelé *destinée complète* de la structure, et se définit comme suit :

Définition 29 (Destinée complète d'une structure)
Soit $\mathcal{X}$ une structure sur un langage relationnel fini σ, et $p \geq 1$. La p-destinée **complète** *de $\mathcal{X}$ est la p-destinée de $\mathcal{X}$ telle que chaque nœud de rang $k < p$ a pour étiquette de ses fils tous les éléments de la structure.*

Remarque 4 *Si la structure est infinie, chaque nœud, sauf les feuilles, est de degré infini. En revanche, si la structure est finie, cette destinée complète reste un objet fini.*

Définition 30 (Destinée exhaustive d'une structure)
Soit $\mathcal{X}$ une structure sur un langage relationnel fini σ, et $p \geq 1$. Une p-destinée T de $\mathcal{X}$ est **exhaustive** *si elle est isomorphe à la p-destinée complète de $\mathcal{X}$.*

Remarque 5 *En particulier, la p-destinée complète d'une structure est une p-destinée exhaustive. Ce sera le point de départ de nombre de constructions de destinées exhaustives que nous ferons par la suite.*

3.2.2 Exemple

Reprenons l'exemple du paragraphe 3.1.2. On suppose que la structure sous-jacente est la structure $\langle \mathbb{N}, \perp, = \rangle$, qui est de domaine infini. La 2-destinée complète de cette structure est impossible à dessiner entièrement, mais se présente comme sur la Figure 3.5.

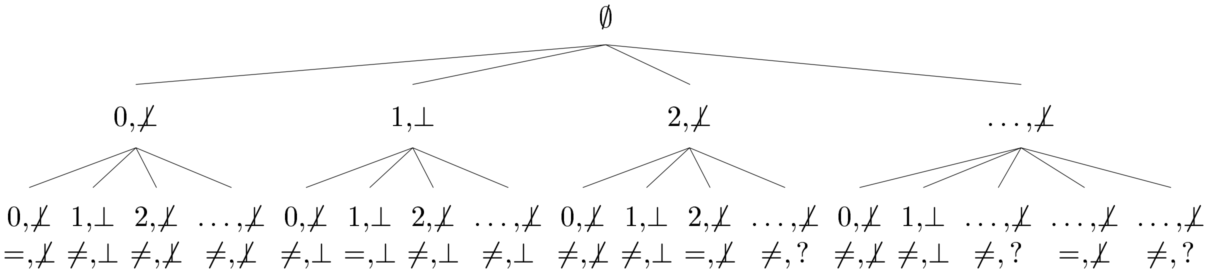


FIG. 3.5 – La 2-destinée complète de la structure $\langle \mathbb{N}, \perp, = \rangle$.

3.2.3 Destinée exhaustive et essentielle

Si on groupe maintenant les deux notions d'exhaustivité et d'essentialité, on obtient une destinée exhaustive essentielle, qui est un objet fini : il n'y a qu'un nombre fini de classes d'isomorphisme d'un sous arbre donné (on s'en convainc en faisant la liste des relations possible d'un nœud avec ses ancêtres : il n'y en a qu'un nombre fini), et la destinée est essentielle, donc elle ne comporte pour chaque classe d'isomorphisme de sous-arbre qu'un seul représentant. Cet objet est également totalement informateur sur la structure au rang de quantification p, car exhaustif.

Notre objet d'étude privilégié sera par conséquent les destinées essentielles exhaustives d'une structure donnée. Pour obtenir une p-destinée essentielle et exhaustive d'une structure donnée, une solution simple mais longue consiste à partir de la p-destinée complète et à effectuer un processus de simplifications successives comme décrit au Paragraphe 3.1. Il reste à choisir quel représentant de chaque classe on va garder, ce dont nous discutons au Paragraphe 3.4.

Remarque 6 *Deux p-destinées exhaustives et essentielles d'une même structure sont non seulement isomorphes au sens défini ci-dessus, mais ont également la même structure d'arbre* (*chaque nœud d'une destinée a exactement le même nombre de fils que l'unique nœud qui lui correspond par isomorphisme dans l'autre destinée*).

Exemple : Si on effectue des simplifications successives sur la 2-destinée complète de la Figure 3.5, cela nous donne les résultats des Figures 3.6 et 3.7.

Quand on simplifie les feuilles dans chaque sous-arbre, cela nous donne :

- Dans le sous-arbre de 0 : un entier, par rapport à 0 est :
  - soit égal à 0 (et nécessairement non premier avec lui-même et 0)

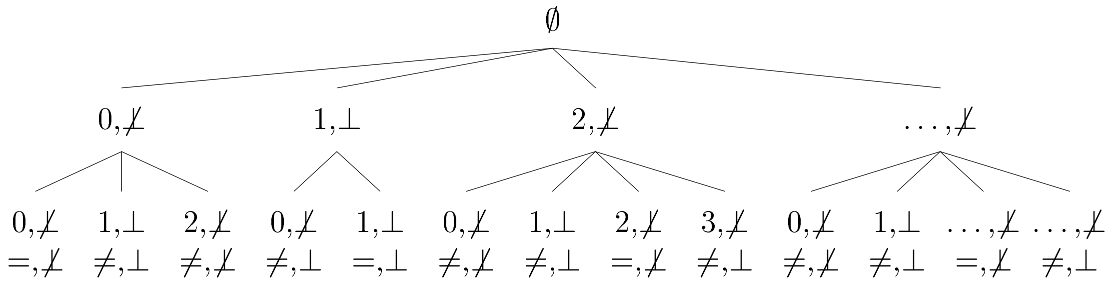


FIG. 3.6 – Simplification des feuilles dans la 2-destinée complète de $\langle \mathbb{N}, \perp, = \rangle$.

  - soit égal à 1 (et nécessairement premier avec lui-même et avec 0)
  - soit supérieur ou égal à 2 (et nécessairement non premier avec lui-même et avec 0)
- Dans le sous-arbre de 1 : un entier, par rapport à 1, est :
  - soit égal à 1 (et nécessairement premier avec lui-même et avec 1)
  - soit différent de 1 (et nécessairement non premier avec lui-même et premier avec 1)
- Dans le sous-arbre d'un entier $n \geq 2$: un entier, par rapport à n est :
  - soit égal à n (et nécessairement non premier avec lui-même et n)
  - soit égal à 1 (et nécessairement premier avec lui-même et avec n)
  - soit différent de n et 1 et premier avec n (et nécessairement non premier avec lui-même) (par exemple $n+1$)
  - soit différent de n et 1 et non premier avec n (et nécessairement non premier avec lui-même) (par exemple 0)

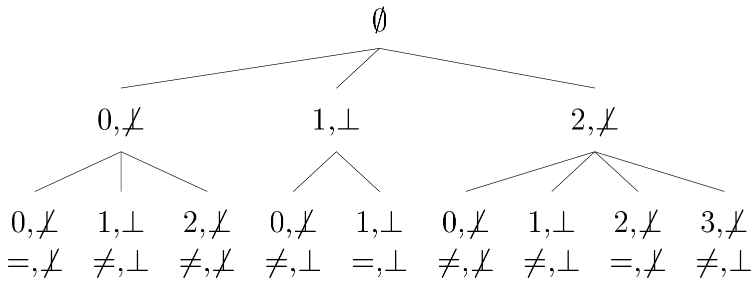


FIG. 3.7 – Simplification au rang 1 dans la 2-destinée complète de $\langle \mathbb{N}, \perp, = \rangle$.

Quand on simplifie les sous-arbres de rang 1, on est obligé de garder les sous-arbres de 0 et 1 qui jouent un rôle particulier (en effet, 0 est non premier avec tout le monde sauf 1, et 1 est premier avec lui-même, et c'est le seul entier ayant cette propriété), en revanche les sous-arbres de tous les entiers supérieurs ou égaux à 2 sont isomorphes, on peut donc garder uniquement le sous-arbre de 2.

3.3 Bornes sur les nœuds d'une destinée d'une structure normée

Dans ce paragraphe, on se restreint au cas de certaines structures, appelées structures normées. On rappelle la notion de bon ordre sur un ensemble :

Définition 31 (Bon ordre)
Soit A un ensemble ordonné par une relation R. On dit que R est une relation de **bon ordre** *si toute partie non vide de A admet un plus petit élement pour l'ordre R. On dit alors que l'ensemble A est* **bien ordonné***.*

Remarque 7 *Un bon ordre est total.*

Définition 32 (Structure normée)
Soit $\mathcal{X}$ une structure sur un langage relationnel fini σ. Une **norme** *sur cette structure est une application $||.||$ du domaine X de la structure $\mathcal{X}$ vers un ensemble bien ordonné (par exemple $\mathbb{N}$). La* **structure normée** *associée est la donnée du couple $(\mathcal{X}, ||.||)$.*

Dans cette thèse, nous considérons le plus souvent des structures normées, avec une norme à valeur dans $\mathbb{N}$. La norme sur une structure permet de déterminer la taille d'un sous-arbre : on connaît la hauteur d'un sous-arbre dans une p-destinée, mais il faut pouvoir également déterminer quelles étiquettes sont susceptibles d'apparaître dans un sous-arbre donné. On définit pour cela deux bornes décrites ci-après.

3.3.1 Borne sur les fils d'un nœud

Définition 33 (Borne sur les fils d'un nœud)
Soit $(\mathcal{X}, ||.||)$ une σ-structure normée. Soit $p \geq 1$ et T une p-destinée de $\mathcal{X}$. Soit x un nœud de T de rang $k < p$. Soit $(u_i)_{i \in I}$ la famille des fils de x. On définit la **borne sur les fils de** *x, notée $Sup_f(x)$, de la façon suivante :*

$$Sup_f(x) = \sup_{i \in I}(||l(u_i)||).$$

Remarque 8 *Cette borne peut éventuellement avoir une valeur infinie, si la destinée est infinie.*

Exemple : On considère la structure $\langle \mathbb{N}, \leq \rangle$, normée par l'identité sur $\mathbb{N}$. Sur la 2-destinée de la Figure 3.8, on a :

- $Sup_f(\emptyset) = 4$
  - $Sup_f(0, \emptyset) = 1$
  - $Sup_f(1, \emptyset) = 42$
  - $Sup_f(4, \emptyset) = 12$

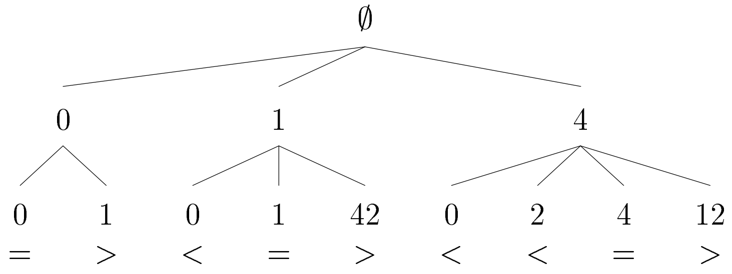


FIG. 3.8 – Une 2-destinée de $\langle \mathbb{N}, \leq \rangle$.

3.3.2 Borne sur les descendants d'un nœud

La borne sur les fils d'un nœud rend compte de ce qui se passe au rang supérieur, mais on a parfois besoin de connaître l'extension de la destinée sur tout un sous-arbre et plus seulement sur les fils d'un nœud.

Définition 34 (Borne sur les descendants d'un nœud)
Soit $(\mathcal{X}, ||.||)$ *une* σ*-structure normée. Soit* $p \geq 1$ *et* T *une* p*-destinée de* $\mathcal{X}$*. Soit* x *un nœud de rang* $k \leq p$ *de* T*. Soit* $(v_j)_{j \in J}$ *la famille des descendants de* x *(tous les nœuds du sous-arbre de* x *qui sont de rang strictement supérieur à* k*). On définit la* **borne sur les descendants de** x*, notée* $Sup_d(x)$*, de la façon suivante :*

- *Si* J *n'est pas vide (le nœud* x *n'est pas une feuille) :*

$$Sup_d(x) = \sup_{j \in J}(||l(v_j)||).$$

- *Si* J *est vide (on a* $k = p$ *et* x *est une feuille) :*

$$Sup_d(x) = ||l(x)||.$$

Remarque 9 *Cette borne peut également avoir une valeur infinie, si la destinée est infinie.*

Remarque 10 *Cette borne est définie sur les feuilles, contrairement à la borne sur les fils d'un nœud.*

Exemple : On considère à nouveau la structure $\langle \mathbb{N}, < \rangle$, normée par l'identité sur $\mathbb{N}$. Sur la 2-destinée de la Figure 3.8, on a :

- $Sup_d(\emptyset) = 42$
  - $Sup_d(0, \emptyset) = 1$
    - $Sup_d(0, 0, \emptyset) = 0$
    - $Sup_d(1, 0, \emptyset) = 1$
  - $Sup_d(1, \emptyset) = 42$
    - $Sup_d(0, 1, \emptyset) = 0$
    - $Sup_d(1, 1, \emptyset) = 1$
    - $Sup_d(42, 1, \emptyset) = 42$
  - $Sup_d(4, \emptyset) = 12$
    - $Sup_d(0, 4, \emptyset) = 0$
    - $Sup_d(2, 4, \emptyset) = 2$
    - $Sup_d(4, 4, \emptyset) = 4$
    - $Sup_d(12, 4, \emptyset) = 12$

Une estimation des bornes sur les fils et sur les descendants d'un nœud sera primordiale lorsque l'on étudiera, en Partie II, la construction des destinées d'une structure infinie.

3.4 Destinée réduite

Étant donnée une structure et une hauteur p, on dispose déjà d'une notion de p-destinée sur cette structure qui est satisfaisante au sens suivant :

- c'est un objet complet (exhaustivité) ;
- c'est un objet fini et compact (essentialité).

En revanche, on a laissé en suspens le problème du choix du représentant d'une classe d'isomorphisme de sous-arbres donnée. Si ce choix est judicieux, la p-destinée peut devenir assez simple à décrire, notamment en ce qui concerne les bornes sur les fils et les bornes sur les descendants pour chaque nœud. La façon de choisir ce représentant fait l'objet de ce paragraphe.

3.4.1 Destinée réduite d'une structure normée

Dans une structure normée, le choix des représentants est dicté par la norme de la racine du sous-arbre : on choisit un représentant dans chaque classe de sous-arbres dont la racine a une étiquette de norme minimale.

Le processus de réduction d'une destinée complète s'exécute par induction sur la structure de la destinée, à commencer par les feuilles. Soit $p \geq 1$ et T la p-destinée complète d'une structure $\mathcal{X}$.

- Pour chaque sous-arbre de rang $p-1$ (les fils de la racine sont tous des feuilles), on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les feuilles et ce représentant a une étiquette de norme minimale.
- Pour chaque sous-arbre de rang $p-2$ (les fils sont racines de sous-arbres de rang $p-1$), on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur

les sous-arbres de rang $p-1$ et ce représentant a une racine dont l'étiquette est de norme minimale.
- ...
- Pour chaque sous-arbre de rang $k < p$ (les fils sont racines de sous-arbres de rang $k+1$), on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les sous-arbres de rang $k+1$ et ce représentant a une racine dont l'étiquette est de norme minimale.
- ...
- Dans la p-destinée, on ne garde qu'un seul représentant de chaque classe d'isomorphisme sur les sous-arbres de rang 1 et ce représentant a une racine dont l'étiquette est de norme minimale.

Définition 35 (Destinée réduite d'une structure normée)
Soit $(\mathcal{X}, ||.||)$ une σ-structure normée, et $p \geq 1$. On appelle **p-destinée réduite de $\mathcal{X}$** *toute p-destinée essentielle et exhaustive obtenue à partir de la p-destinée complète de $\mathcal{X}$ par le processus décrit ci-dessus.*

Remarque 11 *Sans précision supplémentaire sur la norme, il n'y a pas forcément unicité de la destinée réduite.*

Exemple : On considère la structure $\langle \mathbb{R}, \leq \rangle$, où $\leq$ est l'ordre total sur les réels, normée par l'application partie entière supérieure de la valeur absolue :

$$\begin{array}{rcl} ||.|| : \mathbb{R} & \to & \mathbb{N} \\ x & \mapsto & \lceil |x| \rceil \end{array}$$

La Figure 3.9 présente deux 2-destinées réduites de cette structure.

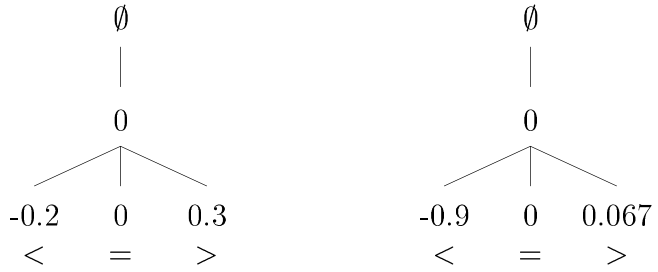


FIG. 3.9 – Deux 2-destinées réduites de $\langle \mathbb{R}, \leq \rangle$.

3.4.2 Destinée réduite d'une structure bien ordonnée

On suppose que $\mathcal{X}$ est une structure dont le domaine est bien ordonné, normé par l'identité. Le processus de réduction produit alors une p-destinée unique : partant de la p-destinée complète de $\mathcal{X}$, on réduit les feuilles de chaque sous-arbre de rang $p-1$, et pour chaque classe d'isomorphisme de ces feuilles dans un sous-arbre donné, on a une feuille d'étiquette minimale, qui est celle que l'on choisit. Il en est de même à chaque niveau de la réduction, et ainsi, l'objet final est unique.

Exemple : On considère une structure arithmétique sur le domaine $\mathbb{N}$. La norme choisie est l'identité. Alors la destinée réduite est unique. Sur la Figure 3.10 on peut voir la 2-destinée réduite de la structure $\langle \mathbb{N}, \leq, P \rangle$, où $\leq$ est l'ordre total sur les entiers et P le prédicat "être premier" (figuré par un encadrement des nœuds dont l'étiquette est un nombre premier).

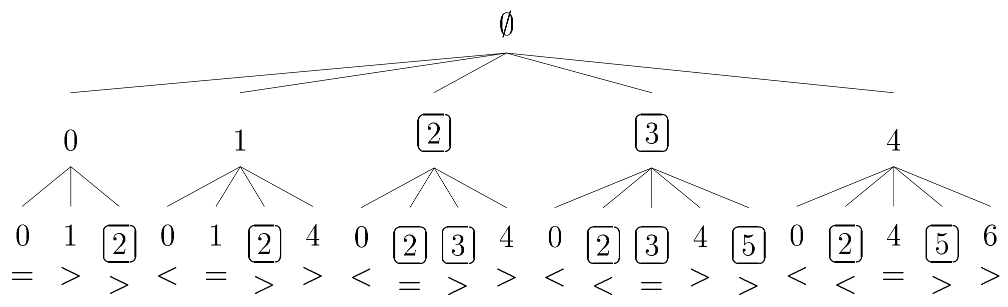


FIG. 3.10 – La 2-destinée réduite de la structure $\langle \mathbb{N}, \leq, P \rangle$.

Chapitre 4

Satisfaction des énoncés et algorithme de décision utilisant les destinées

Ce chapitre présente une application primordiale des destinées, à savoir renseigner sur la satisfaction des énoncés dans une structure. On dispose d'une σ-structure $\mathcal{X}$ (où σ est un langage relationnel fini), on se fixe la profondeur de quantification $p \geq 1$ des énoncés que l'on souhaite décider et on suppose que l'on dispose également d'une p-destinée exhaustive T de la structure $\mathcal{X}$. On commence par décrire la transposition d'une formule du langage σ vers une formule du langage σ', appelé forme destinale de la σ-formule. Cette mise sous forme destinale consiste en une relativisation des quantifications. La satisfaction d'un σ-énoncé dans la structure $\mathcal{X}$ est alors équivalente à la satisfaction de sa forme destinale dans la p-destinée T (Paragraphe 4.1).

Cette équivalence justifie le fait que l'on puisse utiliser les destinées comme outils de décision pour les énoncés de profondeur de quantification donnée. Nous présentons ensuite l'algorithme de décision, qui est une élimination des quantificateurs adaptée à la forme arborescente des destinées (Paragraphe 4.2), ainsi qu'un exemple d'exécution (Paragraphe 4.3), avant de nous intéresser à la complexité de cet algorithme de décision (Paragraphe 4.4).

4.1 Satisfaction d'une formule dans une destinée exhaustive d'une structure

Pour une σ-structure $\mathcal{X}$ donnée, et $p \geq 1$ fixé, on se donne T une p-destinée exhaustive de $\mathcal{X}$. La p-destinée T est une structure sur le langage $\sigma' = \sigma \cup \{P, \emptyset\}$ où le prédicat P est interprété par la relation de paternité entre deux nœuds de la destinée. Si F est un énoncé de σ, on a donc une notion de satisfaction de F dans T en tant que σ-structure, mais cela n'est pas suffisant. On a en effet construit les destinées de manière que les relations soient uniquement pertinentes entre nœuds d'une même branche (et éventuellement avec des constantes). On va donc relativiser les quantifications d'une profondeur donnée grâce au prédicat de paternité, de manière à ce que chaque sous-formule atomique de F ne soit évaluée que sur des étiquettes de nœuds d'une même branche. Pour cela, on passe par la transformation de l'énoncé F en sa "forme destinale", intégrant la relation P. On montre ensuite que cette transformation est adéquate pour transporter la satisfaction des énoncés de profondeur de quantification inférieure ou égale à p dans la σ-structure $\mathcal{X}$ (éventuellement infinie) vers la satisfaction dans la σ'-structure T (qui est finie si la destinée T est essentielle).

4.1.1 Transformation d'une σ-formule en une σ'-formule : forme destinale

Soit F une formule à k variables libres nommées $x_1, \ldots, x_k$ et de profondeur de quantification $p - k$. On suppose, sans perte de généralité, que F a pour noms de variables liées $x_{k+1}, \ldots, x_p$, quantifiées dans cet ordre.

On définit la transformation syntaxique "$\tilde{}$ " sur une telle formule de la façon suivante : pour j allant de $k+1$ à p :

- On remplace les sous-formules de F de la forme $\forall x_j G(x_1, \ldots, x_j)$ par :

$$\forall x_j (P(x_{j-1}, x_j) \to G(x_1, \ldots, x_j)) \ ;$$

- On remplace les sous-formules de F de la forme $\exists x_j G(x_1, \ldots, x_j)$ par :

$$\exists x_j (P(x_{j-1}, x_j) \wedge G(x_1, \ldots, x_j)) \ ;$$

Dans le cas où $k = 0$, on prend pour convention que la "variable x_0" est la constante $\emptyset$ du langage σ'. On obtient une σ'-formule $\tilde{F}$, à k variables libres $x_1, \ldots, x_k$ et de profondeur de quantification $p - k$.

Définition 36 (Forme destinale d'une formule)
Soit F une σ-formule à k variables libres $x_1, \ldots, x_k$ et de profondeur de quantification $p - k$. On appelle **forme destinale de** F *la σ'-formule*

$$\tilde{F}(x_1, \ldots, x_k).$$

Exemple : Soit $\sigma = \{C, R\}$ un langage relationnel fini où C est un prédicat unaire et R un prédicat binaire. La formule

$$R(x_1, x_2) \wedge \forall x_3 [\neg C(x_3) \vee \exists x_4 (R(x_3, x_4) \vee C(x_2))]$$

prend comme forme destinale la σ'-formule :

$$R(x_1, x_2) \wedge \forall x_3 (P(x_2, x_3) \to [\neg C(x_3) \vee \exists x_4 (P(x_3, x_4) \wedge (R(x_3, x_4) \vee C(x_2)))]).$$

Proposition 2
Soient F et G deux formules à k variables libres nommées $x_1, \ldots, x_k$ et de profondeur de quantification $p-k$. Alors on a (la relation $\equiv$ est l'équivalence logique) :
- $(F \tilde{\wedge} G) \equiv \tilde{F} \wedge \tilde{G}$
- $(F \tilde{\vee} G) \equiv \tilde{F} \vee \tilde{G}$
- $(\tilde{\neg F}) \equiv \neg\tilde{F}$

Preuve : Il est clair que $(F \tilde{\wedge} G) \equiv \tilde{F}\wedge\tilde{G}$ et $(F \tilde{\vee} G) \equiv \tilde{F}\vee\tilde{G}$. Pour la dernière équivalence logique, il suffit de remarquer que, comme on a $\neg(A \to B) \equiv A \wedge \neg B$:

$$\begin{array}{ccc} \neg(\forall x_j G(x_1,\ldots,x_j)) & \stackrel{\sim}{\longmapsto} & \neg(\forall x_j(P(x_{j-1},x_j) \to G(x_1,\ldots,x_j))) \\ \equiv & & \equiv \\ \exists x_j \neg G(x_1,\ldots,x_j) & \stackrel{\sim}{\longmapsto} & \exists x_j(P(x_{j-1},x_j) \wedge \neg G(x_1,\ldots,x_j)) \end{array}$$

et

$$\begin{array}{ccc} \neg(\exists x_j G(x_1,\ldots,x_j)) & \stackrel{\sim}{\longmapsto} & \neg(\exists x_j(P(x_{j-1},x_j) \wedge G(x_1,\ldots,x_j))) \\ \equiv & & \equiv \\ \forall x_j \neg G(x_1,\ldots,x_j) & \stackrel{\sim}{\longmapsto} & \forall x_j(P(x_{j-1},x_j) \to \neg G(x_1,\ldots,x_j)) \end{array}$$

$\square$

4.1.2 Equivalence entre la satisfaction d'une formule dans une structure et la satisfaction de sa forme destinale dans une destinée de la structure

On montre maintenant le résultat suivant :

Theorème 2
Soit σ un langage relationnel fini, $\mathcal{X}$ une σ-structure et F un énoncé de profondeur de quantification $p \geq 1$. Soit T une p-destinée exhaustive et essentielle de $\mathcal{X}$. On l'équivalence suivante :

$$\mathcal{X} \models F$$

si et seulement si

$$T \models \tilde{F}$$

où $\tilde{F}$ est la forme destinale de l'énoncé F.

Ce théorème est conséquence de la Proposition 3 suivante, plus générale :
Soit σ un langage relationnel fini, $\mathcal{X}$ une σ-structure et T une p-destinée exhaustive de $\mathcal{X}$. Soit F une formule à k variables libres $x_1, \ldots, x_k$ et de profondeur de quantification $p-k$. Soit $(a_1, \ldots, a_k)$ un k-uple d'éléments du domaine X de la structure $\mathcal{X}$. Il existe un nœud b_k d'ascendants $b_{k-1}, \ldots, b_1, \emptyset$ dans la p-destinée complète de $\mathcal{X}$ tel que pour tout $i \in \{1, \ldots, k\}$, on a $l(b_i) = a_i$. Le sous-arbre $ST(b_k)$ admet au moins un sous-arbre isomorphe dans la destinée T puisqu'elle est exhaustive.

Proposition 3
Pour tout nœud b'_k de T, d'ascendants $b'_{k-1}, \dots, b'_1, \emptyset$ dans T, tel que le sous-arbre $ST(b'_k)$ de T est isomorphe au sous-arbre $ST(b_k)$ de la p-destinée complète, on a :

$$\mathcal{X} \models F(a_1, \dots, a_k) \text{ si et seulement si } T \models \tilde{F}(b'_1, \dots, b'_k).$$

Preuve : (de la proposition)
On montre le résultat par récurrence descendante sur $k \in \{0, \dots, p\}$.

$\boxed{\Rightarrow}$

Cas $k = p$: Soit F à p variables libres $x_1, \dots, x_p$, donc sans quantificateur. La formule F est alors une combinaison booléenne de formules atomiques, et on a $\tilde{F} = F$ (pas de quantificateur, donc pas de transformation de la formule F). Soit $(a_1, \dots, a_p)$ un p-uple d'éléments de X, et b_p un nœud d'ascendants $b_{p-1}, \dots, b_1, \emptyset$ dans la p-destinée complète de $\mathcal{X}$, tel que pour tout $i \in \{1, \dots p\}$ on ait $l(b_i) = a_i$. Soit b'_p un nœud d'ascendants $b'_{p-1}, \dots, b'_1, \emptyset$ tel que le sous-arbre de ce nœud dans T est isomorphe au sous-arbre du nœud b_p. Le nœud b'_p existe puisque la destinée T est exhaustive. Par définition de l'isomorphisme entre les feuilles, les p-uples $(l(b_1), \dots, l(b_p))$ et $(l(b'_1), \dots, l(b'_p))$ satisfont exactement les mêmes formules atomiques, et donc on a $\mathcal{X} \models F(l(b'_1), \dots, l(b'_p))$. Par définition d'une destinée d'une structure, on a $T \models F(b'_1, \dots, b'_p)$, et donc $T \models \tilde{F}(b'_1, \dots, b'_p)$.

Montrons que l'hypothèse de récurrence pour $1 \leq k + 1 \leq p$ implique l'hypothèse de récurrence pour k.
Soit F une formule à k variables libres nommées $x_1, \dots, x_k$ et de profondeur de quantification $p - k$. Soit $(a_1, \dots, a_k)$ un k-uple d'éléments du domaine X de la structure $\mathcal{X}$. Soit b_k le nœud d'ascendants $b_{k-1}, \dots, b_1, \emptyset$ dans la p-destinée complète de $\mathcal{X}$ tel que pour tout $i \in \{1, \dots k\}$ on ait $l(b_i) = a_i$. Soit b'_k un nœud de T d'ascendants $b'_{k-1}, \dots, b'_1, \emptyset$ tel que le sous-arbre de ce nœud dans T est isomorphe au sous-arbre du nœud b_k dans la p-destinée complète. D'après la Proposition 2, on peut se restreindre à l'étude du cas où F est de la forme : $F(x_1, \dots, x_k) = Q_{k+1} x_{k+1} G(x_1, \dots, x_{k+1})$.

- Si $Q_{k+1} = \exists$: alors $\tilde{F}(x_1, \dots, x_k) = \exists x_{k+1}(P(x_k, x_{k+1}) \wedge \tilde{G}(x_1, \dots, x_{k+1}))$.

  On suppose que $\mathcal{X} \models F(a_1, \dots, a_k)$, donc $\mathcal{X} \models \exists x_{k+1} G(a_1, \dots, a_k, x_{k+1})$. Donc il existe un élément a_{k+1} dans X tel que $\mathcal{X} \models G(a_1, \dots, a_k, a_{k+1})$. Puisque les sous-arbres $ST(b_k)$ dans la p-destinée complète de $\mathcal{X}$ et $ST(b'_k)$ dans T sont isomorphes, il existe un fils b'_{k+1} du nœud b'_k tel que $ST(b_{k+1})$ dans la p-destinée complète de $\mathcal{X}$, avec $l(b_{k+1}) = a_{k+1}$, et $ST(b'_{k+1})$ dans T sont isomorphes. On applique l'hypothèse de récurrence à G.
  On a donc : $T \models \tilde{G}(b'_1, \dots, b'_{k+1})$. Comme b'_{k+1} est fils de b'_k, on a également : $T \models P(b'_k, b'_{k+1})$, donc :

  $$T \models P(b'_k, b'_{k+1}) \wedge \tilde{G}(b'_1, \dots, b'_{k+1}),$$

ce qui implique

$$T \models \exists x_{k+1}(P(b'_k, x_{k+1}) \wedge \tilde{G}(b'_1, \ldots, b'_k, x_{k+1})),$$

soit

$$T \models \tilde{F}(b'_1, \ldots, b'_k).$$

– Si $Q_{k+1} = \forall$: alors $\tilde{F}(x_1, \ldots, x_k) = \forall x_{k+1}(P(x_k, x_{k+1}) \to \tilde{G}(x_1, \ldots, x_{k+1}))$.

  On suppose que $\mathcal{X} \models F(a_1, \ldots, a_k)$, donc $\mathcal{X} \models \forall x_{k+1} G(a_1, \ldots, a_k, x_{k+1})$. Pour tout élément a_{k+1} dans X, on a donc $\mathcal{X} \models G(a_1, \ldots, a_k, a_{k+1})$.
  Pour chaque fils b'_{k+1} du nœud b'_k dans T, il existe un fils b_{k+1} du nœud b_k dans la destinée complète, tel que $ST(b_{k+1})$ dans la p-destinée complète de $\mathcal{X}$ et $ST(b'_{k+1})$ dans T sont isomorphes. On a : $\mathcal{X} \models G(a_1, \ldots, a_k, l(b_{k+1}))$.
  On applique l'hypothèse de récurrence à G. On a donc : $T \models \tilde{G}(b'_1, \ldots, b'_{k+1})$, et ce pour tout fils b'_{k+1} du nœud b'_k dans T. On peut donc écrire pour tout nœud b'_{k+1} de T :

$$T \models P(b'_k, b'_{k+1}) \to \tilde{G}(b'_1, \ldots, b'_{k+1}),$$

  ce qui implique

$$T \models \forall x_{k+1}(P(b'_k, x_{k+1}) \to \tilde{G}(b'_1, \ldots, b'_k, x_{k+1})),$$

  soit

$$T \models \tilde{F}(b'_1, \ldots, b'_k).$$

$\boxed{\Leftarrow}$ Analogue.

□

Preuve : (du théorème) On applique la Proposition 3 à l'énoncé F ($k = 0$).

□

4.2 Description de l'algorithme

Grâce au Théorème 2, on peut décider les énoncés de profondeur de quantification $p \geq 1$ sur une structure $\mathcal{X}$ en passant par une p-destinée exhaustive de $\mathcal{X}$. On exige de plus que la destinée soit essentielle, afin de manipuler un objet fini (et de taille minimale en l'occurrence). Il s'agit, pour un énoncé F de profondeur de quantification p donnée, d'évaluer sa forme destinale sur l'objet fini qu'est la p-destinée exhaustive essentielle choisie. Cet algorithme de décision est décrit dans [Cha01] et [Cha02], ainsi que sur un exemple dans [Néz97].

Le principe de l'algorithme est le suivant : étant donnée une p-destinée exhaustive et essentielle T de la structure $\mathcal{X}$, et un énoncé F de profondeur de quantification p, on évalue l'énoncé $\tilde{F}$ sur T, ce qui revient à exécuter un processus d'analyse de la formule F et à lire sur la destinée la satisfaction ou la non satisfaction de $\tilde{F}$.

On adopte pour cet algorithme la structure de données suivante : la p-destinée exhaustive (et essentielle) se présente sous la forme d'un arbre, étiqueté par des éléments de la structure, et chaque nœud s'accompagne de la liste des relations satisfaites par le nœud et ses ascendants. L'évaluation d'une formule atomique concernant un nœud et ses ascendants correspond alors à une lecture dans cette liste. La longueur de la liste dépend des relations du langage σ et de leurs arités, ainsi que du rang du nœud considéré.

L'algorithme consiste en une traduction "en direct" de l'énoncé F en sa forme destinale $\tilde{F}$ afin d'évaluer cette dernière sur la destinée T. Cela consiste donc en une élimination des quantificateurs en remplaçant chaque quantification par un nombre fini d'instanciations des variables :

Entrée : un énoncé F de profondeur de quantification p
Sortie : la réponse à la question "F est-il vrai dans $\mathcal{X}$?"
Hypothèse : on connaît T une p-destinée exhaustive (et essentielle) de $\mathcal{X}$

1. Le nœud courant est initialisé à $\emptyset$.
2. Si F est de la forme $G \wedge H$, alors $\tilde{F} = \tilde{G} \wedge \tilde{H}$: évaluer $\tilde{G}$ et $\tilde{H}$ dans la destinée T et renvoyer `VRAI` si $\tilde{G}$ et $\tilde{H}$ sont vrais dans T, et `FAUX` sinon.
3. Si F est de la forme $G \vee H$, alors $\tilde{F} = \tilde{G} \vee \tilde{H}$: $\tilde{G}$ et $\tilde{H}$ dans la destinée T et renvoyer `VRAI` si $\tilde{G}$ ou $\tilde{H}$ est vrai dans T, et `FAUX` sinon.
4. Si F est de la forme $\neg G$, alors $\tilde{F} = \neg\tilde{G}$: évaluer $\tilde{G}$ dans la destinée T et renvoyer `VRAI` si $\tilde{G}$ est faux dans T, et `FAUX` sinon.
5. Si F est de la forme $\forall x_k G(a_1, \ldots, a_{k-1}, x_k)$, alors $\tilde{F} = \forall x_k P(a_{k-1}, x_k) \rightarrow \tilde{G}(a_1, \ldots, a_{k-1}, x_k)$: le nœud courant est a_{k-1}, d'ascendants $a_{k-2}, \ldots, a_1, \emptyset$. Pour tous les fils a_k du nœud courant, évaluer $\tilde{G}(a_1, \ldots, a_{k-1}, a_k)$ dans T (le nœud courant devient a_k durant cette étape) et renvoyer `VRAI` si pour chacune de ces évaluations $\tilde{G}(a_1, \ldots, a_{k-1}, a_k)$ est vrai dans T, et `FAUX` sinon.
6. Si F est de la forme $\exists x_k G(a_1, \ldots, a_{k-1}, x_k)$, alors $\tilde{F} = \exists x_k P(a_{k-1}, x_k) \wedge \tilde{G}(a_1, \ldots, a_{k-1}, x_k)$: le nœud courant est a_{k-1}, d'ascendants $a_{k-2}, \ldots, a_1, \emptyset$. Pour tous les fils a_k du nœud courant, évaluer $\tilde{G}(a_1, \ldots, a_{k-1}, a_k)$ dans T (le nœud courant devient a_k durant cette étape) et renvoyer `VRAI` si pour au moins une de ces évaluations $\tilde{G}(a_1, \ldots, a_{k-1}, a_k)$ est vrai dans T, et `FAUX` sinon.
7. Si F est une formule atomique dont toutes les variables sont instanciées, alors $\tilde{F} = F$: renvoyer `VRAI` si cette formule atomique apparaît comme vraie dans la liste associée au nœud courant dans T, et `FAUX` sinon.

Remarque 12 *Cet algorithme est parallélisable, comme nous le verrons au Paragraphe 4.4.4.*

4.3 Exemple

Faisons tourner cet algorithme sur un exemple : on considère la structure $\langle \mathbb{N}, \leq, P \rangle$ déjà vue au paragraphe 3.4.2. On fixe $p = 2$ la profondeur de quantification des énoncés que l'on va décider. On choisit une 2-destinée exhaustive et essentielle de la structure $\langle \mathbb{N}, \leq, P \rangle$, par exemple la 2-destinée réduite (Figure 4.1).

On veut évaluer l'énoncé : $\forall x P(x) \wedge \exists y[(x < y) \vee P(y)]$

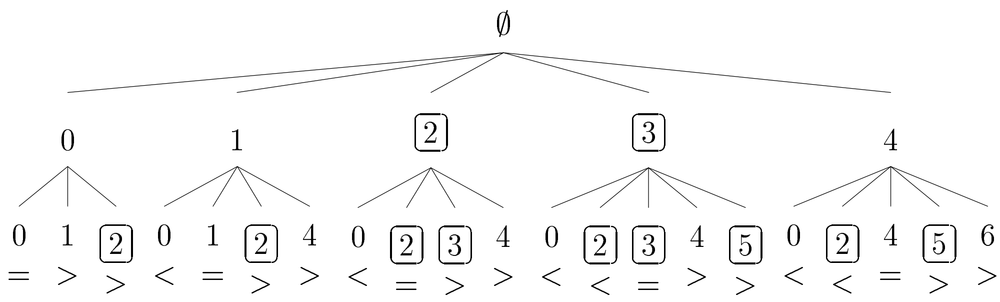


FIG. 4.1 – La 2-destinée réduite de la structure $\langle \mathbb{N}, \leq, P \rangle$.

L'énoncé F est de la forme $\forall x G(x)$, donc, $\tilde{F} = \forall x(P(\emptyset, x) \rightarrow \tilde{G}(x))$. Pour tout fils a du nœud courant (qui est $\emptyset$), on évalue $\tilde{G}(a)$:

- Fils 0 : on évalue $\tilde{G}(0)$, or $G(0)$ est $P(0) \wedge \exists y[(0 < y) \vee P(y)]$ qui est de la forme $A(0) \wedge B(0)$.
  Donc $\tilde{G}(0) = \tilde{A}(0) \wedge \tilde{B}(0)$.
  On évalue chaque sous-formule $\tilde{A}(0)$ et $\tilde{B}(0)$:
  - $A(0) = P(0)$ est une formule atomique, donc $\tilde{A}(0) = P(0)$.
    On lit sur la liste associée au nœud courant (0) qu'elle est fausse dans T.
    On renvoie `FAUX`.

  L'une des deux sous-formules de $\tilde{A} \wedge \tilde{B}$ est fausse dans T donc la formule $\tilde{G}(0)$ est fausse dans T.
  On renvoie donc `FAUX`.

L'une des formules $\tilde{G}(a)$ est fausse donc $\tilde{F}$ est fausse dans T, donc on renvoie `FAUX`.
L'énoncé de départ n'est pas un théorème.

On peut remarquer sur cet exemple que lors de l'exécution de l'algorithme, on n'est pas forcément obligé d'évaluer les sous-formules sur toutes les instanciations possibles dans la destinée : si la quantification est universelle, on peut arrêter la descente dans l'algorithme dès qu'on a un résultat négatif, et si la quantification est existentielle, on peut arrêter la descente dans l'algorithme dès qu'on a un résultat positif (et on peut même garder l'instanciation comme témoin pour cette quantification existentielle).

4.4 Paramètres et calcul de la complexité de l'algorithme

Pour calculer la complexité de cet algorithme, on a besoin de connaitre :

- la taille de l'énoncé de départ, que l'on appelera n ;
- la profondeur de quantification de l'énoncé, que l'on appelera p (remarquons qu'on a nécessairement $p \leq n$) ;
- le nombre de nœuds d'une p-destinée exhaustive et essentielle (excepté $\emptyset$), que l'on appelera N_p .

Pour effectuer le calcul de complexité, on a besoin d'indexer d'une part chaque sous-formule de l'énoncé, d'autre part chaque nœud de la destinée.

4.4.1 Indexation des sous-formules de l'énoncé

Soit Ψ l'énoncé de profondeur de quantification p à décider.

- Au rang de quantification p : la formule Ψ est une combinaison booléenne de n_0 sous-énoncés de profondeur de quantification p.
  On les indice par $i_0 \in \{1, \ldots, n_0\}$.
  Ces sous-formules sont appelées Ψ_{i_0}.
- Au rang de quantification $p-1$: pour chaque $i_0 \in \{1, \ldots, n_0\}$, la formule Ψ_{i_0} est de la forme $Q_{i_0} x_1 \Psi'_{i_0}(x_1)$, où Ψ'_{i_0} est une combinaison booléenne de formules de profondeur de quantification $p-1$.
  On suppose qu'il y a $n_{i_0,1}$ sous-formules composant Ψ'_{i_0}.
  On les indicera par $i_{i_0,1} \in \{1, \ldots, n_{i_0,1}\}$.
  Ces sous-formules sont appelées $\Psi_{i_0,1}$ (au lieu de $\Psi_{i_{i_0,1}}$, pour des raisons de commodités de lecture).
- ...
- Au rang de quantification k : pour chaque $i_{i_0,i_1,\ldots,i_{k-2},k-1} \in \{1, \ldots, n_{i_0,i_1,\ldots,i_{k-2},k-1}\}$, la formule $\Psi_{i_0,i_1,\ldots,i_{k-2},k-1}$ est de la forme
  $$Q_{i_0,\ldots,i_{k-2},k-1} x_{p-k} \Psi'_{i_0,i_1,\ldots,i_{k-2},k-1}(x_1, \ldots, x_{p-k}),$$
  où $\Psi'_{i_0,i_1,\ldots,i_{k-2},k-1}$ est une combinaison booléenne de formules de profondeur de quantification k.
  On suppose qu'il y a $n_{i_0,i_1,\ldots,i_{k-1},k}$ telles sous-formules.
  On les indice par $i_{i_0,i_1,\ldots,i_{k-1},k} \in \{1, \ldots, n_{i_0,i_1,\ldots,i_{k-1},k}\}$.
  Ces sous-formules sont appelées $\Psi_{i_0,i_1,\ldots,i_{k-1},k}$.
- ...
- Au rang de quantification 0 : pour chaque $i_{i_0,i_1,\ldots,i_{p-2},p-1} \in \{1, \ldots, n_{i_0,i_1,\ldots,i_{p-2},p-1}\}$, la formule $\Psi_{i_0,i_1,\ldots,i_{p-2},p-1}$ est de la forme
  $$Q_{i_0,\ldots,i_{p-2},p-1} x_p \Psi'_{i_0,i_1,\ldots,i_{p-2},p-1}(x_1, \ldots, x_p),$$
  où $\Psi'_{i_0,i_1,\ldots,i_{p-2},p-1}$ est une combinaison booléenne de formules atomiques.
  On suppose qu'il y a $n_{i_0,i_1,\ldots,i_{p-1},p}$ telles sous-formules.
  On les indice par $i_{i_0,i_1,\ldots,i_{p-1},p} \in \{1, \ldots, n_{i_0,i_1,\ldots,i_{p-1},p}\}$.
  Ces sous-formules sont appelées $\Psi_{i_0,i_1,\ldots,i_{p-1},p}$.

Le nombre total de formules atomiques dans Ψ est alors :

$$\left(\sum_{i_0=1}^{n_0}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\cdots\sum_{i_{i_0,\ldots,i_{k-1},k}=1}^{n_{i_0,\ldots,i_{k-1},k}}\cdots\sum_{i_{i_0,\ldots,i_{p-1},p}=1}^{n_{i_0,\ldots,i_{p-1},p}} 1\right)=\mathcal{O}(n).$$

4.4.2 Indexation des nœuds de la destinée

Soit T la p-destinée exhaustive et essentielle sur laquelle se déroule l'algorithme, et N_p son nombre total de nœuds.

- Au rang 1 : on suppose que le nœud $\emptyset$ a M_1 fils de rang 1.
  On les indice par $j_1 \in \{1,\ldots,M_1\}$.
  On les appelle u_{j_1}.
- Au rang 2 : on suppose que le nœud u_{j_1} a $M_{j_1,2}$ fils de rang 2.
  On les indice par $j_{j_1,2} \in \{1,\ldots,M_{j_1,2}\}$.
  On les appelle $u_{j_{j_1,2}}$.
- ...
- Au rang k : on suppose que le nœud $u_{j_1,\ldots,j_{k-2},k-1}$ a $M_{j_1,\ldots,j_{k-1},k}$ fils de rang k.
  On les indice par $j_{j_1,\ldots,j_{k-1},k} \in \{1,\ldots,M_{j_1,\ldots,j_{k-1},k}\}$.
  On les appelle $u_{j_{j_1,\ldots,j_{k-1},k}}$.
- ...
- Au rang p : on suppose que le nœud $u_{j_1,\ldots,j_{p-2},p-1}$ a $M_{j_1,\ldots,j_{p-1},p}$ fils de rang p.
  On les indice par $j_{j_1,\ldots,j_{p-1},p} \in \{1,\ldots,M_{j_1,\ldots,j_{p-1},p}\}$.
  On les appelle $u_{j_{j_1,\ldots,j_{p-1},p}}$.

On a :

$$\left(\sum_{j_1=1}^{M_1}\sum_{j_{j_1,2}=1}^{M_{j_1,2}}\cdots\sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{M_{j_1,\ldots,j_{k-1},k}}\cdots\sum_{j_{j_1,\ldots,j_{p-1},p}=1}^{M_{j_1,\ldots,j_{p-1},p}} 1\right)=N_p.$$

4.4.3 Calcul de la complexité

On note $T(\Psi)$ le coût d'évaluation de Ψ, et de manière générale, $T(F)$ le coût d'évaluation d'une sous-formule F. On a :

- Le coût d'évaluation de la formule Ψ est la somme des coûts d'évaluation des formules Ψ_{i_0}, plus le coût d'évaluation de la combinaison booléenne, qui est en $\mathcal{O}(n_0)$:

$$T(\Psi)=\sum_{i_0=1}^{n_0} T(\Psi_{i_0})+\mathcal{O}(n_0)$$
$$T(\Psi)\leq\sum_{i_0=1}^{n_0} T(\Psi_{i_0})+\mathcal{O}(n)$$

- Pour chaque Ψ_{i_0}, on doit évaluer pour chaque instanciation de la première variable par les fils du nœud courant (ici $\emptyset$), la formule Ψ'_{i_0}, et le coût d'évaluation de cette dernière est la somme des coûts d'évaluation des $\Psi_{i_0,1}$, plus le coût d'évaluation des combinaisons booléennes pour chacune de ces instanciations, qui sont en $\mathcal{O}(n_{i_0,1})$:

$$T(\Psi) \leq \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} (T(\Psi_{i_0,1}) + \mathcal{O}(n_{i_0,1})) + \mathcal{O}(n)$$

$$T(\Psi) \leq \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} T(\Psi_{i_0,1}) + \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} \mathcal{O}(n_{i_0,1}) + \mathcal{O}(n)$$

$$T(\Psi) \leq \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} T(\Psi_{i_0,1}) + \sum_{j_1=1}^{M_1}\sum_{i_0=1}^{n_0}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} \mathcal{O}(n_{i_0,1}) + \mathcal{O}(n)$$

$$T(\Psi) \leq \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} T(\Psi_{i_0,1}) + \sum_{j_1=1}^{M_1} \mathcal{O}(n) + \mathcal{O}(n)$$

$$T(\Psi) \leq \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} T(\Psi_{i_0,1}) + M_1 \times \mathcal{O}(n) + \mathcal{O}(n)$$

$$T(\Psi) \leq \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} T(\Psi_{i_0,1}) + 2 \times \mathcal{O}(n \times N_p)$$

- ...
- Pour chaque $\Psi_{i_0,i_1,\ldots,i_{k-2},k-1}$, on doit évaluer, pour chaque instanciation de la variable x_k par les fils du nœud courant $(u_{j_{j_1,\ldots,j_{k-1},k}})$, la formule $\Psi'_{i_0,i_1,\ldots,i_{k-2},k-1}$, et le coût d'évaluation de cette dernière est la somme des coûts d'évaluation des $\Psi_{i_0,i_1,\ldots,i_{k-1},k}$ plus le coût d'évaluation des combinaisons booléennes :

$$\begin{aligned}
T(\Psi) \leq & \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} \cdots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{M_{j_1,\ldots,j_{k-1},k}} \sum_{i_{i_0,\ldots,i_{k-1},k}=1}^{n_{i_0,\ldots,i_{k-1},k}} (T(\Psi_{i_0,\ldots,i_{k-1},k}) + \mathcal{O}(n_{i_0,\ldots,i_{k-1},k})) \\
& + (k-1) \times \mathcal{O}(n \times N_p) \\
T(\Psi) \leq & \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} \cdots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{M_{j_1,\ldots,j_{k-1},k}} \sum_{i_{i_0,\ldots,i_{k-1},k}=1}^{n_{i_0,\ldots,i_{k-1},k}} T(\Psi_{i_0,\ldots,i_{k-1},k}) \\
& + \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} \cdots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{M_{j_1,\ldots,j_{k-1},k}} \sum_{i_{i_0,\ldots,i_{k-1},k}=1}^{n_{i_0,\ldots,i_{k-1},k}} \mathcal{O}(n_{i_0,\ldots,i_{k-1},k}) \\
& + (k-1) \times \mathcal{O}(n \times N_p) \\
T(\Psi) \leq & \sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}} \cdots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{M_{j_1,\ldots,j_{k-1},k}} \sum_{i_{i_0,\ldots,i_{k-1},k}=1}^{n_{i_0,\ldots,i_{k-1},k}} T(\Psi_{i_0,\ldots,i_{k-1},k})
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{j_1=1}^{M_1}\dots\sum_{j_{j_1,\dots,j_{k-1},k}=1}^{M_{j_1,\dots,j_{k-1},k}}\sum_{i_0=1}^{n_0}\dots\sum_{i_{i_0,\dots,i_{k-1},k}=1}^{n_{i_0,\dots,i_{k-1},k}}\mathcal{O}(n_{i_0,\dots,i_{k-1},k})+(k-1)\times\mathcal{O}(n\times N_p)\\
T(\Psi)\leq&\sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\dots\sum_{j_{j_1,\dots,j_{k-1},k}=1}^{M_{j_1,\dots,j_{k-1},k}}\sum_{i_{i_0,\dots,i_{k-1},k}=1}^{n_{i_0,\dots,i_{k-1},k}}T(\Psi_{i_0,\dots,i_{k-1},k})\\
&+\sum_{j_1=1}^{M_1}\dots\sum_{j_{j_1,\dots,j_{k-1},k}=1}^{M_{j_1,\dots,j_{k-1},k}}\mathcal{O}(n)+(k-1)\times\mathcal{O}(n\times N_p)\\
T(\Psi)\leq&\sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\dots\sum_{j_{j_1,\dots,j_{k-1},k}=1}^{M_{j_1,\dots,j_{k-1},k}}\sum_{i_{i_0,\dots,i_{k-1},k}=1}^{n_{i_0,\dots,i_{k-1},k}}T(\Psi_{i_0,\dots,i_{k-1},k})\\
&+\mathcal{O}(n\times N_p)+(k-1)\times\mathcal{O}(n\times N_p)\\
T(\Psi)\leq&\sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\dots\sum_{j_{j_1,\dots,j_{k-1},k}=1}^{M_{j_1,\dots,j_{k-1},k}}\sum_{i_{i_0,\dots,i_{k-1},k}=1}^{n_{i_0,\dots,i_{k-1},k}}T(\Psi_{i_0,\dots,i_{k-1},k})+k\times\mathcal{O}(n\times N_p)
\end{aligned}
$$

- …
- Au final, on arrive à l'évaluation des formules atomiques :

$$
T(\Psi)\leq\sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\dots\sum_{j_{j_1,\dots,j_{p-1},p}=1}^{M_{j_1,\dots,j_{p-1},p}}\sum_{i_{i_0,\dots,i_{p-1},p}=1}^{n_{i_0,\dots,i_{p-1},p}}T(\Psi_{i_0,\dots,i_{p-1},p})+p\times\mathcal{O}(n\times N_p).
$$

L'évaluation de la formule atomique consiste à accéder à la liste de relations portées par le nœud concerné (ici, dans le pire des cas, on a supposé que l'on était obligé d'aller jusqu'aux feuilles, mais on peut rencontrer des formules atomiques avant). Cette liste de relations a une taille qui dépend du langage σ. On notera $\Gamma_\sigma(k)$ la longueur de la liste des relations vérifiées par le nœud vis-à-vis de lui-même et ses ascendants, si le nœud est de rang k. L'évaluation d'une formule atomique faisant intervenir k variables instanciées dans l'arbre est de l'ordre de $\Gamma_\sigma(k)$. La fonction Γ_σ est croissante, donc l'évaluation d'une formule atomique est majorée par $\Gamma_\sigma(p)$, quel que soit le rang auquel on doit l'évaluer.

On peut dès lors majorer $T(\Psi)$:

$$
T(\Psi)\leq\sum_{i_0=1}^{n_0}\sum_{j_1=1}^{M_1}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\dots\sum_{j_{j_1,\dots,j_{p-1},p}=1}^{M_{j_1,\dots,j_{p-1},p}}\sum_{i_{i_0,\dots,i_{p-1},p}=1}^{n_{i_0,\dots,i_{p-1},p}}\Gamma_\sigma(p)+\mathcal{O}(p\times n\times N_p).
$$

Les sommes concernant les indices j d'une part et les indices i d'autre part, sont devenues indépendantes, et l'on peut écrire :

$$
T(\Psi)\leq\sum_{i_0=1}^{n_0}\sum_{i_{i_0,1}=1}^{n_{i_0,1}}\dots\sum_{i_{i_0,\dots,i_{p-1},p}=1}^{n_{i_0,\dots,i_{p-1},p}}\left(\sum_{j_1=1}^{M_1}\dots\sum_{j_{j_1,\dots,j_{p-1},p}=1}^{M_{j_1,\dots,j_{p-1},p}}\Gamma_\sigma(p)\right)+\mathcal{O}(p\times n\times N_p)
$$

d'où

$$T(\Psi) \leq \sum_{i_0=1}^{n_0} \sum_{i_{i_0,1}=1}^{n_{i_0,1}} \cdots \sum_{i_{i_0,\ldots,i_{p-1},p}=1}^{n_{i_0,\ldots,i_{p-1},p}} (N_p \times \Gamma_\sigma(p)) + \mathcal{O}(p \times n \times N_p)$$

et finalement, puisque $\Gamma_\sigma(p)$ est supérieur ou égal à p :

$$T(\Psi) \leq \mathcal{O}(n \times N_p \times \Gamma_\sigma(p)).$$

Remarque 13 *Ce calcul de complexité met en évidence la part de chaque paramètre intervenant dans la décision : il y a en effet la part de l'énoncé avec le paramètre n, la part du langage avec la fonction Γ_σ, et enfin la part de la structure avec le paramètre N_p.*

Exemple : Si on considère la structure $\langle \mathbb{N}, \leq, P \rangle$ et le rang de quantification 2, on a :

- Le nombre de nœuds dans une destinée exhaustive et essentielle est 26, comme on peut le voir Figure 4.2 ;

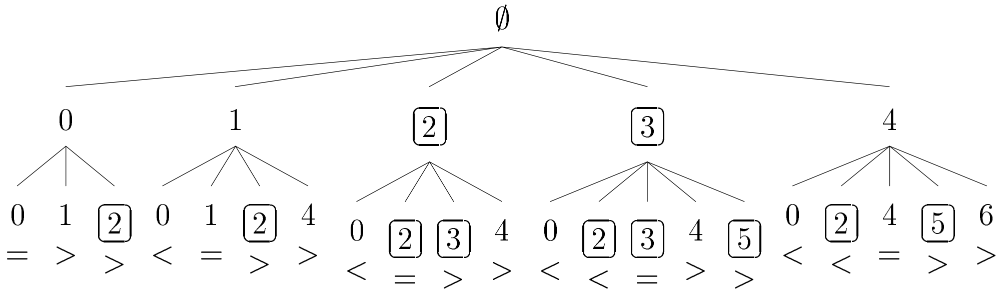


FIG. 4.2 – La 2-destinée réduite de la structure $\langle \mathbb{N}, \leq, P \rangle$.

- Les nœuds de rang 1 s'accompagnent d'une liste d'un booléen qui est vrai si le nœud est premier, et faux sinon, les nœuds de rang 2 s'accompagnent d'une liste de trois booléens : le premier qui indique le caractère premier du nœud, le deuxième qui dit si le nœud est inférieur ou égal à son père, et le troisième qui indique si le nœud est supérieur ou égal à son père. Donc on a ici $\Gamma(2) = 3$.

La complexité de la décision d'un énoncé de profondeur de quantification 2 sur cette structure sera donc en $\mathcal{O}(78 \times n)$ où n est la longueur de l'énoncé.

Pour la 3-destinée réduite de cette même structure, le nombre de nœuds est de l'ordre de 800 nœuds, et $\Gamma(3)$ est égal à 5. On assiste à une explosion combinatoire de N_p lorsque p dépasse la valeur 3. Cette explosion combinatoire entre la profondeur 2 et la profondeur 3 s'observe dans toutes les structures que nous avons étudiées.

4.4.4 Complexité pour les machines de Turing alternantes

Afin de pouvoir comparer l'algorithme que nous venons de présenter à d'autres algorithmes d'élimination de quantificateurs, nous présentons ici un deuxième calcul de complexité, dans le modèle de calcul des machines de Turing alternantes. Nous intégrons donc l'aspect parallèlisable de l'algorithme pour le calcul de sa complexité. Nous commençons par rappeler le modèle de calcul des machines de Turing alternantes, avant de nous lancer dans le calcul de la complexité.

4.4.4.1 Machines de Turing alternantes

On peut se référer pour la notion de machine de Turing alternante à [BDG90], [Pap94] ou [VL90]. C'est une variante de la notion de machine de Turing non déterministe, que nous supposons connue.

Définition 37 (Machine de Turing alternante)
Une **machine de Turing alternante** *est un quadruplet $M = (Q, \Sigma, \delta, q_0)$, où :*
- *Q est l'ensemble des états, partitionné en 4 sous-ensembles disjoints, respectivement états universels, existentiels, acceptants et rejetants ;*
- *Σ est l'alphabet ;*
- *δ est la fonction de transition (non déterministe) ;*
- *q_0 est l'état initial.*

Définition 38 (Configuration)
Une **configuration** *pour une machine de Turing alternante M est un vecteur composé de l'état courant de la machine, de l'état des rubans, du contenu de l'entrée et la position des têtes sur les rubans. Si l'état d'une configuration est universel (respectivement existentiel, acceptant, rejetant), on dit que la configuration est universelle (respectivement existentielle, acceptante, rejetante).*

Définition 39 (Successeur d'une configuration)
Soient α_1 et α_2 deux configurations pour une machine alternante M. On dit que α_2 est un **sucesseur** *de α_1, et on note $\alpha_1 \vdash \alpha_2$ si α_1 mène à la configuration α_2 par une application de la fonction de transition.*

Il reste à définir la notion d'acceptance pour une machine de Turing alternante. Pour cela, on s'appuie sur l'arbre de calcul sur une entrée :

Définition 40 (Arbre de calcul)
*Soit M une machine de Turing alternante et x une entrée. L'***arbre de calcul** *pour M sur l'entrée x est un arbre dont les nœuds sont des configurations, dont la racine est la configuration initiale α_0 et pour tout nœud α_k, les fils de α_k sont exactement ses configurations successeurs.*

Définition 41 (Sous-arbre acceptant)
Soit M une machine de Turing alternante et x une entrée, et T l'arbre de calcul pour M sur l'entrée x. On dit qu'une feuille de T répond "oui" quand elle est acceptante, et qu'un nœud qui n'est pas une feuille répond "oui" lorsque :
- *si le nœud est universel, tous ses fils répondent "oui" ;*
- *si le nœud est existentiel, au moins l'un de ses fils répond "oui".*

Si la racine de T répond "oui", on dit que le sous-arbre de T constitué des nœuds qui répondent "oui" est un **sous-arbre acceptant** *pour M sur l'entrée x.*

Définition 42 (Acceptation)
Soit M une machine de Turing alternante et x une entrée. On dit que M **accepte** *x s'il existe un sous-arbre de calcul acceptant pour M sur l'entrée x.*

Pour définir la complexité associée à ce modèle de calcul, on se sert également des arbres de calcul :

Définition 43 (Classe de complexité $ATIME(k, f(n))$)
Soit M une machine de Turing alternante. On dit que M est dans la classe de complexité $ATIME(k, f(n))$, si sur une entrée x de taille n, la réponse de M sur l'entrée x est donnée après un temps $f(n)$ et k alternances de quantification (c'est-à-dire que l'arbre de calcul a ses branches de longueur inférieure à $f(n)$ et il y a au plus k alternances des états existentiels et universels sur une branche).

4.4.4.2 Complexité de l'algorithme de décision utilisant les destinées

Nous allons décrire l'arbre de calcul pour cet algorithme. Il ressemble fortement à la p-destinée. Soit Ψ un énoncé de profondeur de quantification p. On utilisera les notations du Paragraphe 4.4.1.
- Une première partie de l'arbre décrit la combinaison booléenne des Ψ_{i_0}. Cett partie est de hauteur au plus n_0.
- Pour chacune des Ψ_{i_0}, on a un nœud de l'arbre de calcul qui est dans un état existentiel si Q_{i_0} est $\exists$ et universel sinon. Ce nœud a pour fils toutes les situations correspondant aux instanciations de x_1 par les fils de $\emptyset$ dans la p-destinée.
- Pour chacun de ces fils, on a ensuite une partie de l'arbre de calcul qui décrit la combinaison booléenne des $\Psi_{i_0,1}$, cette partie est de hauteur au plus $n_{i_0,1}$ dans le sous-arbre correspondant à la formules Ψ_{i_0}
- ...
- Lorsque l'on en est à évaluer une formule atomique dont toutes les variables sont instanciées, on lit dans une liste de longueur au plus $\Gamma(p)$. Le nœud a alors pour fils

les situations correspondant aux élément d'une telle liste, ce qui nous rajoute une longueur 1.

En conclusion, chaque branche est de longueur $\mathcal{O}(n)$, et contient au plus p alternances de quantificateurs : l'algorithme appartient à la classe de complexité $ATIME(p, \mathcal{O}(n))$.

4.5 Conclusion

La présentation sous forme de destinée permet de gagner du temps par rapport à une élimination des quantificateurs classique. D'une part, elle ne nécessite pas de prétraitement de la formule (par exemple, une mise sous forme prénexe), et d'autre part les évaluations des formules atomiques ont déjà été effectuées lors de la construction de la p-destinée (et elles sont stockées sous forme des listes de relations attachées aux nœuds). La p-destinée permet donc de stocker dans une structure de données finie tout ce dont on a besoin pour évaluer rapidement n'importe quel énoncé de profondeur de quantification inférieure ou égale à p.

Il y a évidemment une ombre au tableau :

- la destinée est ici supposée connue, mais il reste à déterminer comment la construire, et si c'est possible, automatiser cette construction. Nous verrons dans la Partie II que cela n'est pas toujours simple ;
- de plus la complexité de l'algorithme de décision présenté ci-dessus dépend de la taille de la p-destinée utilisée. Or cette taille peut facilement devenir énorme, de l'ordre de la tour d'exponentielle par exemple, lorsque p croît (ce qui n'est pas en soi étonnant, vu les bornes inférieures de complexité de la décision de certaines théories arithmétiques). Nous verrons quelques exemples également dans la Partie II.

L'algorithme est en revanche particulièrement intéressant dès qu'il s'agit de décider un nombre important d'énoncés de profondeur de quantification donnée sur une même structure : la construction de la destinée étant faite, la complexité d'évaluation de chaque énoncé est linéaire (certes avec une grande constante !), contrairement à ce qu'il se passe avec d'autres algorithmes. Nous verrons une comparaison détaillée au chapitre 9 entre l'algorithme de décision de Nézondet, utilisant les destinées, et un algorithme de décision dû à Ferrante et Rackoff ([FR79]).

L'implémentation de l'algorithme ci-dessus a été réalisée en langage Caml, et est accessible sur la page :

`http://llaic3.u-clermont1.fr/~chateau/pub/decision.tgz`

On peut se renseigner sur le langage Caml dans [LW99] ou [CMP00].

Chapitre 5

Comparaison des destinées de Nézondet avec les k-isomorphismes et les jeux

Ce chapitre fait le lien entre les destinées de Nézondet, outil nouveau pour l'étude des structures lorsque l'on fixe la profondeur de quantification des énoncés, et les outils déjà existants dans le même contexte, à savoir les k-isomorphismes de Fraïssé et les jeux d'Ehrenfeucht. Ces notions désormais classiques sont amplement présentées dans [Hod93], [EF99], [EFT94], [Mau94] ou [Poi85].

Nous rappelons ici les définitions essentielles de ces notions, puis nous établissons la comparaison avec les destinées dans le Théorème 3.

Les isomorphismes locaux et la notion de va-et-vient ont été introduits par Fraïssé dans les années 1950, puis Ehrenfeucht décrit de manière différente les va-et-vient en introduisant un jeu entre deux personnes, pour lequel l'existence d'une stratégie gagnante pour l'un des joueurs implique l'équivalence élémentaire entre deux structures.

5.1 Rappel sur les k-isomorphismes de Fraïssé

Soient $\mathcal{A}$ et $\mathcal{B}$ deux σ-structures de domaines respectifs A et B.

Définition 44 (Isomorphisme local)
Un **isomorphisme local** *entre $\mathcal{A}$ et $\mathcal{B}$ est un isomorphisme de structures entre une partie finie de $\mathcal{A}$ et une partie finie de $\mathcal{B}$.*

La notion de k-isomorphisme est définie par récurrence sur k.

Définition 45 (k-isomorphisme)
Un isomorphisme local f entre $\mathcal{A}$ et $\mathcal{B}$ est un $(k+1)$**-isomorphisme** *si et seulement s'il vérifie les conditions suivantes :*
- *condition de "va" : pour tout $a \in A$, il existe un k-isomorphisme g prolongeant f et défini en a ;*
- *condition de "vient" : pour tout $b \in B$, il existe un k-isomorphisme g prolongeant f et dont l'image contient b.*

Définition 46 (k-équivalence)
Deux n-uples $(a_1, \ldots, a_n)$ et $(b_1, \ldots, b_n)$ d'éléments de A et B respectivement sont dits k**-équivalents**, *ou k-isomorphes, si l'application de domaine $\{a_1, \ldots, a_n\}$ qui envoie chaque a_i sur b_i pour tout $i \in \{1, \ldots, n\}$ est un k-isomorphisme.*

5.2 Rappel sur les jeux d'Ehrenfeucht

Définition 47 (Jeu d'Ehrenfeucht)
Soit $k \in \mathbb{N}$. Le **jeu d'Ehrenfeucht** $G_k(\mathcal{A}, \mathcal{B})$ *est un jeu à deux joueurs* Joueur I *et* Joueur II, *en k coups. Un coup se décompose en deux actions :*
- Joueur I *choisit un élément de $\mathcal{A}$ ou de $\mathcal{B}$;*
- Joueur II *choisit un élément de $\mathcal{B}$ si* Joueur I *a choisi un élément de $\mathcal{A}$, ou un élément de $\mathcal{A}$ si* Joueur I *a choisi un élément de $\mathcal{B}$.*

On appelle $(\alpha_1, \ldots, \alpha_k)$ les éléments de $\mathcal{A}$ qui ont été choisis au cours du jeu par l'un ou l'autre joueur, et $(\beta_1, \ldots, \beta_k)$ les éléments de $\mathcal{B}$ choisis au cours du jeu (le couple (α_i, β_i) étant choisi au coup numéro i). Joueur II *gagne le jeu si $(\alpha_1, \ldots, \alpha_k)$ et $(\beta_1, \ldots, \beta_k)$ vérifient les mêmes formules sans quantificateur dans leurs structures respectives.*

Définition 48 (Stratégie gagnante)
Soit $k \in \mathbb{N}$. Une **stratégie gagnante** *pour le jeu d'Ehrenfeucht $G_k(\mathcal{A}, \mathcal{B})$ est la description d'une suite d'actions à exécuter pour* Joueur II *en fonction des actions possiblement réalisées par* Joueur I, *telles que si* Joueur II *suit cette stratégie, il gagne le jeu à coup sûr.*

Nous verrons au Paragraphe 5.4 que l'existence d'une stratégie gagnante équivaut à l'existence d'un k-isomorphisme.

5.3 Formules d'Hintikka

Soit $\mathcal{A}$ une σ-structure, et $(a_1, \ldots, a_n)$ un n-uple d'éléments de $\mathcal{A}$. On définit les formules d'Hintikka d'ordre k du n-uple $(a_1, \ldots, a_n)$ par récurrence sur k.

Définition 49 (Formules d'Hintikka)
Les **formules d'Hintikka** *du n-uple $(a_1, \ldots, a_n)$ sont :*
- *pour $k = 0$:*

$$\varphi^0_{(a_1,\ldots,a_n)}(x_1, \ldots, x_n) \equiv \bigwedge\{\varphi(x_1, \ldots, x_n) \text{ atomique ou négatomique telle que } \mathcal{A} \models \varphi(a_1, \ldots, a_n)\}$$

- *pour $k > 0$:*

$$\varphi^k_{(a_1,\ldots,a_n)}(x_1, \ldots, x_n) \equiv \bigwedge_{a\in A} \exists x_{n+1} \varphi^{k-1}_{(a_1,\ldots,a_n,a)}(x_1, \ldots, x_n, x_{n+1})$$

$$\wedge \forall x_{n+1} \bigvee_{a\in A} \varphi^{k-1}_{(a_1,\ldots,a_n,a)}(x_1, \ldots, x_n, x_{n+1})$$

Les formules d'Hintikka caractérisent le **type de k-isomorphisme** du n-uple $(a_1, \ldots, a_n)$, comme nous le verrons dans le Théorème 3. D'autre part, les formules d'Hintikka d'ordre k forment une base de représentants des formules de profondeur de quantification inférieure ou égale à k, au sens précisé par la proposition ci-dessous.

Proposition 4
Soit φ une σ-formule à n variables libres de profondeur de quantification inférieure ou égale à $k \geq 0$. La formule φ est logiquement équivalente à la formule :

$$\bigvee\{\varphi^k_{(\mathcal{A},a_1,\ldots,a_n)} / \ \mathcal{A}\ \sigma\text{-structure}, (a_1, \ldots, a_n) \text{ dans } \mathcal{A} \text{ et } \mathcal{A} \models \varphi(a_1, \ldots, a_n)\}.$$

5.4 Théorème de comparaison

Les notions définies précédemment sont équivalentes, et les destinées de Nézondet s'intègrent parfaitement au sein de cette équivalence, ce qui s'exprime dans le Théorème 3. Nézondet montre une partie de ce résultat dans [Néz97] (Proposition 4.4.1 page 75).

Theorème 3
Soient $\mathcal{A}$ et $\mathcal{B}$ deux σ-structures, $(a_1, \dots, a_n)$ et $(b_1, \dots, b_n)$ deux n-uples d'éléments de $\mathcal{A}$ et $\mathcal{B}$ respectivement. Soit $k \geq 0$. Les assertions suivantes sont équivalentes :

1. *Les n-uples $(a_1, \dots, a_n)$ et $(b_1, \dots, b_n)$ sont k-isomorphes (au sens de Fraïssé).*
2. Joueur II *a une stratégie gagnante pour le jeu d'Ehrenfeucht en k coups $G_k((\mathcal{A}, a_1, \dots, a_n), (\mathcal{B}, b_1, \dots, b_n))$.*
3. $\mathcal{B} \models \varphi^k_{(a_1,\dots,a_n)}(b_1, \dots, b_n)$.
4. *Les n-uples $(a_1, \dots, a_n)$ dans $\mathcal{A}$ et $(b_1, \dots, b_n)$ dans $\mathcal{B}$ satisfont respectivement les mêmes formules de profondeur de quantification k à n variables libres.*
5. *Deux k-destinées exhaustives des structures $(\mathcal{A}, a_1, \dots, a_n)$ et $(\mathcal{B}, b_1, \dots, b_n)$ sont isomorphes.*

Preuve : L'équivalence entre les quatre premières assertions a été démontrée dans [Hod93] ou [EF99]. Il reste à démontrer l'équivalence entre le point *5.* et les autres.

On s'appuie sur le lemme suivant, qui caractérise les classes d'isomorphisme de sous-arbres dans une destinée par une formule qui ressemble (et ce n'est pas un hasard !) à une formule d'Hintikka.

Lemme 1 *Soit $p \geq 1$ et T et T' deux p-destinées exhaustives de σ-structures, respectivement $\mathcal{X}$ et $\mathcal{X}'$, où σ est un langage relationnel fini. Pour tout nœud α_k de rang $k \leq p$ et d'ascendants $\alpha_{k-1}, \dots, \alpha_1, \emptyset$ dans la destinée T, il existe une σ-formule $F_{(\alpha_1,\dots,\alpha_k)}(x_1, \dots, x_k)$ à k variables libres et de profondeur de quantification $p-k$ telle que pour tout nœud β_k de rang $k \leq p$ et d'ascendants $\beta_{k-1}, \dots, \beta_1, \emptyset$ dans la destinée T', le sous-arbre $ST(\beta_k)$ est isomorphe au sous-arbre $ST(\alpha_k)$ si et seulement si $T' \models \tilde{F}_{(\alpha_1,\dots,\alpha_k)}(\beta_1, \dots, \beta_k)$.*

Preuve : (du lemme). On montre ce lemme par récurrence décroissante sur $k \in \{0, \dots, p\}$.

<u>Cas $k = p$:</u>
Soit α_p une feuille de T d'ascendants $\alpha_{p-1}, \dots, \alpha_1, \emptyset$. On pose $F_{(\alpha_1,\dots,\alpha_p)}(x_1, \dots, x_p)$ la conjonction de toutes les formules atomiques et négatomiques sur le langage σ qui sont satisfaites par le p-uple $(l(\alpha_1), \dots, l(\alpha_p))$ dans la structure $\mathcal{X}$. C'est une formule sans quantificateurs à p variables libres. Montrons qu'elle convient : soit β_p une feuille d'ascendants $\beta_{p-1}, \dots, \beta_1, \emptyset$ dans la destinée T'.

- Si la feuille β_p dans T' et la feuille α_p dans T sont isomorphes, alors les p-uples $(\alpha_1, \dots, \alpha_p)$ et $(\beta_1, \dots, \beta_p)$ satisfont les mêmes formules atomiques et négatomiques, et comme $\mathcal{X} \models F_{(\alpha_1,\ldots,\alpha_p)}(l(\alpha_1),\ldots,l(\alpha_p))$, on a $T \models \tilde{F}_{(\alpha_1,\ldots,\alpha_p)}(\alpha_1,\ldots,\alpha_p)$, donc $T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_p)}(\beta_1,\ldots,\beta_p)$.

– Si $T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_p)}(\beta_1,\ldots,\beta_p)$, cela signifie que les p-uples $(\alpha_1,\ldots,\alpha_p)$ et $(\beta_1,\ldots,\beta_p)$ satisfont les mêmes formules atomiques et négatomiques sur le langage σ. Comme ils satisfont également les mêmes formules atomiques et négatomiques faisant intervenir la relation de paternité et la constante $\emptyset$, ils satisfont les mêmes formules atomiques et négatomiques sur le langage σ'. Cela signifie donc que la feuille α_p et la feuille β_p sont isomorphes.

Cas $k < p$: Montrons que l'hypothèse pour le rang $k+1$ implique l'hypothèse pour le rang k.

Soit α_k un nœud de T d'ascendants $\alpha_{k-1},\ldots,\alpha_1,\emptyset$. Le nœud α_k a un nombre fini de fils à isomorphisme de sous-arbre près.

Soient $y_1,\ldots,y_{n_k}$ des fils de α_k tels que les sous-arbres $ST(y_i)$ pour $i \in \{1,\ldots,n_k\}$ soient non isomorphes deux à deux et recouvrent toutes les classes d'isomorphisme des sous-arbres dont la racine est un fils de α_k.

Par hypothèse de récurrence, il existe pour chaque $i \in \{1,\ldots,n_k\}$ une formule, notée $F_{(\alpha_1,\ldots,\alpha_k,y_i)}(x_1,\ldots,x_k,x_{k+1})$, à $k+1$ variables libres et de profondeur de quantification $p-k-1$ qui caractérise la classe d'isomorphisme du sous-arbre $ST(y_i)$.

Soit $G(x_1,\ldots,x_k)$ la conjonction de toutes les formules atomiques ou négatomiques satisfaites par le uple $(l(\alpha_1),\ldots,l(\alpha_k))$ dans la structure $\mathcal{X}$.

On pose comme formule $F_{(\alpha_1,\ldots,\alpha_k)}(x_1,\ldots,x_k)$:

$$G(x_1,\ldots,x_k)\wedge\bigwedge_{i=1}^{n_k}\exists x_{k+1}F_{(\alpha_1,\ldots,\alpha_k,y_i)}(x_1,\ldots,x_{k+1})\wedge\forall x_{k+1}\bigvee_{i=1}^{n_k}F_{(\alpha_1,\ldots,\alpha_k,y_i)}(x_1,\ldots,x_{k+1}).$$

C'est une formule à k variables libres et de profondeur de quantification $p-k$. Montrons qu'elle convient : soit β_k un nœud de T' d'ascendants $\beta_{k-1},\ldots,\beta_1,\emptyset$.

– Si le sous-arbre $ST(\beta_k)$ et le sous-arbre $ST(\alpha_k)$ sont isomorphes, alors :
  – Les k-uples $(\alpha_1,\ldots,\alpha_k)$ et $(\beta_1,\ldots,\beta_k)$ satisfont les mêmes formules atomiques et négatomiques sur le langage σ'. Comme $\mathcal{X} \models G(l(\alpha_1),\ldots,l(\alpha_k))$, on a
  $$T \models \tilde{G}(\alpha_1,\ldots,\alpha_k),$$
  donc par isomorphisme
  $$T' \models \tilde{G}(\beta_1,\ldots,\beta_k).$$
  – Pour tout fils y de α_k, il existe un fils z de β_k tel que les sous-arbres $ST(z)$ et $ST(y)$ sont isomorphes. Soit $i \in \{1,\ldots,n_k\}$ tel que $ST(y)$ et $ST(y_i)$ sont isomorphes. Par hypothèse de récurrence, on a :
  $$T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1,\ldots,\beta_k,z).$$
  Donc pour tout $i \in \{1,\ldots,n_k\}$, on a :
  $$T' \models \exists x_{k+1}(P(\beta_k,x_{k+1}) \wedge \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1,\ldots,\beta_k,x_{k+1})),$$
  donc
  $$T' \models \bigwedge_{i=1}^{n_k}\exists x_{k+1}(P(\beta_k,x_{k+1}) \wedge \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1,\ldots,\beta_k,x_{k+1})).$$

- Pour tout fils z de β_k, il existe un fils y de α_k tel que les sous-arbres $ST(z)$ et $ST(y)$ sont isomorphes. Soit $i \in \{1, \ldots, n_k\}$ tel que $ST(y)$ et $ST(y_i)$ sont isomorphes. Par hypothèse de récurrence, on a :

$$T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, z).$$

  Donc pour tout fils z de β_k, il existe un $i \in \{1, \ldots, n_k\}$ tel que

$$T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, z),$$

  ce qui s'écrit également :

$$T' \models \forall x_{k+1}(P(\beta_k, x_{k+1}) \to \bigvee\!\!\!\bigvee_{i=1}^{n_k} \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, x_{k+1})).$$

Finalement :

$$\begin{aligned} T' \models & \tilde{G}(\alpha_1, \ldots, \alpha_k) \\ & \wedge \forall x_{k+1}(P(\beta_k, x_{k+1}) \to \bigvee\!\!\!\bigvee_{i=1}^{n_k} \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, x_{k+1})) \\ & \wedge \bigwedge\!\!\!\bigwedge_{i=1}^{n_k} \exists x_{k+1}(P(\beta_k, x_{k+1}) \wedge \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, x_{k+1})), \end{aligned}$$

ou encore :

$$T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k)}(\beta_1, \ldots, \beta_k).$$

- Si $T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k)}(\beta_1, \ldots, \beta_k)$:
  - On a : $T' \models \tilde{G}(\beta_1, \ldots, \beta_k)$. Donc les uples $(\alpha_1, \ldots, \alpha_k)$ et $(\beta_1, \ldots, \beta_k)$ satisfont les mêmes formules atomiques et négatomiques sur le langage σ. Comme par ailleurs ils satisfont les mêmes formules atomiques et négatomiques faisant intervenir la relation de paternité et la constante $\emptyset$, ils satisfont les mêmes formules atomiques et négatomiques sur le langage σ'.
  - On a : $T' \models \bigwedge\!\!\!\bigwedge_{i=1}^{n_k} \exists x_{k+1}(P(\beta_k, x_{k+1}) \wedge \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, x_{k+1}))$. Donc pour tout $i \in \{1, \ldots, n_k\}$, il existe un fils z de β_k tel que $T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, z)$. Par hypothèse de récurrence cela signifie que les sous-arbres $ST(z)$ et $ST(y_i)$ sont isomorphes. Or, pour tout fils y de α_k , il existe un $i \in \{1, \ldots, n_k\}$ tel que $ST(y)$ et $ST(y_i)$ sont isomorphes. Donc pour tout fils de α_k, il existe un fils de β_k tel que leurs sous-arbres sont isomorphes.
  - On a : $T' \models \forall x_{k+1}(P(\beta_k, x_{k+1}) \to \bigvee\!\!\!\bigvee_{i=1}^{n_k} \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, x_{k+1}))$. Donc pour tout fils z de β_k, il existe un $i \in \{1, \ldots, n_k\}$ tel que $T' \models \tilde{F}_{(\alpha_1,\ldots,\alpha_k,y_i)}(\beta_1, \ldots, \beta_k, z)$. Par hypothèse de récurrence cela signifie que les sous-arbres $ST(z)$ et $ST(y_i)$ sont isomorphes. Donc pour tout fils de β_k, il existe un fils de α_k tel que leurs sous-arbres sont isomorphes.

  Donc les sous-arbres $ST(\alpha_k)$ et $ST(\beta_k)$ sont isomorphes.

$\square$

Retour à la preuve du théorème : Il nous reste encore à démontrer l'équivalence entre le point *5.* et les autres.

On commence par montrer *4.⇒5.* : soient T_a et T_b deux k-destinées exhaustives des structures $(\mathcal{A}, a_1, \ldots, a_n)$ et $(\mathcal{B}, b_1, \ldots, b_n)$ respectivement. Le langage considéré sur ces structures est $\sigma \cup \{c_1, \ldots, c_n\}$, où les c_i sont des symboles de constantes interprétés dans $\mathcal{A}$ par les a_i et dans $\mathcal{B}$ par les b_i. Les destinées T_a et T_b sont donc des $\sigma \cup \{c_1, \ldots, c_n\} \cup \{P, \emptyset\}$-structures. Pour montrer qu'elles sont isomorphes, il suffit, d'après le Lemme 1, de montrer que

$$T_b \models \tilde{F}_{(\emptyset_a)}.$$

où $F_{(\emptyset_a)}$ est l'énoncé caractérisant la k-destinée T_a. Or cet énoncé est de profondeur de quantification k, et fait intervenir les constantes $c_1, \ldots, c_n$ du langage $\sigma \cup \{c_1, \ldots, c_n\}$. On a :

$$T_a \models \tilde{F}_{(\emptyset_a)}$$

donc

$$(\mathcal{A}, a_1, \ldots, a_n) \models F_{(\emptyset_a)}.$$

Si on considère maintenant les c_i comme des variables libres, on a :

$$\mathcal{A} \models F_{(\emptyset_a)}(a_1, \ldots, a_n).$$

Le point *4.* nous dit que :

$$\mathcal{B} \models F_{(\emptyset_a)}(b_1, \ldots, b_n).$$

On fait l'opération inverse (*i.e.* on repasse au langage étendu $\sigma \cup \{c_1, \ldots, c_n\}$) :

$$(\mathcal{B}, b_1, \ldots, b_n) \models F_{(\emptyset_a)}.$$

Cela implique que :

$$T_b \models \tilde{F}_{(\emptyset_a)}.$$

C'est ce qu'on voulait.

On montre maintenant que *5.⇒2.* : soient T_a et T_b les deux k-destinées complètes des structures $(\mathcal{A}, a_1, \ldots, a_n)$ et $(\mathcal{B}, b_1, \ldots, b_n)$ respectivement. La stratégie gagnante pour Joueur II est la suivante :

- Au premier coup :
  - Si Joueur I choisit α_1 dans $(\mathcal{A}, a_1, \ldots, a_n)$, il existe un fils α'_1 de $\emptyset$ dans la destinée T_a qui ait pour étiquette α_1. Alors Joueur II choisit un β_1 dans $(\mathcal{B}, b_1, \ldots, b_n)$ tel que β_1 est l'étiquette d'un fils β'_1 de $\emptyset$ dans la destinée T_b dont le sous-arbre est isomorphe au sous-arbre $ST(\alpha'_1)$. Un tel fils existe puisque les destinées T_a et T_b sont isomorphes.
  - Si Joueur I choisit β_1 dans $(\mathcal{B}, b_1, \ldots, b_n)$: Joueur II fait la même chose en inversant les rôles de T_a et T_b.
- On suppose que les $i < k$ premiers coups ont été joués, et qu'aux uples $(\alpha_1, \ldots, \alpha_i)$ et $(\beta_1, \ldots, \beta_i)$ correspondent (par l'étiquetage) respectivement dans les destinées T_a et T_b des nœuds α'_i et β'_i d'ascendants respectifs $\alpha'_{i-1}, \ldots, \alpha'_1, \emptyset$, et $\beta'_{i-1}, \ldots, \beta'_1, \emptyset$, avec les sous-arbres $ST(\alpha'_i)$ et $ST(\beta'_i)$ isomorphes.

  - Si Joueur I choisit α_{i+1} dans $(\mathcal{A}, a_1, \ldots, a_n)$, il existe un fils α'_{i+1} du nœud α'_i dans la destinée complète T_a tel que l'étiquette de α'_{i+1} est α_{i+1}. Alors Joueur II choisit un β_{i+1} dans $(\mathcal{B}, b_1, \ldots, b_n)$ tel que β_{i+1} est l'étiquette d'un fils β'_{i+1} de $\emptyset$ dans la destinée T_b dont le sous-arbre est isomorphe au sous-arbre $ST(\alpha'_{i+1})$. Un tel fils existe puisque les destinées T_a et T_b sont isomorphes.
  - Si Joueur I choisit β_{i+1} dans $(\mathcal{B}, b_1, \ldots, b_n)$: Joueur II fait la même chose en inversant les rôles de T_a et T_b.
- Au bout des k coups, aux uples $(\alpha_1, \ldots, \alpha_k)$ et $(\beta_1, \ldots, \beta_k)$ correspondent (par l'étiquetage) respectivement dans les destinées T_a et T_b des feuilles α'_k et β'_k d'ascendants respectifs $\alpha'_{k-1}, \ldots, \alpha'_1, \emptyset$, et $\beta'_{k-1}, \ldots, \beta'_1, \emptyset$, tels que les feuilles $ST(\alpha'_k)$ et $ST(\beta'_k)$ sont isomorphes. Comme ce sont des feuilles isomorphes, elles satisfont les mêmes formules atomiques et négatomiques, donc les mêmes formules sans quantificateur. Par conséquent, les étiquettes satisfont également les mêmes formules sans quantificateurs dans leurs structures respectives.

On a donc établi une stratégie gagnante pour Joueur II, ce qui termine la preuve.

□

5.5 Conclusion

Dans ce chapitre, nous avons montré l'équivalence entre la notion de destinée et les autres notions permettant d'étudier les structures sous l'angle des k-isomorphismes. Cette équivalence est primordiale pour élargir le champ d'application des destinées. En effet, nous avons ainsi prouvé qu'elles peuvent servir à montrer que deux structures sont équivalentes (il faudrait prouver que les destinées exhaustives sont isomorphes pour toute hauteur), ou bien montrer qu'une propriété n'est pas définissable au premier ordre (il faudrait pour toute hauteur exhiber deux structures dont les destinées sont isomorphes et dont une seule satisfait la propriété). D'autre part, nous avons démontré dans ce chapitre un lemme qui va nous être très utile par la suite, à savoir la caractérisation des classes d'isomorphisme de sous-arbres par une formule. Nous nous en servirons dans le Chapitre 8, ainsi que tout au long de la Partie III, où nous jonglerons entre toutes les notions présentées dans ce chapitre afin d'étudier certaines structures finies.

Deuxième partie

Structures infinies

Introduction

Dans la partie I, nous avons décrit un agorithme de décision des énoncés de profondeur de quantification inférieure ou égale à un entier k donné. Cet algorithme nécessite la connaissance a priori d'une k-destinée exhaustive finie (par exemple une destinée exhaustive essentielle) de la structure choisie. La question qui se pose dès lors est la suivante : peut-on construire les destinées d'une structure automatiquement ?

Nous pouvons dès à présent fournir un élément de réponse : si on dispose d'un algorithme qui prend en entrée un entier k et produit une k-destinée exhaustive et essentielle de la structure, alors on dispose d'un algorithme de décision de la théorie complète en couplant l'algorithme de construction des destinées et l'algorithme décrit au Chapitre 4. Donc, si la structure est indécidable, il est vain de chercher un algorithme de construction des destinées. En revanche, pour un k donné et une structure donnée, on peut toujours envisager la construction d'une k-destinée exhaustive et essentielle de cette structure, étant donnée la décidabilité de l'ensemble des énoncés de profondeur de quantification inférieure ou égale à k qui sont vrais dans la structure (Proposition 1).

Cette deuxième partie présente quelques exemples de destinées.

Le premier exemple, présenté au Chapitre 6, consiste en une tentative inachevée de construction de la 3-destinée réduite de la structure $\langle \mathbb{N}, S, \perp \rangle$, dont la théorie complète est indécidable. Cette étude, née de la volonté d'exhiber un exemple non trivial de destinée, nous apprend des choses étonnantes, à la fois sur la structure et sur les destinées en général.

Le deuxième exemple, faisant l'objet du Chapitre 7, consiste en la présentation d'un algorithme de construction des k-destinées d'une structure dont la théorie est décidable, la structure $\langle \mathbb{N}, \leq \rangle$.

Dans un souci de généralisation de cet algorithme de construction de destinées à une classe de structures dont les théories possèdent, tout comme la théorie de la structure $\langle \mathbb{N}, \leq \rangle$, une élimination des quantificateurs, nous nous sommes penché au Chapitre 8 sur le cas des structures H-bornées, décrites par Ferrante et Rackoff dans [FR79]. Nous obtenons le résultat suivant : si une structure est H-bornée, avec une fonction H récursive, on peut exhiber un algorithme de construction de k-destinées exhaustives et essentielles de la structure. En revanche, il existe une structure, également présentée au Chapitre 8, pour laquelle on peut explicitement construire des k-destinées exhaustives et essentielles pour tout $k \geq 1$, mais qui n'est H-bornée pour aucune fonction H récursive.

Enfin, cette partie s'achève au Chapitre 9, par une comparaison en terme de complexité de l'algorithme de décision utilisant les destinées, que nous appelerons "Algorithme de décision de Nézondet", et l'algorithme de décision dans les structures H-bornées, que nous appelerons "Algorithme de décision de Ferrante et Rackoff".

Chapitre 6

Construction de la 3-destinée réduite de $\langle \mathbb{N}, S, \perp \rangle$

Les origines de ce problème remontent à une collaboration entre Marcel Guillaume, Denis Richard et Ji Lei Yin au cours du séjour de ce dernier au LLAIC1[1] en 1995. Leur volonté était d'exhiber un exemple 3-destinée d'une structure arithmétique indécidable. De fait, leur travail ainsi que ce qui a été fait par la suite n'a pas réussi à clore le problème : on ne sait toujours pas exhiber une 3-destinée de la structure $\langle \mathbb{N}, S, \perp \rangle$. En revanche, le processus pour aboutir à une destinée presque terminée, et les problèmes restés ouverts, nous apprennent beaucoup sur la structure $\langle \mathbb{N}, S, \perp \rangle$ et notamment sur ce qu'elle exprime au rang de quantification 3. Nous présentons ici un résumé de ce travail, une version plus étendue peut être trouvée en [Cha00] ou [Gui01]. Les travaux antérieurs peuvent être consultés dans [Gui96] ou [Gui01].

6.1 En apéritif : la 2-destinée réduite de $\langle \mathbb{N}, S, \perp \rangle$

Comme la structure $\langle \mathbb{N}, S, \perp \rangle$ a un domaine bien ordonnée, on peut parler de <u>la</u> destinée réduite de cette structure (voir Paragraphe 3.4).

6.1.1 Notations

Les deux prédicats considérés sur la structure sont représentés de la manière suivante sur les figures :

- le successeur S, où l'on a $S(x, y)$ si et seulement si $y = x + 1$, est représenté par une flèche pointant vers le haut si le père est successeur du fils, et une flèche vers le bas si le fils est successeur du père ;
- la coprimarité $\perp$, où l'on a $x \perp y$ si et seulement si x et y ont pour pgcd 1, est représentée par :
  - "$(\perp)$" si le nœud est premier avec lui-même (ce cas n'arrive que si le nœud est 1), "$(\not\perp)$" si le nœud n'est pas premier avec lui-même,

[1] Laboratoire de Logique, Algorithmique et Informatique de Clermont 1

- "$\perp$" si le nœud est premier avec son père, et "$\not\perp$" si le nœud n'est pas premier avec son père.

6.1.2 Résultat

Pour construire la 2-destinée réduite, on commence par construire un arbre complet de hauteur 3 dont la racine est $\emptyset$ et tel que chaque nœud qui n'est pas une feuille a pour fils tous les entiers naturels. Puis on fait apparaître les relations que chaque nœud satisfait avec lui-même et ses ascendants. Enfin on réduit les feuilles dans les sous-arbres puis les sous-arbres dans l'arbre lui-même. Ceci est laissé en exercice au lecteur, le résultat est le suivant :

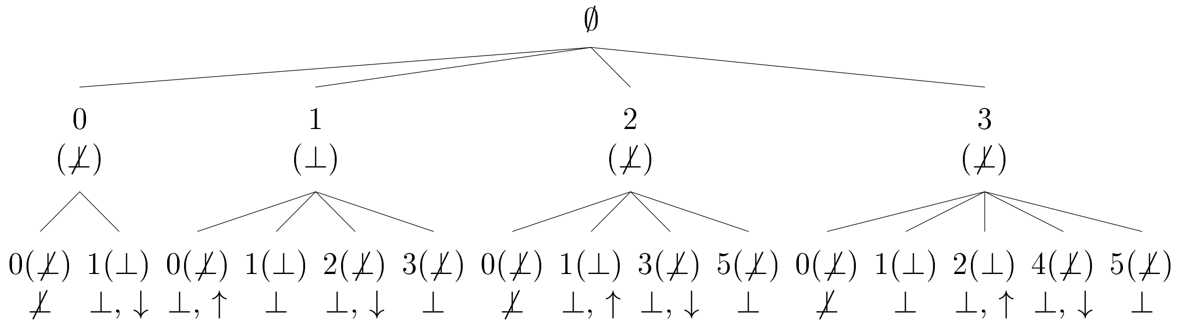


FIG. 6.1 – La 2-destinée réduite de $\langle \mathbb{N}, S, \perp \rangle$.

6.2 Présentation des cas à étudier pour construire la 3-destinée réduite

6.2.1 Notations

Les notations seront sensiblement les mêmes que pour la 2-destinée, à ceci près que pour la 3-destinée, on rajoute un niveau dans l'arbre, et donc nous aurons affaire à des nœuds de rang 1, 2, 3. Concernant un nœud de rang 3, on fera apparaître ses relations avec ses ascendants de la façon suivante :

- avec lui même : entre parenthèses (), à côté du nœud,
- avec son père : entre crochets [], juste en dessous,
- avec son grand-père : sans parenthésage, encore en dessous.

De plus, on prendra pour convention que sur une même branche, la lettre n désigne le nœud de rang 1, la lettre k désigne le nœud de rang 2 et la lettre u désigne le nœud de rang 3. Ces notations sont résumées dans la Figure 6.2.

6.2.2 Discussion préliminaire

On remarquera tout d'abord que si un nœud est successeur d'un autre, alors ils sont nécessairement premiers entre eux. On va séparer le cas où un nœud est proche d'un de

$$
\begin{array}{c}
\emptyset \\
| \\
n,\ (\{\perp, \not\perp\}) \\
| \\
k,\ (\{\perp, \not\perp\}) \\
[\{\perp, \not\perp\},\ \{\uparrow, \downarrow, \emptyset\}] \\
| \\
u,\ (\{\perp, \not\perp\}) \\
[\{\perp, \not\perp\},\ \{\uparrow, \downarrow, \emptyset\}] \\
\{\perp, \not\perp\},\ \{\uparrow, \downarrow, \emptyset\}
\end{array}
$$

FIG. 6.2 – Notations pour l'étude d'une 3-destinée de $\langle \mathbb{N}, S, \perp \rangle$.

ses ascendants (*i.e.* dans une relation de succession avec l'un de ses ascendants), et le cas où le nœud est éloigné de ses ascendants.

La démarche générale pour réaliser l'étude des sous-arbres de la 3-destinée exhaustive est la suivante :

- On commence par examiner les configurations que l'on peut rencontrer pour u par rapport à n et k, ce qui nous donne un ensemble de feuilles qui peuvent *potentiellement* apparaître dans le sous-arbre de k.
- Ensuite on examine, en fonction des relations entre le support de n et le support de k (où le support d'un entier est l'ensemble de ses diviseurs premiers), lesquelles de ces feuilles u apparaissent *effectivement*. Cela nous donne un certain nombre de sous-arbres de hauteur 2 pouvant apparaître *potentiellement* en dessous de n.
- Puis on regarde, en fonction de n, quels sont les sous-arbres qui apparaissent *effectivement*. Cela nous donne un certain nombre de sous-arbres de hauteur 3 pouvant apparaître *potentiellement* dans la 3-destinée réduite.
- Enfin, pour chacun de ces sous-arbres, on va chercher à savoir s'ils apparaissent *effectivement* dans la destinée : on va essayer soit de trouver un n qui correspond à ce sous-arbre (un témoin), soit de prouver que ce sous-arbre n'existe pas (aucun témoin).

  C'est cette dernière étape qui reste inachevée.

Afin d'étudier le comportement d'une feuille u par rapport à n et k, on doit séparer les cas où u admet une relation de succession avec l'un des deux ou les deux, et les cas où u est éloigné de k et n. On doit également séparer les cas où k est en relation de successeur avec n, et les autres. On dit que k est **éloigné de** n si $k \notin \{n-2, n-1, n, n+1, n+2\}$. Cela signifie que k ne peut pas être successeur ou prédécesseur de n ou d'un successeur ou prédécesseur de n.

On dit que u est **éloigné de** k si $u \notin \{k-1, k+1\}$, et **éloigné de** n si $u \notin \{n-1, n+1\}$. Cela signifie que u n'est ni successeur ni prédécesseur de k ou de n.

On suppose dans un premier temps que $n \geq 3$, et $k \geq 3$, pour ne pas faire apparaître de succession avec 0 et 1 dans la discussion générale (voir Paragraphe 6.2.5).

6.2.3 Cas où k est éloigné de n

On suppose que $k \notin \{n-2, n-1, n, n+1, n+2\}$. On n'a pas de relation de succession à examiner entre k et n, donc il s'agit simplement de discuter sur le fait qu'ils soient premiers entre eux.

6.2.3.1 Sous-cas où u est éloigné de n et de k

Il y a cinq possibilités pour u concernant la coprimarité avec lui-même, n et k :

1. $u = 0$ n'est premier ni avec u, ni avec k, ni avec n,
2. $u = 1$ est premier avec u, k et n,
3. u est non premier avec lui-même ($u \neq 1$), premier avec k et avec n,
4. u est non premier avec lui-même ($u \neq 1$), premier avec k et pas avec n,
5. u est non premier avec lui-même ($u \neq 1$), premier avec n et pas avec k.

Les deux premières feuilles apparaissent toujours, et au moins l'une des trois dernières. Il y a donc au moins trois feuilles "u éloigné de n et k", et au plus cinq. Les feuilles qui apparaissent dépendent de la position relative du support de n et du support de k. On note $Supp(n) = \{p_1, \ldots, p_r\}$ et $Supp(k) = \{q_1, \ldots, q_s\}$ ces supports respectivement. On distingue cinq cas :

I. $\mathbf{Supp(k) \subsetneq Supp(n)}$: Quatre feuilles sont présentes.
- Le cas 3 est satisfait par une solution du système de congruence : $x = 1\ [p_i]$.
- Le cas 4 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [q_i] \\ x = 0 & [p_j \notin Supp(k)]. \end{cases}$$

- Le cas 5 est impossible.

II. $\mathbf{Supp(n) \subsetneq Supp(k)}$: Quatre feuilles sont présentes.
- Le cas 3 est satisfait par une solution du système de congruence : $x = 1\ [q_i]$.
- Le cas 4 est impossible.
- Le cas 5 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [p_i] \\ x = 0 & [q_j \notin Supp(n)]. \end{cases}$$

III. $\mathbf{Supp(n) = Supp(k)}$: Trois feuilles sont présentes.
- Le cas 3 est satisfait par une solution du système de congruence : $x = 1\ [q_i]$.
- Les cas 4 et 5 sont impossibles.

IV. $\mathbf{Supp(n) \backslash Supp(k) \neq \emptyset}$ et $\mathbf{Supp(k) \backslash Supp(n) \neq \emptyset}$ et $\mathbf{Supp(n) \cap Supp(k) \neq \emptyset}$: Les cinq feuilles sont présentes.

– Le cas 3 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [q_i] \\ x = 1 & [p_i \notin Supp(k)]. \end{cases}$$

– Le cas 4 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [q_i] \\ x = 0 & [p_j \notin Supp(k)]. \end{cases}$$

– Le cas 5 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [p_i] \\ x = 0 & [q_j \notin Supp(n)]. \end{cases}$$

V. $\mathbf{Supp(n) \cap Supp(k) = \emptyset}$: Les cinq feuilles sont présentes.

– Le cas 3 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [q_i] \\ x = 1 & [p_i \notin Supp(k)]. \end{cases}$$

– Le cas 4 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [q_i] \\ x = 0 & [p_j \notin Supp(k)]. \end{cases}$$

– Le cas 5 est satisfait par une solution du système de congruence :

$$\begin{cases} x = 1 & [p_i] \\ x = 0 & [q_j \notin Supp(n)]. \end{cases}$$

On a donc cinq cas à différencier (I., II., III., IV. et V.), dépendant des positions respectives du support de n et du support de k, lorsque k est éloigné de n, afin de décrire les différentes configurations des feuilles u pour u éloigné de n et de k.

6.2.3.2 Sous-cas où u est proche de n ou de k

Il reste à regarder ce qu'il peut se passer lorsque $u \in \{n-1, n+1, k-1, k+1\}$. Chacun des éléments de cet ensemble peut être premier avec n ou non, et premier avec k ou non. On sait déjà que, étant données les relations de succession, on a :

- $k \pm 1 \perp k$,
- $n \pm 1 \perp n$.

Il reste donc à discuter des cas :

- $k \pm 1 \perp n$?
- $n \pm 1 \perp k$?

Pour chacun de ces quatre cas, on a deux possibilités (premiers entre eux, ou non premiers entre eux). Cela nous donne 16 configurations possibles pour le paquet de 4 feuilles "u proches de n ou k", représentés dans la Figure 6.3.

En combinant ces 16 cas avec les 5 possibilités que l'on avait pour les feuilles u éloignées de k et n, cela nous donne 80 cas à discuter concernant les sous-arbres de racine k éloigné de n, pour un n donné. Heureusement, nous le verrons en 6.3, beaucoup de ces cas n'apparaissent pas.

$\boxed{1}$					$\boxed{2}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\not\perp]$
	$\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$		$\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{3}$					$\boxed{4}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\not\perp]$
	$\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$		$\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{5}$					$\boxed{6}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\not\perp]$
	$\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$		$\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{7}$					$\boxed{8}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\not\perp]$
	$\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$		$\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{9}$					$\boxed{10}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\not\perp]$
	$\not\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$		$\not\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{11}$					$\boxed{12}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\not\perp]$
	$\not\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$		$\not\perp$	$\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{13}$					$\boxed{14}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\perp]$	$[\not\perp]$
	$\not\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$		$\not\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$
$\boxed{15}$					$\boxed{16}$				
	$k-1$	$k+1$	$n-1$	$n+1$		$k-1$	$k+1$	$n-1$	$n+1$
	$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\perp]$		$[\perp\uparrow]$	$[\perp\downarrow]$	$[\not\perp]$	$[\not\perp]$
	$\not\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$		$\not\perp$	$\not\perp$	$\perp\uparrow$	$\perp\uparrow$

FIG. 6.3 – Cas où u est proche de k ou n.

6.2.4 Cas où k est proche de n

Lors de l'étude des cas où $k \in \{n-2, n-1, n, n+1, n+2\}$, on constate qu'il se passe des choses différentes pour les sous-arbres en question, selon la forme de n. Les différentes formes de n différenciées sont présentées Figure 6.4 (voir [Cha00] pour une discussion plus détaillée).

$n = 2(2^\alpha + 1)$	et	$n \equiv 0[3]$	(pair)
$n = 2(2^\alpha + 1)$	et	$n \equiv 1[3]$	(pair)
$n = 2(2^\alpha - 1)$	et	$n \equiv 0[3]$	(pair)
$n = 2(2^\alpha - 1)$	et	$n \equiv 2[3]$	(pair)
$n = 2^\alpha$	et	$n \equiv 1[3]$	(pair)
$n = 2^\alpha$	et	$n \equiv 2[3]$	(pair)
$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$ et $n \neq 2^\alpha$	et	$n \equiv 0[6]$	(pair)
$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$ et $n \neq 2^\alpha$	et	$n \equiv 4[6]$	(pair)
$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$ et $n \neq 2^\alpha$	et	$n \equiv 2[6]$	(pair)
$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$	et	$n \equiv 3[6]$	(impair)
$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$	et	$n \equiv 1[6]$	(impair)
$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$	et	$n \equiv 5[6]$	(impair)

FIG. 6.4 – Formes de n à différencier pour l'étude des sous-arbres k proche de n.

6.2.5 Cas limites

L'étude des cas limites $k = 0, 1, 2$ n'apporte pas de nouvelles distinctions à faire sur les différentes formes de n par rapport à la Figure 6.4 (voir [Cha00] pour une discussion plus détaillée).

6.3 Etude des cas

On doit maintenant déterminer, pour chacun des cas :
- I. $Supp(k) \subsetneq Supp(n)$,
- II. $Supp(n) \subsetneq Supp(k)$,
- III. $Supp(k) = Supp(n)$,
- IV. $Supp(k) \backslash Supp(n) \neq \emptyset$ et $Supp(n) \backslash Supp(k) \neq \emptyset$ avec $Supp(n) \cap Supp(k) \neq \emptyset$,
- V. $Supp(k) \cap Supp(n) = \emptyset$,

lesquelles des configurations 1 à 16 (les configurations de la Figure 6.3) sont représentées dans le sous-arbre de n, en fonction de la forme de n. Cela consiste à rechercher des témoins k avec des conditions de coprimarité et de support par rapport à n correspondant à ces configurations.

6.3.1 Le cas III. $Supp(k) = Supp(n)$

Comme $k \pm 1 \perp k$, on a également $k \pm 1 \perp n$ et de même, comme $n \pm 1 \perp n$, on a également $n \pm 1 \perp k$ donc les cas 2 à 16 sont impossibles. Il reste à statuer sur le cas

1 ($k \pm 1 \perp n$ et $n \pm 1 \perp k$), et essayer de voir s'il existe un tel k. On voit que $k = n^2$ convient, donc l'étude de ce cas est terminée.

Les autres études de cas sont moins simples mais relèvent du même genre de discussion.

6.3.2 Le cas II. $Supp(n) \subsetneq Supp(k)$

Du fait de l'inclusion des supports, on a nécessairement $k \pm 1 \perp n$, ce qui rend les cas 5 à 16 impossibles. Pour chacun des cas 1 et 4, on trouve un témoin k qui lui correspond (voir dans [Cha00] ou [Gui01]), mais la discussion des cas 2 et 3 amène à faire la distinction entre les cas :

- $n + 1$ est une puissance de 2,
- $n - 1$ est une puissance de 2,
- n est impair et ni $n - 1$ ni $n + 1$ ne sont des puissances de 2.

Ces nouveaux critères sont à articuler avec les 12 précédemment trouvés (Paragraphe 6.4).

6.3.3 Le cas V. $Supp(k) \cap Supp(n) = \emptyset$

Si n est pair, alors on a $k \pm 1$ pair également, et donc les cas 1 à 12 sont impossibles. En revanche, pour les cas 13 à 16 on peut toujours trouver un témoin.

Si n est impair, alors on doit considérer tous les cas, et la discussion apporte un nouveau critère :

- n impair et n est une puissance d'un premier,
- n impair et n n'est pas une puissance d'un premier.

6.3.4 Le cas IV. $Supp(k) \backslash Supp(n) \neq \emptyset$ et $Supp(n) \backslash Supp(k) \neq \emptyset$ avec $k \not\perp n$

Ce cas n'est possible que si n n'est pas une puissance d'un nombre premier. La discussion des cas ne soulève pas de nouvelles distinctions à faire sur la forme de n.

6.3.5 Le cas I. $Supp(k) \subsetneq Supp(n)$

Ce cas n'est possible que si n n'est pas une puissance d'un nombre premier. Du fait de l'inclusion des supports, on a nécessairement $n \pm 1 \perp k$, donc seuls les cas 1, 5, 9 et 13 sont à considérer.

Les cas 1 et 9 sont entièrement réglés, mais une partie des cas 5 et 13 reste ouverte. On sait statuer pour certaines formes de n, mais pas pour d'autres.

On note Y l'ensemble des n vérifiant le cas 5 :

$$Y = \{n / \exists k \text{ tel que } Supp(k) \subsetneq Supp(n), k - 1 \perp n \text{ et } k + 1 \not\perp n\}.$$

De même on note X l'ensemble des n vérifiant le cas 13 :

$$X = \{n / \exists k \text{ tel que } Supp(k) \subsetneq Supp(n), k - 1 \not\perp n \text{ et } k + 1 \not\perp n\}.$$

Ces ensembles sont remarquables, car ils ont une caractérisation en terme d'ordre entre les facteurs premiers de n :

Proposition 5 (Caractérisation de Y)
Un entier n est un élément de Y si et seulement si n possède deux facteurs premiers distincts p et q tels que $Ord(p,q)$ (l'ordre de p modulo q) est pair.

Proposition 6 (Caractérisation de $X \cap (2\mathbb{N}+1)$)
Un entier impair n est un élément de X si et seulement si n possède un facteur premier p, et il existe deux entiers h et l a supports disjoints mais tous deux inclus dans $Supp(n)$, tels que $v_2(Ord(h,p)) \neq v_2(Ord(h,l))$ (où v_2 représente la valuation 2-adique).

Ces résultats sont dus à Marcel Guillaume, Denis Richard et Ji Lei yin, on peut les trouver dans [Cha00] et [Gui96] .

6.4 Recherche de témoins et problèmes restés ouverts

L'étude précédente nous amène à un maximum de 63 sous-arbres de racine n au plus dans la 3-destinée réduite de $\langle \mathbb{N}, S, \perp \rangle$. Pour chacun de ces sous-arbres, on a une condition sur la forme de n, et il reste à déterminer si un tel n existe (notamment en l'exhibant), ou bien prouver qu'un tel n ne peut pas exister. Les résultats de cette recherche de témoins apparaissent dans la Figure 6.5 (il manque les cas $n = 0$, 1, 2, 3, 4 et 6, traités à part).

Il y a donc dans la 3-destinée de $\langle \mathbb{N}, S, \perp \rangle$ au plus 61 sous-arbres de racine n, et au moins 56. Les témoins présentés dans la Figure 6.5 sont les plus petits que l'on a trouvés pour les formes de n concernées. On remarque qu'il faut parfois chercher très loin (par exemple $2^{227} - 1$).

Pour les cas restant indéterminés à l'heure actuelle, de nombreux tests ont été réalisés grâce aux tables de diviseurs de Cunningham ([CP]), mais tous les tests que j'ai pu mener ont été négatifs. Cela nous donne l'intuition que ces cas sont impossibles, mais il reste à le prouver.

Si un témoin existe, cela signifie que l'on peut exprimer l'existence de nombres entiers vraiment très grands à profondeur de quantification 3 sur le langage du successeur et de la coprimarité. Cette possibilité ne peut être complètement écartée, l'exemple $2^{227} - 1$ en est la preuve.

En résumé, les cinq cas restant à étudier sont les suivants :

n pair	$n = 2(2^\alpha - 1)$	$n \equiv 2[3]$	$\vert Supp(n)\vert = 2$		$14 = 2 \times 7$
			$\vert Supp(n)\vert > 2$	Y	$1022 = 2 \times 7 \times 73$
		$n \equiv 0[3]$	$\vert Supp(n)\vert > 2$		$30 = 2 \times 3 \times 5$
	$n = 2(2^\alpha + 1)$	$n \equiv 1[3]$			$10 = 2 \times 5$
		$n \equiv 0[3]$	$Supp(n) = \{2, 3\}$		$18 = 2 \times 3^2$
			$\vert Supp(n)\vert > 2$		$66 = 2 \times 3 \times 11$
	$n \neq 2(2^\alpha - 1)$ et $n \neq 2(2^\alpha + 1)$	$n \equiv 2[3]$	n primaire		$8 = 2^3$
			$\vert Supp(n)\vert \geq 2$	Y	$20 = 2^2 \times 5$
				$\neg Y$	$56 = 2^3 \times 7$
		$n \equiv 4[6]$	n primaire		$16 = 2^4$
			$\vert Supp(n)\vert \geq 2$	Y	$22 = 2 \times 11$
				$\neg Y$	$28 = 2^2 \times 7$
		$n \equiv 0[6]$	$Supp(n) = \{2, 3\}$		$12 = 2^2 \times 3$
			$\vert Supp(n)\vert > 2$		$42 = 2 \times 3 \times 7$
n impair	$n \equiv 5[6]$	$n = 2^\alpha + 1$	n primaire		5
			$\vert Supp(n)\vert = 2$	Y	$65 = 5 \times 13$
				$\neg Y$	**Ouvert**
			$\vert Supp(n)\vert > 2$	X	$16385 = 5 \times 29 \times 113$
				$Y \backslash X$	**Ouvert**
				$\neg Y$	**Ouvert**
		$n \neq 2^\alpha + 1$ et $n \neq 2^\alpha - 1$	n primaire		11
			$\vert Supp(n)\vert = 2$	Y	$35 = 5 \times 7$
				$\neg Y$	$155 = 5 \times 31$
			$\vert Supp(n)\vert > 2$	X	$455 = 5 \times 7 \times 13$
				$Y \backslash X$	$5225 = 5^2 \times 11 \times 19$
				$\neg Y$	$78275 = 5^2 \times 31 \times 101$
	$n \equiv 3[6]$	$n = 2^\alpha + 1$	n primaire		$9 = 3^2$
			$\vert Supp(n)\vert = 2$ (Y)		$33 = 3 \times 11$
			$\vert Supp(n)\vert > 2$ (Y)	X	$32769 = 3^2 \times 11 \times 331$
				$Y \backslash X$	**Ouvert**
		$n = 2^\alpha - 1$	$\vert Supp(n)\vert = 2$ (Y)		$15 = 3 \times 5$
			$\vert Supp(n)\vert > 2$ (X)		$255 = 3 \times 5 \times 17$
		$n \neq 2^\alpha + 1$ et $n \neq 2^\alpha - 1$	n primaire		$27 = 3^3$
			$\vert Supp(n)\vert = 2$	Y	$21 = 3 \times 7$
				$\neg Y$	$39 = 3 \times 13$
			$\vert Supp(n)\vert > 2$	X	$105 = 3 \times 5 \times 7$
				$Y \backslash X$	$435 = 3 \times 5 \times 29$
				$\neg Y$	$42627 = 3 \times 13 \times 1093$
	$n \equiv 1[6]$	$n = 2^\alpha - 1$	n primaire		7
			$\vert Supp(n)\vert = 2$	Y	$2047 = 23 \times 89$
				$\neg Y$	$2^{227} - 1$
			$\vert Supp(n)\vert > 2$	X	$32767 = 7 \times 31 \times 151$
				$Y \backslash X$	$33554431 = 31 \times 601 \times 1801$
				$\neg Y$	**Ouvert**
		$n \neq 2^\alpha + 1$ et $n \neq 2^\alpha - 1$	n primaire		13
			$\vert Supp(n)\vert = 2$	Y	$85 = 5 \times 17$
				$\neg Y$	$55 = 5 \times 11$
			$\vert Supp(n)\vert > 2$	X	$385 = 5 \times 7 \times 11$
				$Y \backslash X$	$1045 = 5 \times 11 \times 19$
				$\neg Y$	$15655 = 5 \times 31 \times 101$

FIG. 6.5 – Témoins et problèmes restés ouverts.

1. Existe-t-il un entier n tel que :
   (a) n est impair,
   (b) $n \equiv 2 \ [3]$,
   (c) $n = 2^\alpha + 1$, avec α de la forme $2^a p$ avec $a > 0$, et tel qu'on ait l'un des cas suivants :
   – $p = 1$ et $F_a = 2^{2^a} + 1$ a exactement deux facteurs premiers,
   – p et F_a sont tous les deux premiers,
   (d) $|Supp(n)| = 2$,
   (e) et n n'appartient pas à Y ?
2. Existe-t-il un entier n tel que :
   (a) n est impair,
   (b) $n \equiv 2 \ [3]$,
   (c) $n = 2^\alpha + 1$,
   (d) $|Supp(n)| \geq 3$
   (e) et n est dans $Y \backslash X$?
3. Existe-t-il un entier n tel que :
   (a) n est impair,
   (b) $n \equiv 2 \ [3]$,
   (c) $n = 2^\alpha + 1$,
   (d) $|Supp(n)| \geq 3$
   (e) et n n'appartient pas à Y ? (Ce cas est encore moins probable que le précédent.)
4. Existe-t-il un entier n tel que :
   (a) n est impair,
   (b) $n \equiv 0 \ [3]$,
   (c) $n = 2^\alpha + 1$, avec α, premier ou puissance d'un nombre premier, congru à 1 modulo 3,
   (d) $|Supp(n)| \geq 3$
   (e) et n est dans $Y \backslash X$?
5. Existe-t-il un entier n tel que :
   (a) n est impair,
   (b) $n \equiv 1 \ [3]$,
   (c) $n = 2^\alpha - 1$, avec α, premier ou puissance d'un nombre premier, congru à 1 modulo 4,
   (d) $|Supp(n)| \geq 3$
   (e) et n n'appartient pas à Y ?

6.5 Conclusion

La première leçon à tirer de cette recherche de la 3-destinée de $\langle \mathbb{N}, S, \perp \rangle$ est la suivante : on a beau savoir que l'ensemble des énoncés du langage $\{S, \perp\}$ de profondeur de quantification 3 qui sont vrais sur $\mathbb{N}$ est décidable, cela ne signifie pas pour autant que l'on est capable d'exhiber un algorithme de décision. Ici, on a 5 sous-arbres de la destinée qui peuvent être soit présents soit absents, ce qui nous donne au final 32 possibilités pour la 3-destinée. Une seule de ces possibilités est la bonne et nous donne l'algorithme. On a donc, en attendant mieux, un processus de décision partiel (on peut faire tourner les 32 algorithmes, et s'ils sont tous d'accord, on sait ce qu'il en est de l'énoncé testé).

Cette étude nous permet ensuite d'en apprendre beaucoup sur la structure elle-même. Ainsi, on voit qu'au rang de quantification 3, on est capable d'exprimer la parité de l'ordre d'un facteur premier d'un entier modulo un autre, ce qui n'est a priori pas évident.

Enfin, on peut constater que la recherche de destinées sur les structures arithmétiques est à même de faire émerger de nouveaux problèmes ouverts, ce qui renforce les liens déjà étroits entre, d'une part, le domaine de la logique mathématique et, d'autre part, celui de la théorie des nombres et de l'arithmétique.

Chapitre 7

Un algorithme de construction des destinées de $\langle \mathbb{N}, \leq \rangle$

Face à la complexité manifeste de l'exemple précédent, nous nous sommes attaqué à une structure plus simple, avec un langage à un seul prédicat binaire, la structure $\langle \mathbb{N}, \leq \rangle$. Cette structure a une théorie complète décidable, et le but de ce chapitre est de présenter un algorithme de décision qui utilise les destinées réduites de cette structure. Cet algorithme s'appuie sur deux sous-algorithmes :

- un algorithme de construction des destinées, qui, prenant en entrée la profondeur de quantification k, produit une k-destinée exhaustive essentielle de $\langle \mathbb{N}, \leq \rangle$;
- l'algorithme vu au Chapitre 4, qui à partir d'une k-destinée exhaustive et essentielle, est à même de décider tous les énoncés de profondeur de quantification $\leq k$.

Pour décrire le premier algorithme, nous avons besoin de déterminer précisément les bornes sur les fils et descendants d'un nœud définies au Chapitre 3, comme nous le verrons au Paragraphe 7.3. Les bornes sur les fils d'un nœud et les descendants d'un nœud sont calculées au Paragraphe 7.2. Les résultats présentés dans ce chapitre ont été obtenus en collaboration avec Maxim Vsemirnov[1].

7.1 Les 2-destinée et 3-destinée réduites de $\langle \mathbb{N}, \leq \rangle$

Pour connaître les bornes sur les fils des nœuds, on commence par s'intéresser à ce qui peut se passer pour des petites valeurs de la profondeur de quantification. Concernant les notations, on indiquera par un signe "$<$" si le nœud est inférieur à son père, par un signe "$=$" s'il lui est égal, et par un signe "$>$" s'il lui est supérieur, et on indiquera du bas vers le haut les relations du nœud avec son père, avec son grand-père, et ainsi de suite.

7.1.1 La 2-destinée réduite de $\langle \mathbb{N}, \leq \rangle$

L'ensemble $\mathbb{N}$ possède un minimum qui va jouer un grand rôle dans la structure. C'est le seul entier qui n'ait pas d'entier inférieur à lui-même. On a donc deux sous-arbres dans une 2-destinée exhaustive et essentielle de $\langle \mathbb{N}, \leq \rangle$: le sous-arbre de 0, qui comporte deux

[1] Steklov Institute, St. Petersbourg

branches (un entier est soit égal à 0, soit supérieur à 0), et le sous-arbre d'un autre entier $n \geq 1$, qui comporte trois branches (un entier est soit inférieur, soit égal, soit supérieur à n). Si on réduit une 2-destinée en prenant à chaque fois le plus petit représentant de chaque classe d'équivalence, cela nous donne la 2-destinée réduite de la Figure 7.1.

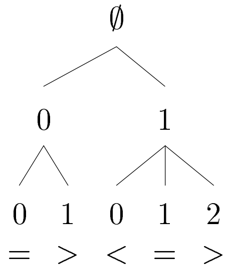


FIG. 7.1 – La 2-destinée réduite de $\langle \mathbb{N}, \leq \rangle$.

7.1.2 La 3-destinée réduite de $\langle \mathbb{N}, \leq \rangle$

Les sous-arbres des entiers $n \geq 3$ sont isomorphes au sous-arbre de 3. On a donc à considérer uniquement les sous-arbres de 0, 1, 2 et 3, qui sont non-isomorphes. Pour clarifier la présentation, nous présentons ces sous-arbres de façon séparée (*i.e.* sans la racine $\emptyset$), dans les Figures 7.2, 7.3, 7.4 et 7.5. Pour donner une idée de l'allure générale de la 3-destinée réduite, on l'a représentée Figure 7.6, sans faire apparaître les relations.

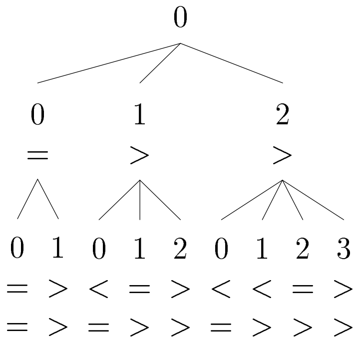


FIG. 7.2 – Le sous-arbre de 0 dans la 3-destinée réduite de $\langle \mathbb{N}, \leq \rangle$.

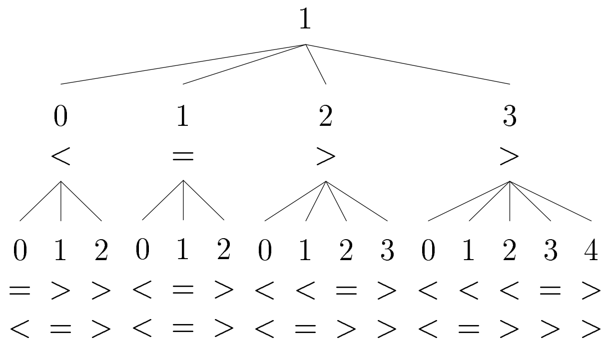


FIG. 7.3 – Le sous-arbre de 1 dans la 3-destinée réduite de $\langle \mathbb{N}, \leq \rangle$.

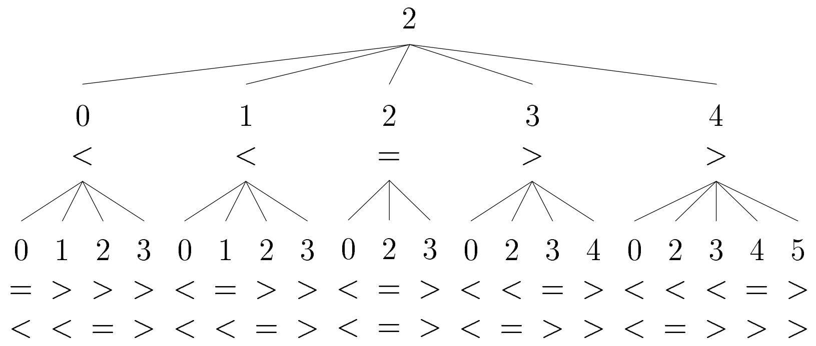


FIG. 7.4 – Le sous-arbre de 2 dans la 3-destinée réduite de $\langle \mathbb{N}, \leq \rangle$.

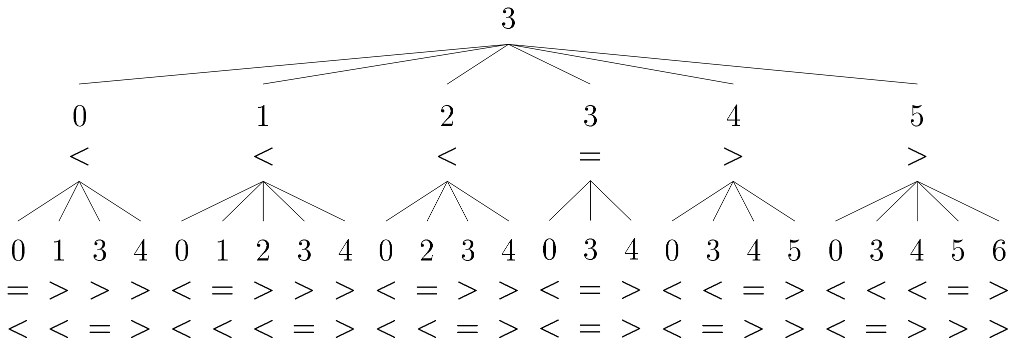


FIG. 7.5 – Le sous-arbre de 3 dans la 3-destinée réduite de $\langle \mathbb{N}, \leq \rangle$.

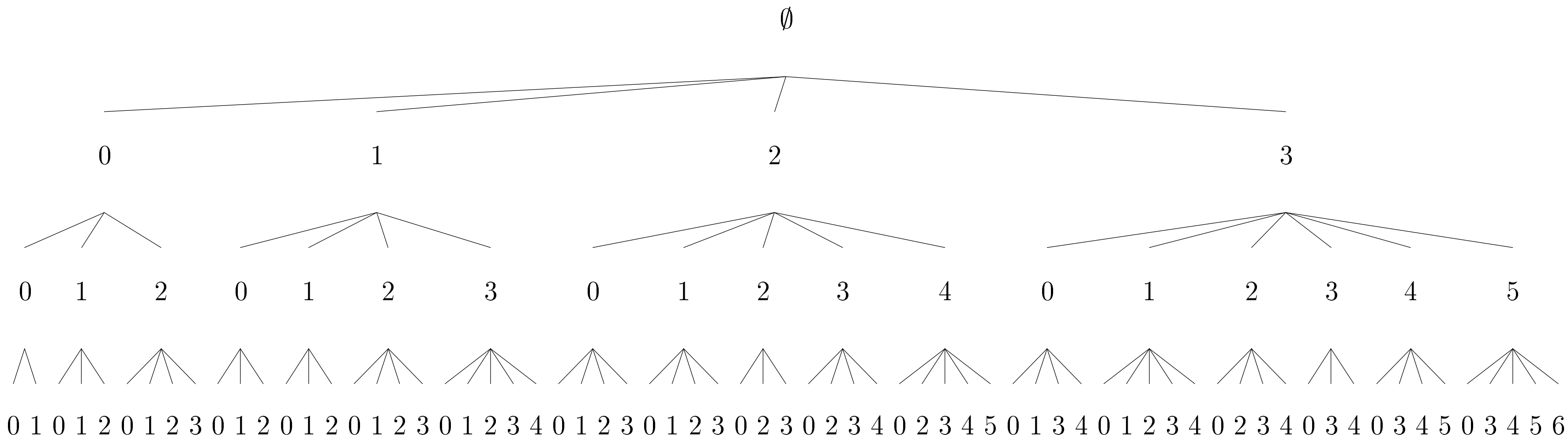


FIG. 7.6 – La 3-destinée réduite de $\langle \mathbb{N}, \leq \rangle$.

7.2 Bornes sur les fils et les descendants d'un nœud

On rappelle les définitions des bornes sur les fils ou descendants d'un nœud :

Soit $(\mathcal{X}, ||.||)$ une σ-structure normée. Soit $p \geq 1$ et T une p-destinée de $\mathcal{X}$. Soit x un nœud de rang $k < p$ de T. Soit $(u_i)_{i \in I}$ la famille des fils de x. La borne sur les fils de x est :

$$Sup_f(x) = \sup_{i \in I}(||l(u_i)||).$$

Soit $(v_j)_{j \in J}$ la famille des descendants de x. La borne sur les descendants de x est :

$$Sup_d(x) = \sup_{j \in J}(||l(v_j)||).$$

Si $k = p$:

$$Sup_d(x) = ||l(x)||.$$

Theorème 4
On fixe $p \geq 1$, et on considère la p-destinée réduite de $\langle \mathbb{N}, \leq \rangle$. Soit x_k un nœud de rang $k < p$ de cette destinée, d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$. Alors :

1. $Sup_f(x_k) = \max(l(x_k), \ldots, l(x_1)) + 2^{p-k-1}$, *et*
2. $Sup_d(x_k) = \max(l(x_k), \ldots, l(x_1)) + 2^{p-k} - 1$.

On prouve le Théorème 4 de la façon suivante :

- On commence par prouver, en utilisant des chaînes ordonnées d'ensembles de nœuds, que
$$Sup_d(x_k) \leq \max(l(x_k), \ldots, l(x_1)) + 2^{p-k} - 1.$$
C'est l'objet du Paragraphe 7.2.1.
- Puis on prouve, en utilisant une propriété des formules sur la structure $\langle \mathbb{N}, \leq \rangle$, que
$$Sup_f(x_k) \geq \max(l(x_k), \ldots, l(x_1)) + 2^{p-k-1}.$$
C'est l'objet du Paragraphe 7.2.2.
- Enfin, on utilise ces deux résultats partiels pour pouver le theorème, au Paragraphe 7.2.3.

7.2.1 Majorant de Sup_d dans la destinée réduite

On va montrer qu'on peut envoyer n'importe quelle p-destinée ($p \geq 1$) de la structure sur une p-destinée isomorphe dont les étiquettes des nœuds sont compris dans un segment de la forme $[0, \ldots, 2^{p-1} - 1]$. On introduit pour ce faire des ensembles de nœuds que l'on appellera MEBs (abbréviation non pas de “Microscope Électronique à Balayage”, mais de “Même Étiquette et Branche”).

Dans un premier temps, on ne travaille pas directement sur les p-destinées (on n'a pas besoin de la structuration), mais sur des q-arbres, définis ci-dessous. Le lien avec les p-destinées est établi au Paragraphe 7.2.1.6.

Définition 50 (q-arbre)
Un q-arbre est un arbre de degré fini dont les branches sont toutes de longueur q, et dont les nœuds sont étiquetés par des entiers positifs ou nuls. Si v est un nœud dans un q-arbre, on notera $l(v)$ son étiquette (plusieurs nœuds peuvent avoir la même étiquette).

7.2.1.1 Définition des ensembles MEBs

Afin d'obtenir un morphisme adéquat, on doit rassembler les nœuds qui ont la même étiquette et qui sont sur une même branche, car cette information doit être préservée tout au long des transformations que l'on fait subir à l'arbre. Au lieu de travailler sur les nœuds, on travaille donc sur des ensembles de nœuds appelés MEBs (Même Etiquette et Branche), et définis par induction sur la structure de l'arbre.

Définition 51 (MEBs dans un q-arbre)
Soit T un q-arbre.

1. *Si r est la racine de T, $S(r) = \{u$ nœud de T tel que $l(u) = l(r)\}$;*
2. *Pour tout nœud v de rang k, en supposant que les MEBs des nœuds de rangs inférieurs ont été définis :*
   - *Si v est un nœud de rang k et v appartient au MEB d'un nœud w, où w est un ancêtre de v, on a $S(v) = S(w)$;*
   - *Si v est un nœud de rang k et v n'appartient pas au MEB d'un de ses ancêtres, on définit :*

$$S(v) = \{u \text{ nœud du sous-arbre de racine } v \text{ tel que } l(u) = l(v)\}.$$

Remarque 14 *Chaque nœud appartient à exactement un MEB.*

Remarque 15 *Tous les nœuds d'un MEB noté S ont la même étiquette, on peut donc parler de l'étiquette de S comme étant l'étiquette commune des nœuds de S. On la note $l(S)$.*

Remarque 16 *Deux nœuds de la même branche qui ont la même étiquette sont nécessairement dans le même MEB.*

Remarque 17 *Deux MEBs distincts S_1 et S_2 peuvent avoir la même étiquette, s'ils sont inclus dans des sous-arbres disjoints.*

Remarque 18 *Lorsqu'un MEB noté S et une branche B s'intersectent, il existe un nœud de rang minimal dans cette intersection, et ce nœud est également le nœud de rang minimal dans le MEB. Ce nœud est appelé "racine du MEB", et on le note $r(S)$. Le MEB est entièrement inclus dans le sous-arbre de sa racine.*

7.2.1.2 Définition d'un ordre sur les MEBs

On commence par définir un successeur sur les MEBs :

Définition 52 (Successeur sur les MEBs)
Soit T un q-arbre, et S_1 et S_2 deux MEBs dans T. On définit $S_1 <^*_T S_2$ de la façon suivante :

1. *Il existe une branche B de T telle que $B \cap S_1 \neq \emptyset$ et $B \cap S_2 \neq \emptyset$*
2. *Pour toute branche B satisfaisant les conditions précédentes, on a :*
   (a) $\forall x \in S_1 \cap B,\ \forall y \in S_2 \cap B,\ l(x) < l(y)$.
   (b) *Il n'existe pas de nœud $z \in B$ tel que $l(x) < l(z) < l(y)$.*

Définition 53 (Ordre sur les MEBs)
On définit la relation $\prec$ comme la clôture transitive de $<^*$.

Lemme 2
La relation $\prec$ est un ordre partiel strict.

Preuve : La transitivité est donnée par la définition. L'antisymmétrie, l'irréflexivité et le caractère strict viennent du fait que si on a $S_1 \prec S_2$, alors on a nécessairement

$$l(S_1) < l(S_2).$$

$\square$

7.2.1.3 Préservation de la succession $<^*$ par passage au sous-arbre

Proposition 7
Soit T un q-arbre et v un nœud de T. Soient S_1 et S_2 deux MEBs de T tels que $S_1 \subset ST(v)$ (le sous-arbre de racine v) et $S_2 \subset ST(v)$. Alors :

1. *S_1 et S_2 sont des MEBs de $ST(v)$.*
2. *$S_1 <^*_T S_2 \Leftrightarrow S_1 <^*_{ST(v)} S_2$.*

Preuve :

1. Est une conséquence immédiate de la Définition 51.

2. $\Rightarrow$: La Condition 1 de la Définition 52 est préservée : si une branche B de T est telle que $B \cap S_1 \neq \emptyset$ et $B \cap S_2 \neq \emptyset$, puisque $S_1 \subset ST(v)$ et $S_2 \subset ST(v)$, on a $[B \cap ST(v)] \cap S_1 \neq \emptyset$ et $[B \cap ST(v)] \cap S_2 \neq \emptyset$, et par définition de $ST(v)$, $B \cap ST(v)$ est une branche de $ST(v)$.
   La Condition 2.a vient du fait que tous les nœuds concernés sont dans $ST(v)$.
   La Condition 2.b : si la condition est vraie dans T, elle est également vraie dans $ST(v)$.

   $\Leftarrow$: Laissé en exercice...

□

Corollaire 1
On peut noter $<^*$ le successeur sur les MEBs dans un q-arbre et ses sous-arbres, puisque c'est la même notion de successeur.

7.2.1.4 Majoration de la longueur des chaînes de MEBs

Lemme 3
Soit T un q-arbre. La longueur k d'une chaîne de MEBs $S_0 <^* S_1 <^* \ldots <^* S_{k-1}$ est inférieure ou égale à $2^q - 1$.

Corollaire 2
Il en va de même pour la longueur des chaînes $S_0 \prec \ldots \prec S_{k-1}$.

Preuve : (Du lemme) On procède par récurrence sur la hauteur de l'arbre q.

Pour $q = 1$ c'est clair.

Soit $S_0 <^* S_1 <^* \ldots <^* S_{k-1}$ une chaîne (C) de MEBs dans un $(q+1)$-arbre T.
On distingue les deux cas suivants :
Cas 1 : Si la racine de T n'est dans aucun MEB de la chaîne (C), alors tous ces MEBs sont inclus dans un unique sous-arbre de hauteur q de T. En effet, si on suppose qu'il existe un h tel que S_h est dans un sous-arbre de hauteur q et S_{h+1} dans un autre sous-arbre de hauteur q : puisque $S_h <^* S_{h+1}$, il existe une branche qui intersecte les deux MEBs, ce qui est impossible puisqu'une branche ne peut intersecter deux sous-arbres de hauteur q distincts en-dehors de la racine.
Donc tous les MEBs sont dans un même sous-arbre de hauteur q. Soit v sa racine. En utilisant la Proposition 7 et l'hypothèse de récurrence appliquée à (C) dans le sous-arbre de v, on a : $k \leq 2^q - 1 < 2^{q+1} - 1$.
Cas 2 : Si la racine de T appartient à un (et nécessairement un seul) des MEBs de (C) : soit S_h le MEB contenant la racine de T. Alors, de façon analogue à précédemment :

- $S_0 <^* S_1 <^* \ldots <^* S_{h-1}$ sont dans un même sous-arbre de hauteur q, donc $h \leq 2^q - 1$.
- $S_{h+1} <^* S_{h+2} <^* \ldots <^* S_{k-1}$ sont également dans un même sous-arbre de hauteur q, donc $k - h - 1 \leq 2^q - 1$.

Finalement, on obtient : $k \leq 2^q - 1 + 2^q - 1 + 1 = 2^{q+1} - 1$, ce qui termine la preuve.

□

7.2.1.5 Un lemme de projection général

Lemme 4
Soit $(\mathcal{P}, \prec)$ un ensemble partiellement ordonné tel que la longueur des chaînes est inférieure ou égale à M. Alors il existe un morphisme d'ensembles partiellement ordonnés $\varphi : (\mathcal{P}, \prec) \to ([0, \ldots, M-1], <)$.

Preuve : On procède par récurrence sur M. Pour $M = 1$, c'est trivial (les éléments de $\mathcal{P}$ ne sont pas comparables deux à deux).
Prouvons que l'hypothèse pour $M - 1$ implique l'hypothèse pour M : Soit $\mathcal{M}$ l'ensemble des éléments minimaux de $\mathcal{P}$. Pour tout $x \in \mathcal{M}$, on définit $\varphi_0(x) = 0$. On considère $(\mathcal{P} \backslash \mathcal{M}, \prec_{|\mathcal{P} \backslash \mathcal{M}})$. C'est un ensemble partiellement ordonné, et la longueur des chaînes est inférieure ou égale à $M - 1$. On applique l'hypothèse de récurrence à cet ensemble, et modulo une translation, on obtient un morphisme d'ensembles partiellement ordonnés : $\varphi' : \mathcal{P} \backslash \mathcal{M} \to [1, \ldots, M-1]$. On étend φ_0 et φ' en φ tel que $\varphi_{|\mathcal{P} \backslash \mathcal{M}} = \varphi'$ et $\varphi_{|\mathcal{M}} = \varphi_0 = 0$. Il est clair que φ est un morphisme qui convient.

□

7.2.1.6 Retour aux destinées

Dans le cas qui nous intéresse, les q-arbres vont correspondre aux sous-arbres dans une destinée de $\langle \mathbb{N}, \leq \rangle$. On va décrire un isomorphisme entre une p-destinée exhaustive et essentielle et la p-destinée réduite. Pour un nœud donné x_k d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$, on considère les MEBs du sous-arbre de racine x_k dont l'étiquette est supérieure aux étiquettes de $x_k, \ldots, x_1$. On applique, grâce au Lemme 4, un morphisme aux étiquettes de ces MEBs qui les envoie dans un intervalle $[\max(l(x_1), \ldots, l(x_k)) + 1, \ldots, M]$. Pour un MEB S, et $v \in S$, la nouvelle étiquette du nœud v devient $\varphi(v)$. On obtient alors un sous-arbre, dont la structure d'arbre est la même que celle du sous-arbre originel, mais dont les étiquettes des nœuds ont subi une transformation par le morphisme φ. Nous montrons que cette opération est un isomorphisme de sous-arbres dans une destinée. Comme la structure d'arbre est conservée, on montre simplement que l'ordre sur une branche est préservé, c'est l'objet de la Proposition 8.

Proposition 8
Soient u et v deux nœuds d'une même branche dans un sous-arbre d'une p-destinée. Soient S_1 et S_2 les MEBs de ce sous-arbre tels que $v \in S_1$ et $u \in S_2$. Si $l(v) < l(u)$, alors $S_1 \prec S_2$, et si $l(u) = l(v)$, alors $S_1 = S_2$.

Preuve : On procède par récurrence sur la différence $l(u) - l(v)$:

Si $l(u) = l(v)$, puisque u et v sont sur la même branche, ils appartiennent à un même MEB.

Si $l(u) > l(v)$, on considère les branches B' telle que $B' \cap S_1 \neq \emptyset$ et $B' \cap S_2 \neq \emptyset$. Chacune de ces branches vérifie : $x \in S_1 \cap B' \wedge y \in S_2 \cap B' \Rightarrow l(x) < l(y)$ car $l(S_1) < l(S_2)$.
Si elles satisfont toutes la Condition 2.b, alors $S_1 <^* S_2$, ce qui implique $S_1 \prec S_2$.
Sinon, il existe au moins une de ces branches, disons C, et un nœud $z \in C$, tels que $C \cap S_1 \neq \emptyset$ et $C \cap S_2 \neq \emptyset$, et pour tout $x \in C \cap S_1$ et tout $y \in C \cap S_2$, on a l'inégalité $l(x) < l(z) < l(y)$. Soit S_3 le MEB de z. On a d'une part $l(z) - l(x) < l(y) - l(x)$ et d'autre part $l(y) - l(z) < l(y) - l(x)$. On applique l'hypothèse de récurrence, ce qui nous donne $S_1 \prec S_3$ et $S_3 \prec S_2$. Par transitivité on obtient ce qu'on voulait.

□

7.2.1.7 Conclusion sur Sup_d dans une destinée réduite

Proposition 9
On considère un sous-arbre T dans une p-destinée de la structure $\langle \mathbb{N}, \leq \rangle$. Soit x_k la racine de ce sous-arbre et $x_{k-1}, \ldots, x_1, \emptyset$ ses ancêtres. On pose $m = \max(l(x_1), \ldots l(x_k))$. Le sous-arbre T est isomorphe à un sous-arbre dont les nœuds ont des étiquettes dans $[0, \ldots, m + 2^{p-k} - 1]$.

Preuve : On appelle u le nœud d'étiquette m qui est le plus bas sur le morceau de branche B allant de $\emptyset$ à la racine de T. On pose

$$\mathcal{P} = \{S \text{ est un MEB restreint au sous-arbre } T \text{ et } S(u) \prec S\}.$$

On remarque que :

1. On considère uniquement des MEBs dont l'étiquette est supérieure à m.
2. Comme $m = l(u)$ est la plus grande étiquette dans la branche B allant de $\emptyset$ à la racine de T, les nœuds du sous-arbre T qui ont une étiquette supérieure à m ne peuvent pas appartenir au MEB d'un des nœuds de cette portion de branche.

Ainsi, les éléments de $\mathcal{P}$ sont entièrement inclus dans T, donc la restriction faite dans la définition de $\mathcal{P}$ n'est pas une vraie restriction (ce sont aussi des MEBs dans la destinée). Cela nous permet de transformer les étiquettes des nœuds de ces MEBs sans changer le reste de la destinée, tout en conservant les relations d'ordre.

L'ensemble $\mathcal{P}$ est partitionné en sous-ensembles inclus dans des q-arbres, où $q = p - k$ (T est de hauteur $p - k + 1$ mais sa racine n'est dans aucun MEB de $\mathcal{P}$). Donc la longueur maximale des chaînes dans $\mathcal{P}$ est $2^{p-k} - 1$, d'après le Corollaire 2 du Lemme 3. D'après le Lemme 4 : il existe un morphisme $\varphi :\ (\mathcal{P}, \prec) \to ([m+1, \ldots, m+2^{p-k}-1], <)$.

On distingue deux cas :

1. Si v est un nœud de T dont l'étiquette est supérieure à m, on remplace cette étiquette par l'entier $\varphi(S(v))$, qui est dans l'intervalle $[m+1, \ldots, m+2^{p-k}-1]$.
2. Si v est un nœud de T dont l'étiquette est inférieure ou égale à m, on n'y touche pas.

D'après la propriété d'isomorphisme prouvée au Paragraphe 7.2.1.6, cette transformation est un isomorphisme entre le sous-arbre T et un autre sous-arbre dont les étiquettes sont dans l'intervalle $[0, \ldots, m+2^{p-k}-1]$.

□

Cette proposition est vraie en particulier pour la destinée réduite.

Corollaire 3 (Borne supérieure de Sup_d dans une destinée réduite)
On considère la p-destinée réduite de la structure $\langle \mathbb{N}, \leq \rangle$. Soit T un sous-arbre de racine x_k, d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$. On note $m = \max(l(x_1), \ldots, l(x_k))$. Alors

$$Sup_d(x_k) \leq m + 2^{p-k} - 1.$$

7.2.2 Minorant de Sup_f dans la destinée réduite

On se propose maintenant de trouver un minorant de Sup_f dans la destinée réduite. En utilisant ce minorant et le majorant de Sup_d trouvé au Paragraphe 7.2.1, on pourra donner une valeur exacte de Sup_f dans la destinée réduite. Pour établir la valeur du minorant de Sup_f dans la destinée réduite, on utilise la notion d'**écart** d'une formule à deux paramètres sur le langage $\{\leq\}$, définie au Paragraphe 7.2.2.1.

7.2.2.1 L'écart d'une formule à deux paramètres et l'écart maximum à profondeur de quantification q

On considère des formules sur le langage $\{\leq\}$ de profondeur de quantification q, à deux variables libres, et qui sont satisfaisables dans la structure $\langle \mathbb{N}, \leq \rangle$.

Définition 54 (Écart d'une formule à deux paramètres)
Soit $F(a,b)$ *une telle formule. On note :*

$$T(F) = \{(a,b) \in \mathbb{N}^2 / \mathbb{N} \models F(a,b)\}.$$

Cet ensemble est non-vide, puisque la formule est satisfaisable dans $\langle \mathbb{N}, \leq \rangle$*. Donc l'ensemble* $\{|b-a| / (a,b) \in T(F)\}$ *admet un minimum positif ou nul, noté* $e(F)$*, et qui constitue ce qu'on appelle l'***écart** *de la formule* F*.*

On s'intéresse maintenant à l'ensemble :

$$E_q = \{e(F)/F \text{ de prof. de quant. } q, \text{ à deux variables libres, satisfaisable dans } \langle \mathbb{N}, < \rangle\}.$$

Il existe un nombre fini de formules F de profondeur de quantification q à deux variables libres satisfaisables dans $\langle \mathbb{N}, \leq \rangle$ qui ne sont pas logiquement équivalentes. Comme deux formules équivalents ont le même écart (elles sont satisfaites par les mêmes couples), l'ensemble E_q est fini et admet donc un maximum noté e_q.

7.2.2.2 Minoration de l'écart maximum à profondeur de quantification q

On va montrer que pour tout q, on a : $e_q \geq 2^q$.

Pour cela, on construit par récurrence une suite de formules $(F_q(a,b))_{q \in \mathbb{N}}$ à deux variables libres et de profondeur de quantification q pour $F_q(a,b)$:

$$F_0(a,b) := (a \leq b) \wedge \neg(b \leq a) \text{ (cette formule est équivalente à "} a < b \text{")}.$$

$$\text{Pour tout } q \in \mathbb{N},\ F_{q+1}(a,b) := \exists x_{q+1}(F_q(a, x_{q+1}) \wedge F_q(x_{q+1}, b)).$$

On montre les propriétés suivantes sur ces formules :

Lemme 5
Pour tout q*, la formule* F_q *vérifie les propriétés suivantes :*

1. F_q *est satisfaisable dans* $\langle \mathbb{N}, \leq \rangle$ *;*
2. *Pour tout couple* $(a,b) \in \mathbb{N}^2$ *tel que* $\mathbb{N} \models F_q(a,b)$*, on a :*
   (a) $a + 2^q \leq b$,
   (b) *pour tout* $t \geq 0$, $\mathbb{N} \models F_q(t, t + (b-a))$.

Preuve : On procède par récurrence sur q.

C'est clair pour F_0.

Montrons que l'hypothèse de récurrence pour q implique l'hypothèse de récurrence pour $q+1$.

1. On doit d'abord montrer que la formule F_{q+1} est satisfaisable dans $\langle \mathbb{N}, \leq \rangle$: d'après l'hypothèse de récurrence, la formule F_q est satisfaisable, donc il existe deux entiers a et b tels que $\mathbb{N} \models F_q(a, b)$.
   On prend $t = b$ dans le point *2.(b)* de l'hypothèse de récurrence, ce qui nous donne $\mathbb{N} \models F_q(b, 2b - a)$.
   Alors $\mathbb{N} \models F_q(a, b) \wedge F_q(b, 2b - a)$, donc $\mathbb{N} \models \exists x_{q+1}(F_q(a, x_{q+1}) \wedge F_q(x_{q+1}, 2b - a))$, d'où $\mathbb{N} \models F_{q+1}(a, 2b - a)$.
   Cela prouve que F_{q+1} est satisfaisable dans $\langle \mathbb{N}, \leq \rangle$.
2. On suppose maintenant que a et b sont deux entiers tels que $\mathbb{N} \models F_{q+1}(a, b)$. Alors $\mathbb{N} \models \exists x_{q+1}(F_q(a, x_{q+1}) \wedge F_q(x_{q+1}, b))$.
   Donc il existe un entier $c \in \mathbb{N}$ tel que $\mathbb{N} \models F_q(a, c)$ et $\mathbb{N} \models F_q(c, b)$.
   Par hypothèse de récurrence, on a : $a+2^q \leq c$ et $c+2^q \leq b$, donc $a+2^{q+1} \leq c+2^q \leq b$, ce qui prouve le point *2.(a)*.
   Soit $t \geq 0$ un entier. Par hypothèse de récurrence, on a : $\mathbb{N} \models F_q(t, t + (c - a))$ et $\mathbb{N} \models F_q(t + (c - a), t + (c - a) + (b - c))$.
   On a donc $\mathbb{N} \models F_q(t, t + (c - a)) \wedge F_q(t + (c - a), t + (b - a))$, ce qui implique $\mathbb{N} \models \exists x_{q+1}(F_q(t, x_{q+1}) \wedge F_q(x_{q+1}, t + (b - a)))$.
   On a donc $\mathbb{N} \models F_{q+1}(t, t + (b - a))$, ce qui termine la preuve.

□

Corollaire 4
L'écart maximum à profondeur de quantification e_q vérifie $e_q \geq 2^q$ pour tout $q \in \mathbb{N}$.

Preuve : On fixe $q \geq 0$. D'après le point *2.(b)* du Lemme 5, on a $|b - a| \geq 2^q$ pour tout couple (a, b) tel que $\mathbb{N} \models F_q(a, b)$, donc $e(F_q) \geq 2^q$. Puisque $e_q \geq e(F_q)$, on a $e_q \geq 2^q$.

□

7.2.2.3 Conclusion sur Sup_f

Proposition 10
On considère une p-destinée T de la structure $\langle \mathbb{N}, \leq \rangle$, où $p \geq 2$. On considère un nœud qui n'est pas une feuille, et soient $x_k, \ldots, x_1, \emptyset$ respectivement le nœud et ses ascendants. On pose $m = \max(l(x_1), \ldots, l(x_k))$. On a :

$$Sup_f(x_k) \geq m + 2^{p-k-1}.$$

Preuve : On pose $q = p - k - 1$ et on considère la formule F_q. Puisque cette formule est satisfaite par un couple ordonné (a, b) avec un a choisi arbitrairement (d'après le Lemme 5), on a :

$$\mathbb{N} \models \forall z \exists y F_q(z, y)$$

On suppose que $m = l(x_h)$, pour un certain $h \in [1, \dots, k]$. Puisque les variables $z_1, \dots, z_{h-1}$, $z_{h+1}, \dots, z_k$ n'apparaissent pas dans $F_q(z_h, y)$, on a :

$$\mathbb{N} \models \forall z_1 \dots \forall z_h \dots \forall z_k \exists y F_q(z_h, y)$$

Cet énoncé voit sa forme destinale satisfaite dans T :

$$T \models \forall z_1 (P(\emptyset, z_1) \to \dots \forall z_h (P(z_{h-1}, z_h) \to \dots \forall z_k (P(z_{k-1}, z_k) \to \exists y P(z_k, y) \wedge \tilde{F}_q(z_h, y)) \dots).$$

Cela signifie que pour tout nœud z_k d'ascendants $z_{k-1}, \dots, z_1, \emptyset$, il existe un fils y de ce nœud tel que $T \models \tilde{F}_q(z_h, y)$. En particulier, pour le nœud x_k, il existe un tel fils y. Alors on a $T \models \tilde{F}_q(x_h, y)$. On peut retransposer cette satisfaction dans la structure $\langle \mathbb{N}, \leq \rangle$:

$$\mathbb{N} \models F_q(l(x_h), l(y)).$$

D'après le Lemme 5, l'écart de la formule F_q est supérieur ou égal à 2^q, donc on a : $l(y) - l(x_h) \geq e(F_q) \geq 2^q$.
Donc $Sup_f(x_k, \dots, x_1, \emptyset) \geq l(y) \geq l(x_h) + 2^{p-k-1} = m + 2^{p-k-1}$.

□

Cette proposition est en particulier valable pour la destinée réduite.

7.2.2.4 Minorant de Sup_d dans une destinée

On déduit du résultat précédent le corollaire suivant :

Corollaire 5
On fixe $p \geq 1$ et $k \in \{1, \dots, p\}$. On considère une p-destinée de la structure $\langle \mathbb{N}, \leq \rangle$. Soit $k < p$ et x_k un nœud d'ascendants $x_{k-1}, \dots, x_1, \emptyset$. On pose $m = \max(l(x_1), \dots l(x_k))$. On a :

$$Sup_d(x_k) \geq m + 2^{p-k} - 1.$$

Remarque 19 *On ne considère que des nœuds de rang $k < p$ car dans le cas où le nœud est de rang p (donc une feuille), on a, par définition :*

$$Sup_d(x_p) = l(x_p).$$

Preuve : (du corollaire) On procède par récurrence décroissante sur $k \geq 1$.

Cas $k = p - 1$: dans ce cas, $Sup_d(x_{p-1}) = Sup_f(x_{p-1})$ et comme

$$Sup_f(x_{p-1}) \geq m + 2^{p-k-1} = m + 1 = m + 2^{p-k} - 1,$$

on a le résultat voulu.

Montrons que l'hypothèse de récurrence pour $k+1$ implique l'hypothèse de récurrence pour k ($k \geq 1$). On considère un nœud x_k d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$ et on note $m = \max(l(x_1), \ldots, l(x_k))$. Posons y le fils de x_k qui réalise $l(y) = Sup_f(x_k)$. On a $Sup_d(x_k) \geq Sup_d(y)$. Par hypothèse de récurrence, on a :

$$Sup_d(y) \quad \geq \quad \max(l(y), l(x_k), \ldots, l(x_1)) + 2^{p-k-1} - 1. \tag{1}$$

D'après la proposition 10, on a :

$$l(y) = Sup_f(x_k) \quad \geq \quad m + 2^{p-k-1}, \tag{2}$$

on obtient donc :

$$\begin{aligned} Sup_d(x_k) &\geq Sup_d(y) \\ &\geq m + 2^{p-k-1} + 2^{p-k-1} - 1 \\ &\geq m + 2^{p-k} - 1. \end{aligned} \tag{3}$$

□

7.2.3 Conclusion

On va maintenant prouver le Théorème 4. On fixe $p \geq 1$ et on considère la p-destinée réduite de la structure $\langle \mathbb{N}, \leq \rangle$. D'après le Corollaire 3 et le Corollaire 5, on a pour tout nœud x_k de rang $k < p$ et d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$:

$$Sup_d(x_k) = m + 2^{p-k} - 1.$$

D'autre part, dans la preuve du Corollaire 5, l'addition des inégalités (1) et (2) dans cette preuve implique l'inégalité (3). Puisque les inégalités (1) et (3) sont en fait des égalités, l'inégalité (2) est également une égalité. On en déduit alors :

$$Sup_f(x_k) = m + 2^{p-k-1}.$$

7.3 Algorithme de construction des destinées

Maintenant qu'on dispose de la borne sur les fils des nœuds dans les destinées réduites de la structure, on peut mettre en place l'algorithme de construction suivant :

Entrée : p
Sortie : La p-destinée réduite de la structure $\langle \mathbb{N}, \leq \rangle$
`Début`

1. Création d'un $(p+1)$-arbre ayant "assez de nœuds"
2. Ajout des relations d'ordre sur les nœuds du $(p+1)$-arbre
3. Normalisation de la destinée obtenue (on ordonne les sous-arbres)
4. Elimination des redondances

`Fin`

7.3.1 Création d'un $(p+1)$-arbre ayant "assez de nœuds"

On veut construire un $(p+1)$-arbre de racine $\emptyset$ et dont les autres nœuds ont des étiquettes entières, tel que si on ajoute les listes de relations sur cet arbre (étape 2), on obtienne une p-destinée exhaustive de la structure. Pour cela, on utilise la valeur du plus grand fils d'un nœud dans la destinée réduite, et on construit le $(p+1)$-arbre de la façon suivante :

- On construit un 2-arbre de racine $\emptyset$ et dont les fils ont pour étiquettes les entiers entre 0 et 2^{p-1} ;
- Pour chacun de ces fils x_1, on lui ajoute comme fils des nœuds dont les étiquettes parcourent tous les entiers compris entre 0 et $l(x_1) + 2^{p-2}$;
- ...
- Si tous les nœuds de rang $k < p$ ont été construits, on construit le rang $k+1$ en ajoutant comme fils à chaque nœud de rang $k < p$ et d'ascendants $x_{k-1}, \ldots, \emptyset$ des nœuds dont les étiquettes parcourent tous les entiers entre 0 et $\max(l(x_k), \ldots, l(x_1)) + 2^{p-k-1}$.

7.3.2 Ajout des relations d'ordre sur les nœuds du $(p+1)$-arbre

On doit maintenant associer à chaque nœud de l'arbre précédent la liste de ses relations avec ses ascendants. Cette opération consiste à faire un parcours de l'arbre avec mise en mémoire des ascendants. Le test de l'ordre sur les entiers est une opération récursive peu coûteuse. Pour un nœud de rang k, la liste qui lui est associée est de longueur $k-1$.

Une fois cette opération réalisée, on dispose d'une p-destinée de la structure qui est exhaustive. Il nous reste à la rendre essentielle.

7.3.3 Normalisation de la destinée obtenue

Cette étape permet d'effectuer l'élimination des redondances. On établit un ordre sur les sous-arbres de sorte que les sous-arbres qui sont isomorphes aient leurs racines placées successivement dans la liste des fils de leur père commun. L'ordre choisi sur la liste des relations est arbitraire (par exemple, on décrète qu'une relation "$\leq$" est inférieure à une relation "$\geq$", puis on considère un ordre lexicographique sur les listes de relations), et deux sous-arbres T_1 et T_2 dont les racines ont un même père sont comparés de la façon suivante :

- Si la liste de relations de la racine de T_1 est inférieure à la liste de relations de la racine de T_2, le sous-arbre T_1 est inférieur au sous-arbre T_2 ;
- Si les listes de relations des racines de T_1 et T_2 sont identiques, et que la racine de T_1 a une étiquette inférieure à l'étiquette de la racine de T_2, alors le sous-arbre T_1 est inférieur au sous-arbre T_2.

Du fait de la construction de l'arbre considéré, deux sous-arbres dont les racines ont un même père et qui sont distincts ne peuvent avoir des racines de même étiquette. On vient donc de définir un ordre sur les sous-arbres.

On ordonne le $(p + 1)$-arbre pour cette relation. Grâce à cette opération, lorsque l'on parcours l'arbre en largeur, les sous-arbres isomorphes sont consécutifs et classés par ordre croissant de l'étiquette de la racine.

7.3.4 Elimination des redondances

On élimine les redondances sur la p-destinée normalisée de la manière suivante : on effectue un parcours de l'arbre, et à chaque fois que l'on considère un sous-arbre non isomorphe à ceux déjà sélectionnés, on l'ajoute à la liste des sélectionnés. On commence par les feuilles dans les sous-arbres de hauteur 2, puis les sous-arbres de hauteur 2 dans les sous-arbres de hauteur 3, et ainsi de suite. Comme au sein d'une même classe d'isomorphisme les sous-arbres sont ordonnés par valeur croissante de l'étiquette de leur racine, on obtient bien la p-destinée réduite de la structure.

7.4 Complexité de l'algorithme

On examine la complexité en espace en fonction de la hauteur p. On la note $T(p)$. Cela correspond à la taille de la p-destinée avant élimination des redondances.

On note n_k le nombre de nœuds de la destinée au rang k. On a :

$$\begin{aligned}
n_0 &= 1 \\
n_1 &= 2^{p-1} + 1 \\
n_2 &= \sum_{i_0=0}^{2^{p-1}} (i_0 + 2^{p-2}) = (2^{p-2} + 1)(2^{p-1} + 1) \\
n_3 &= \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{i_0+2^{p-2}} (\max(i_0, i_1) + 2^{p-3} + 1) \\
&\dots \\
n_k &= \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{i_0+2^{p-2}} \dots \sum_{i_{k-2}=0}^{\max(i_0,\dots,i_{k-3})+2^{p-k+1}} (\max(i_0, \dots, i_{k-2}) + 2^{p-k} + 1) \\
&\dots
\end{aligned}$$

On a d'autre part :

$$T(p) = n_0 + \sum_{k=1}^{p} k \times n_k$$

(chaque nœud de rang k est accompagné d'une liste de taille $k-1$).

Vérifions que l'on peut majorer chaque n_k par $(2^p - 1)^k$:

$$
\begin{aligned}
n_k &= \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{i_0+2^{p-2}} \cdots \sum_{i_{k-2}=0}^{\max(i_0,\ldots,i_{k-3})+2^{p-k+1}} (\max(i_0,\ldots,i_{k-2}) + 2^{p-k} + 1) \\
n_k &\leq \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{2^{p-1}+2^{p-2}} \cdots \sum_{i_{k-2}=0}^{\max(i_0,\ldots,i_{k-3})+2^{p-k+1}} (\max(i_0,\ldots,i_{k-2}) + 2^{p-k} + 1) \\
n_k &\leq \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{2^{p-1}+2^{p-2}} \cdots \sum_{i_{k-2}=0}^{2^{p-1}+\ldots+2^{p-k+1}} (\max(i_0,\ldots,i_{k-2}) + 2^{p-k} + 1) \\
n_k &\leq \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{2^{p-1}+2^{p-2}} \cdots \sum_{i_{k-2}=0}^{2^{p-1}+\ldots+2^{p-k+1}} (2^{p-1} + \ldots + 2^{p-k} + 1) \\
n_k &\leq \sum_{i_0=0}^{2^{p-1}} \sum_{i_1=0}^{2^{p-1}+2^{p-2}} \cdots \sum_{i_{k-2}=0}^{2^{p-1}+\ldots+2^{p-k+1}} (2^p - 1) \\
n_k &\leq \sum_{i_0=0}^{2^p-1} \sum_{i_1=0}^{2^p-1} \cdots \sum_{i_{k-2}=0}^{2^p-1} (2^p - 1) \\
n_k &\leq (2^p - 1)^k.
\end{aligned}
$$

En majorant n_k par $(2^p - 1)^k$, on obtient :

$$T(p) \leq 1 + \sum_{k=1}^{p} k \times (2^p - 1)^k \leq 2^{p^2}.$$

On a donc un majorant de la complexité en espace de cet algorithme de construction :

$$T(p) = \mathcal{O}(2^{p^2}).$$

La complexité en temps est du même ordre, car on réalise 4 parcours de l'arbre et les opérations d'ajout des listes de relations ne changent pas la majoration effectuée pour n_k.

7.5 Conclusion

Cet exemple nous fournit des méthodes qui peuvent éventuellement être réutilisées pour d'autres structures, par exemple l'emploi des MEBs, du lemme de projection ou l'écart d'une formule.

D'autre part, on remarque que le point crucial de l'algorithme de construction des destinées est de pouvoir calculer les bornes sur les fils des nœuds. Il est également nécessaire de pouvoir calculer les listes de relations de manière récursive. Une fois qu'on dispose de ces bornes sur les fils des nœuds, on peut construire un arbre suffisamment grand pour qu'une fois les listes de relations ajoutées, la destinée soit exhaustive. Les autres opérations sont récursives sous ces conditions.

Chapitre 8

Comparaison entre structures à destinées récursives et les structures H-bornées

Dans le chapitre précédent, nous avons décrit un exemple de structure pour laquelle il existe un algorithme de construction des destinées exhaustives et essentielles pour toute profondeur de quantification. Pour mettre en place cet algorithme, nous avons utilisé la borne sur les fils d'un nœud Sup_f afin de construire un arbre fini servant de support à une destinée exhaustive. L'algorithme de décision utilisant les destinées (décrit dans le Chapitre 4) consiste en une élimination des quantificateurs par une instanciation finie de chaque variable. Le nombre d'instanciations à effectuer est borné par Sup_f. Dans un souci de généralisation de cette démarche à d'autres structures, Nicole Schweikardt[1] nous suggéra de considérer des structures pour lesquelles on peut borner les quantifications, à savoir les structures H-bornées. La structure $\langle \mathbb{N}, \leq \rangle$ fait partie de la classe des structures H-bornées avec une fonction H récursive.

Nous généralisons donc ici les résultats du chapitre précédent, en établissant au Paragraphe 8.1 que si une structure est H-bornée pour une fonction H récursive, alors la borne Sup_f peut être calculée récursivement. Par conséquent les destinées de la structure sont récursives.

Le Paragraphe 8.2 a pour but de montrer que la classe des structures dont les destinées sont récursivement constructibles et la classe des structures H-bornées avec H récursive ne coïncident pas, mais que la première est plus large que la seconde. Pour cela, on exhibe une structure dont on peut construire récursivement une p-destinée pour tout $p \geq 1$ (sans utiliser la borne sur les fils d'un nœud), mais qui n'est H-bornée pour aucune fonction H récursive. Cet exemple a été rédigé suite à une suggestion de Youri Matiiassevitch[2].

---

[1]Institut für Informatik, Mainz
[2]Steklov Institute, St. Petersbourg

8.1 Les structures H-bornées

8.1.1 Définition des structures H-bornées

On peut trouver une définition plus générale des structures H-bornées dans [FR79]. Nous présentons ici la version de [Dub95] et [Mau94].

Définition 55 (Structure H-bornée)
On considère une structure $\mathcal{X}$ sur un langage relationnel fini σ, munie d'une norme $||.||$ à valeur dans $\mathbb{N}$. Soit $H : \mathbb{N}^3 \to \mathbb{N}$ une application telle que, pour tous entiers n, k, m, pour tout k-uple d'éléments de $\mathcal{X}$, $(a_1, \ldots, a_k) \in X^k$, tels que $\forall i \in \{1, \ldots, k\}$, on a $||a_i|| \leq m$, et pour toute formule $F(x_1, \ldots, x_k)$ de profondeur de quantification au plus n, on a :
Si $(\mathcal{X}, a_1, \ldots, a_k) \models \exists x_{k+1} F(x_1, \ldots, x_{k+1})$, alors il existe un élément $a_{k+1} \in X$, avec $||a_{k+1}|| \leq H(n, k, m)$, tel que $(\mathcal{X}, a_1, \ldots, a_k, a_{k+1}) \models F(x_1, \ldots, x_{k+1})$ (dans ce cas on écrit $(\mathcal{X}, a_1, \ldots, a_k) \models (\exists x_{k+1} \leq H(n, k, m)) F(x_1, \ldots, x_{k+1})$).
La structure $\mathcal{X}$ est alors dite H-bornée.

Ferrante et Rackoff ont décrit dans [FR79] un algorithme de décision dans les structures H-bornées, que nous rappelons en 8.1.2. Cet algorithme est une élimination des quantificateurs qui trouve sa justification dans le théorème suivant (démontré dans [FR79]) :

Theorème 5
On suppose que $\mathcal{X}$ est une σ-structure H-bornée. On fixe $n, k \in \mathbb{N}$ et soit $Q_1 x_1 \ldots Q_k x_k F(x_1, \ldots, x_k)$ un énoncé de profondeur de quantification au plus $n + k$ (c'est-à-dire que F est de profondeur de quantification au plus n). Soit $m_0, \ldots, m_k$ une suite d'entiers tels que $m_0 \leq \ldots \leq m_k$ et $\forall i \in \{1, \ldots, k\}, H(n + k - i, i - 1, m_{i-1}) \leq m_i$.
Alors $Q_1 x_1 \ldots Q_k x_k F(x_1, \ldots, x_k)$ est vrai dans $\mathcal{X}$ si et seulement si

$$(Q_1 x_1 \leq m_1) \ldots (Q_k x_k \leq m_k) F(x_1, \ldots, x_k)$$

est vrai dans $\mathcal{X}$.

Preuve : On prouve par récurrence sur $i \in \{1, \ldots, k + 1\}$ que

$$\mathcal{X} \models Q_1 x_1 \ldots Q_k x_k F(x_1, \ldots, x_k)$$

si et seulement si

$$\mathcal{X} \models (Q_1 x_1 \leq m_1) \ldots (Q_{i-1} x_{i-1} \leq m_{i-1}) Q_i x_i \ldots Q_k x_k F(x_1, \ldots, x_k).$$

Cas $i = 1$: clair.
Montrons, en utilisant l'hypothèse de récurrence pour $i - 1$, l'hypothèse de récurrence

pour $i \in \{1, \dots k\}$. On considère un $(i-1)$-uple $(a_1, \dots, a_{i-1})$ d'éléments de $\mathcal{X}$ tels que pour tout $j \in \{1, \dots, i-1\}$ on a $||a_j|| \leq m_j$. Puisque $\mathcal{X}$ est H-bornée, on a :

$$\begin{aligned}
&\mathcal{X} \models Q_i x_i [Q_{i+1} x_{i+1} \dots Q_k x_k F(a_1, \dots, a_{i-1}, x_i, \dots, x_k)] \Leftrightarrow \\
&\mathcal{X} \models (Q_i x_i \leq H(n, k-i, i-1, m_{i-1}))[Q_{i+1} x_{i+1} \dots Q_k x_k F(a_1, \dots, a_{i-1}, x_i, \dots, x_k)].
\end{aligned}$$

Notons que cela découle directement de la définition de structure H-bornée uniquement dans le cas où $Q_i = \exists$, mais il est facile de montrer que le cas $\forall$ découle également de la définition.

Comme $H(n+k-i, i-1, m_{i-1}) \leq m_i$, on a pour tout $(a_1, \dots, a_{i-1}) \in X^{i-1}$ tel que pour tout $j \in \{1, \dots, i-1\}$ on a $||a_j|| \leq m_j$:

$$\begin{aligned}
&\mathcal{X} \models Q_i x_i Q_{i+1} x_{i+1} \dots Q_k x_k F(a_1, \dots, a_{i-1}, x_i, \dots, x_k) \Leftrightarrow \\
&\mathcal{X} \models (Q_i x_i \leq m_i)[Q_{i+1} x_{i+1} \dots Q_k x_k F(a_1, \dots, a_{i-1}, x_i, \dots, x_k)],
\end{aligned}$$

donc

$$\begin{aligned}
&\mathcal{X} \models Q_1 x_1 \dots Q_k x_k F(x_1, \dots, x_k) \Leftrightarrow \\
&\mathcal{X} \models (Q_1 x_1 \leq m_1) \dots (Q_{i-1} x_{i-1} \leq m_{i-1}) Q_i x_i \dots Q_k x_k F(x_1, \dots, x_k) \text{ (HR) } \Leftrightarrow \\
&\mathcal{X} \models (Q_1 x_1 \leq m_1) \dots (Q_i x_i \leq m_i) Q_{i+1} x_{i+1} \dots Q_k x_k F(x_1, \dots, x_k),
\end{aligned}$$

ce qui achève la preuve.

□

8.1.2 Algorithme de décision "classique" dans les structures H-bornées

La mise en place de l'algorithme de décision dans les structures H-bornées nécessite des hypothèses supplémentaires :

- On suppose qu'il n'y a qu'un nombre fini d'éléments de $\mathcal{X}$ de norme inférieure à un entier donné ;
- On suppose qu'il est possible de les énumérer récursivement ;
- On suppose H récursive.

Les deux premières conditions sont vérifiées dans une structure de domaine $\mathbb{N}$ par exemple (en prenant pour norme l'identité de $\mathbb{N}$). L'algorithme de décision repose sur le fait que si $Q_1 x_1 \dots Q_k x_k F(x_1, \dots x_k)$ est un énoncé sous forme prénexe (F est sans quantificateur), et que $m_0 \leq \dots \leq m_k$ sont des entiers vérifiant $H(k-i, i-1, m_{i-1}) \leq m_i$ pour tout $i \in \{1, \dots, k\}$, le Théorème 5 stipule que décider cet énoncé équivaut à décider l'énoncé

$(Q_1x_1 \leq m_1)\ldots(Q_kx_k \leq m_k)F(x_1,\ldots x_k)$. Comme il n'y a qu'un nombre fini d'instanciations, on vérifie la formule pour chacune d'entre elles, ce qui revient à éliminer les quantifications. Soit $\mathcal{X}$ une σ-structure H-bornée sous ces conditions. L'algorithme de décision des énoncés dans cette structure se présente sous la forme suivante :

Entrées : Un σ-énoncé
Sorties : `VRAI` si l'énoncé est vrai dans la structure, `FAUX` sinon

1. Si l'énoncé n'est pas sous forme prénexe, on le met sous forme prénexe. Il est maintenant de la forme $Q_1x_1\ldots Q_kx_kF(x_1,\ldots x_k)$, avec F sans quantificateur.
2. On calcule une suite d'entiers $m_0 \leq \ldots \leq m_k$ tels que $H(k-i,i-1,m_{i-1}) \leq m_i$ pour tout $i \in \{1,\ldots,k\}$
3. Pour i allant de 1 à k faire
4. Si Q_i est un $\exists$
5. Alors Pour x_i de norme $\leq m_i$ faire
6. Evaluer $Q_{i+1}x_{i+1}\ldots Q_kx_kF(x_1,\ldots x_k)$ avec les valeurs de $x_1,\ldots,x_i$ déjà instanciées
7. Si cette valeur est `VRAI`, renvoyer `VRAI`.
8. Sinon Pour x_i de norme $\leq m_i$ faire
9. Evaluer $Q_{i+1}x_{i+1}\ldots Q_kx_kF(x_1,\ldots x_k)$ avec les valeurs de $x_1,\ldots,x_i$ déjà instanciées
10. Si cette valeur est `FAUX`, renvoyer `FAUX`.

8.1.3 Construction des p-destinées dans une structure H-bornée

L'algorithme de construction des destinées repose sur le théorème suivant :

Theorème 6
Soit $\mathcal{X}$ une σ-structure H-bornée. Soit $p \geq 1$. On considère T une p-destinée réduite de $\mathcal{X}$ et α_k un nœud de rang $k < p$ de cette destinée, d'ascendants $\alpha_{k-1},\ldots,\alpha_1,\emptyset$. On note $m = \max(||l(\alpha_k)||,\ldots,||l(\alpha_1)||)$. Alors

$$Sup_f(\alpha_k) \leq H(p-k-1,k,m).$$

Preuve : On s'appuie sur le Lemme 1, démontré au Chapitre 5. Soit α_k un nœud de rang $k \in \{0,\ldots,p-1\}$ et $\alpha_{k-1},\ldots,\alpha_1,\emptyset$ ses ascendants. Soit $m = \max(||l(\alpha_1)||,\ldots,||l(\alpha_k)||)$ ($m = 0$ si α_k est la racine $\emptyset$). Chaque fils α_{k+1} de α_k est l'unique représentant d'une classe de $(p-k)$-isomorphisme. Chacune de ces classes est caractérisée par une formule $F_{(\alpha_1,\ldots,\alpha_{k+1})}(x_1,\ldots,x_{k+1})$ de profondeur de quantification inférieure ou égale à $p-k-1$

(d'après le Lemme 1). Pour α_{k+1} fils de α_k fixé, on applique la définition d'une structure H-bornée pour la formule $\exists y F_{(\alpha_1,\ldots,\alpha_{k+1})}(\alpha_1,\ldots,\alpha_k,y)$. On sait que :

$$T \models \exists y(P(\alpha_k,y) \wedge \tilde{F}_{(\alpha_1,\ldots,\alpha_{k+1})}(\alpha_1,\ldots,\alpha_k,y))$$

donc

$$\mathcal{X} \models \exists y F_{(\alpha_1,\ldots,\alpha_{k+1})}(l(\alpha_1),\ldots,l(\alpha_k),y).$$

Donc, d'après la définition de structure H-bornée, on a :

$$\mathcal{X} \models (\exists y \leq H(p-k-1,k,m))F_{(\alpha_1,\ldots,\alpha_{k+1})}(l(\alpha_1),\ldots,l(\alpha_k),y).$$

Puisque l'on est dans une destinée réduite et que le nœud α_{k+1} a une étiquette de norme minimale dans sa classe d'équivalence, cela implique que pour chacun de ces nœuds α_{k+1}, on a : $||l(\alpha_{k+1})|| \leq H(p-k-1,k,m)$, et donc :

$$Sup_f(\alpha_k) \leq H(p-k-1,k,m).$$

□

8.1.4 Exemples

Sur la Figure 8.1 sont présentées quelques structures arithmétiques pour lesquelles on connaît une fonction H (la lettre c représente une constante), et un majorant de Sup_f.

Structure	Fonction $H(n,k,m)$	Majorant de $Sup_f(\alpha_k)$
$\langle \mathbb{N}, \leq \rangle$	$m + 2^n$	$m + 2^{p-k-1}$
$\langle \mathbb{Z}, +, <, 0 \rangle$	$(m+1) \times 2^{2^{2^{c(n+k)}}}$	$(m+1) \times 2^{2^{c(p-1)}}$
$\langle \mathbb{N}, \mid \rangle$	$m + 2^{\frac{1}{2}(n+k+1)(n+1)(2k+n+2)}$	$m + 2^{\frac{1}{2}p(p-k)(p+k+1)}$

FIG. 8.1 – Quelques exemples de structures H-bornées

On peut trouver ces exemples dans [FR79] et [Mic81]. Remarquons que dans le cas de la structure $\langle \mathbb{N}, \leq \rangle$, on trouve exactement la même borne qu'au Chapitre 7.

8.1.5 Conclusion

Le Théorème 6 nous dit que lorsque la structure est H-bornée, avec H récursive, on peut majorer la borne sur les fils d'un nœud par une fonction récursive. On peut donc construire un arbre fini en utilisant cette borne, comme dans le Chapitre 7, qui mène à une destinée exhaustive et essentielle après élimination des redondances (en fait, on obtient par ce procédé une destinée réduite de la structure). On voit à travers ce Théorème qu'il y a un lien très fort entre la fonction H et la borne sur les fils d'un nœud. Cependant, l'algorithme de construction des destinées qui utilise la borne sur les fils d'un nœud n'est pas nécessairement le seul algorithme utilisable pour construire les destinées d'une structure. C'est l'objet du paragraphe suivant. Nous reviendrons dans le Chapitre 9 sur les deux algorithmes de décision dans les structures H-bornées, l'algorithme de Ferrante et Rackoff et l'algorithme de Nézondet, notamment en ce qui concerne la complexité. Nous reviendrons également sur la comparaison entre structures H-bornées et structures à destinées récursives dans la Conclusion de cette partie.

8.2 Un exemple de structure non H-bornée à destinées récursives

L'exemple infirmant la réciproque du théorème "H-bornée avec H récursive $\Rightarrow$ p-destinées récursives" se fonde sur des résultats, dus à Mattiassavitch, liant ensembles indécidables de $\mathbb{N}$ et équations diophantiennes ou diophantiennes exponentielles. Ces résultats apparaissent dans [Mat95]. Nous utilisons une définition point par point d'un prédicat ternaire permettant de conserver des propriétés d'indécidabilité[3] tout en offrant la possibilité de construire une p-destinées essentielle et exhaustive pour tout $p \geq 1$. La théorie de cette structure est décidable, puisque l'on peut construire récursivement pour tout p une p-destinée exhaustive et essentielle.

Le présent paragraphe s'organise comme suit : on commence par présenter la structure, au Paragraphe 8.2.2, et nous définissons notamment deux propriétés que doit posséder cette structure : le caractère *pléthorique* des destinées (Paragraphe 8.2.2.1), et la propriété d'éloignement (Paragraphe 8.2.2.2). Ensuite, nous montrons que si la structure vérifie ces deux propriétés, alors elle ne peut pas être H-bornée (Paragraphe 8.2.3). Puis nous décrivons l'interprétation du prédicat ternaire sur le domaine de la structure (Paragraphe 8.2.4) et nous montrons que les deux propriétés (destinées pléthorique et éloignement) sont vérifiées dans cette interprétation. Enfin, nous montrons qu'on dispose d'un algorithme qui calcule, pour tout p, une p-destinée exhaustive et essentielle de la structure (Paragraphe 8.2.5).

8.2.1 Préliminaires

On se donne un ensemble indécidable d'entiers que nous noterons $\mathcal{M}$. Cet ensemble est dénombrable et nous noterons ses éléments $a_0, a_1, \ldots, a_n, \ldots$ par ordre croissant (l'opération d'énumération n'est bien évidemment pas récursive).

L'ensemble $\mathcal{M}$ est caractérisé par une équation diophantienne, c'est-à-dire qu'il existe un polynôme P à $k+1$ variables ($k \geq 2$) tel que :

$$a \in \mathcal{M} \Leftrightarrow \exists x_1, \ldots \exists x_k P(a, x_1, \ldots, x_k) = 0.$$

Nous faisons en outre les hypothèses suivantes :

– les variables $x_1, \ldots, x_k$ sont des variables assujetties au domaines $\mathbb{N}$ (leurs instances sont donc positives ou nulles) ;

[3] Cette indécidabilité est celle d'un sous-ensemble du domaine de la structure.

– si on choisit une caractérisation par une équation diophantienne exponentielle, on peut se restreindre à $k = 2$, c'est-à-dire à un “polynôme” (exponentiel) à trois variables :
$$a \in \mathcal{M} \Leftrightarrow \exists x_1 \geq 0 \ \exists x_2 \geq 0 \ Q(a, x_1, x_2) = 0.$$

Les résultats autorisant ces restrictions apparaissent dans [Mat95].

8.2.2 Présentation de la structure

Le domaine choisi est $\mathbb{Z}$, qui offre l'avantage d'être dénombrable, ce qui va simplifier notre construction, et de contenir suffisamment d'éléments pour nous permettre d'effectuer la construction de destinées que nous nommerons *pléthoriques*, décrites ci-après (8.2.2.1).

Le langage se compose d'un seul prédicat ternaire, noté $\mathcal{Q}$. L'interprétation de $\mathcal{Q}$ est présentée dans le paragraphe 8.2.4, et nous supposerons pour le moment que l'interprétation de $\mathcal{Q}$ est construite de sorte que :

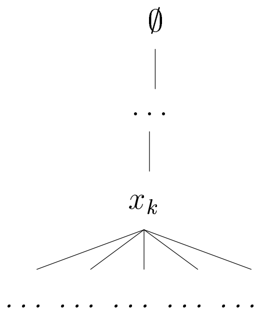
– les p-destinées exhaustives et essentielles sont “pléthoriques”. Nous entendons par là que tous les sous-arbres *possibles* (en un sens précisé plus bas) sont réalisés dans la destinée ;
– pour tout $a \in \mathcal{M}$, si x_1 et x_2 vérifient $\mathcal{Q}(a, x_1, x_2)$, alors x_1 et x_2 sont “plus loin”, en un sens que nous préciserons par la suite, que les racines du polynôme Q caractérisant l'appartenance de a à $\mathcal{M}$.

Précisons tout d'abord ces deux propriétés.

8.2.2.1 Les destinées pléthoriques

On définit la propriété “être pléthorique” pour une p-destinée de la façon suivante :

Définition 56 (Destinées pléthoriques)
Soit T une p-destinée de $\langle \mathbb{Z}, \mathcal{Q} \rangle$ ($p \geq 0$). La destinée T est dite **pléthorique** *si pour tout nœud x_k d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$ de la destinée, les fils de ce nœud sont tels que :*



– *si on considère tous les termes $\mathcal{Q}(a, b, c)$, où a, b, c sont dans $\{x_1, \ldots, x_k, y\}$ et au moins l'un des trois est y,*
– *que l'on considère ensuite toutes les conjonctions de tous ces termes avec une négation ou non devant, alors pour chacune de ces conjonctions il existe un fils du nœud x_k qui la vérifie.*

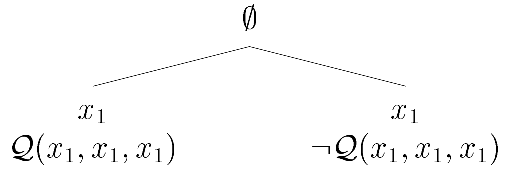
On étend facilement cette définition aux sous-arbres (un sous-arbre pléthorique est un sous-arbre dans une destinée pléthorique).

Exemple : Construisons la 2-destinée pléthorique "formelle" (sans instanciation des noms de variables).

La racine est $\emptyset$.

- On considère tous les termes $\mathcal{Q}(a, b, c)$ où a, b, c sont dans $\{y\}$ et au moins l'un des trois est y. Il n'y en a qu'un, c'est $\mathcal{Q}(y, y, y)$;
- Les "conjonctions" avec une négation ou non devant "les" termes sont au nombre de deux : $\mathcal{Q}(y, y, y)$ et $\neg\mathcal{Q}(y, y, y)$.

On obtient donc le rang 1 de la destinée :

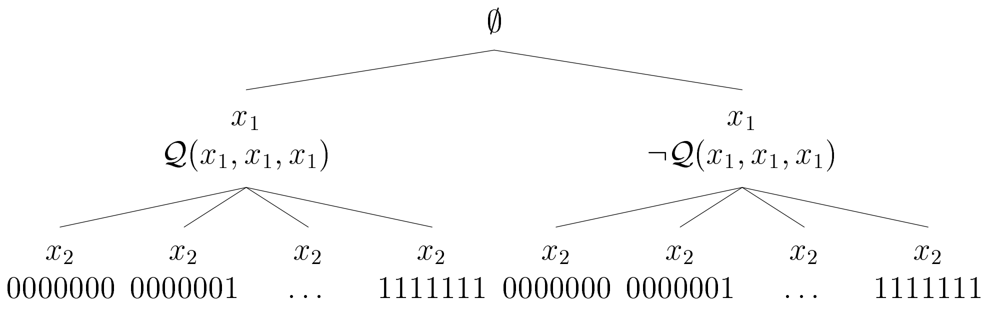


Pour chacun des deux nœuds (x_1) :

- On considère tous les termes $\mathcal{Q}(a, b, c)$ où a, b, c sont dans $\{x_1, y\}$ et au moins l'un des trois est y. Il y en a sept : $\mathcal{Q}(y, y, y)$, $\mathcal{Q}(y, y, x_1)$, $\mathcal{Q}(y, x_1, y)$, $\mathcal{Q}(x_1, y, y)$, $\mathcal{Q}(y, x_1, x_1)$, $\mathcal{Q}(x_1, y, x_1)$, $\mathcal{Q}(x_1, x_1, y)$;
- On considère toutes les conjonctions de ces termes avec une négation ou non : il y en a $2^7 = 128$, de la forme
$$\varepsilon_1\mathcal{Q}(y, y, y) \wedge \varepsilon_2\mathcal{Q}(y, y, x_1) \wedge \varepsilon_3\mathcal{Q}(y, x_1, y) \wedge \varepsilon_4\mathcal{Q}(x_1, y, y) \wedge$$
$$\varepsilon_5\mathcal{Q}(y, x_1, x_1) \wedge \varepsilon_6\mathcal{Q}(x_1, y, x_1) \wedge \varepsilon_7\mathcal{Q}(x_1, x_1, y),$$
où les ε_i sont soit $\neg$, soit rien.

Remarquons que chacune de ces conjonctions peut être codée par un nombre binaire à sept chiffres : $w_1 \dots w_7$ où $w_i = 1$ si et seulement si ε_i est "rien".

La 2-destinée pléthorique commence à devenir indessinable, chacun des nœuds (x_1) ayant 128 fils :



La taille de ces destinées pléthoriques croît exponentiellement.

8.2.2.2 Propriété d'éloignement pour les couples en relation avec $a \in \mathcal{M}$

Soit $a \in \mathcal{M}$. On considère les couples (x_1, x_2) de $\mathbb{N}^2$ tels que $Q(a, x_1, x_2) = 0$. Il en existe, puisque les racines de Q définissent l'ensemble $\mathcal{M}$. On note alors $R(a)$ l'ensemble de ces couples, qui est non vide :

$$R(a) = \{(x_1, x_2) \in \mathbb{N}^2 / Q(a, x_1, x_2) = 0\} \neq \emptyset.$$

On note $\|.\|$ la norme euclidienne sur $\mathbb{Z}^2$:

$$\begin{array}{ccc} \mathbb{Z}^2 & \to & \mathbb{N} \\ (u,v) & \mapsto & u^2+v^2 \end{array}$$

On pose alors $\mu(a) = \min(\|R(a)\|)$. On dit que le couple $(u,v) \in \mathbb{Z}^2$ est *éloigné* de l'origine par rapport aux racines de $Q(a,.,.)$ pour $a \in \mathcal{M}$ si $\|(u,v)\| \geq \mu(a)$.

La propriété que l'on exige de $\mathcal{Q}$ est alors la suivante :

Définition 57 (Propriété d'éloignement)
L'interprétation du prédicat $\mathcal{Q}$ dans $\mathbb{Z}$ vérifie la propriété d'éloignement si pour tout $a \in \mathcal{M}$, pour tout couple $(u,v) \in \mathbb{Z}^2$ tel que $\mathcal{Q}(a,u,v)$, on a :

$$\|(u,v)\| \geq \mu(a)$$

Remarque : Cela signifie que tous les couples (u,v) "proches" de l'origine vérifient $\neg\mathcal{Q}(a,u,v)$.

8.2.3 Cette structure n'est pas H-bornée

Montrons que si $\mathcal{Q}$ vérifie les propriétés suivantes :
- toute destinée de $\langle\mathbb{Z},\mathcal{Q}\rangle$ essentielle et exhaustive est pléthorique,
- et l'interprétation de $\mathcal{Q}$ vérifie la propriété d'éloignement,

alors la structure ne peut pas être H-bornée avec une fonction H récursive. Nous allons procéder par l'absurde et supposer que la structure est H-bornée avec H récursive. Cela va contredire l'indécidabilité de l'ensemble $\mathcal{M}$.

Pour tout $a \in \mathcal{M}$, la formule suivante est satisfaisable dans $\mathbb{Z}$, car le sous-arbre de a est pléthorique :

$$\exists x_1 \exists x_2 \ \mathcal{Q}(a,x_1,x_2).$$

D'après les propriétés du caractère H-borné, la formule suivante est également satisfaisable dans $\mathbb{Z}$:

$$\exists x_{1(|x_1|\leq H(1,1,a))} \exists x_{2(|x_2|\leq H(0,2,H(1,1,a)))} \ \mathcal{Q}(a,x_1,x_2),$$

qui implique cette troisième formule :

$$\exists (x_1,x_2)_{(\|(x_1,x_2)\|\leq H(1,1,a)^2+H(0,2,H(1,1,a))^2)} \ \mathcal{Q}(a,x_1,x_2).$$

Notons, par commodité : $f(a) = H(1,1,a)^2 + H(0,2,H(1,1,a))^2$. Cette fonction est récursive, puisqu'on a fait l'hypothèse que H est récursive. Par propriété d'éloignement, on a : pour tout couple (u,v) tel que $\mathcal{Q}(a,u,v)$, on a $\|(u,v)\| \geq \mu(a)$, donc $f(a) \geq \mu(a)$.

Par définition de $\mu(a)$, cela signifie donc qu'au moins l'un des couples $(n_1,n_2) \in \mathbb{N}^2$ tels que $Q(a,n_1,n_2) = 0$ vérifie $\|(n_1,n_2)\| \leq f(a)$. Autrement dit, on peut décider $a \in \mathcal{M}$ par l'algorithme :

Entrée : $a \in \mathbb{N}$
Sortie : La réponse à la question "a est-il dans $\mathcal{M}$?"
`Début`

1. Calculer $f(a)$
2. Pour tous les couples $(n_1, n_2) \in \mathbb{N}^2$ tels que $n_1^2 + n_2^2 \leq f(a)$ (il y en a un nombre fini)
3. Si $Q(a, n_1, n_2) = 0$ Renvoyer la réponse "$a \in \mathcal{M}$".
   Finsi
   Finpour
4. Renvoyer la réponse "$a \notin \mathcal{M}$".

`Fin`

Cet algorithme contredit l'indécidabilité de $\mathcal{M}$, donc la structure ainsi construite n'est pas H-bornée.

Il reste donc à définir l'interprétation de $\mathcal{Q}$ de manière à ce que les deux propriétés voulues (destinées pléthoriques et éloignement) soient vérifiées, ce que nous faisons au Paragraphe 8.2.4, puis à prouver au Paragraphe 8.2.5 qu'on peut calculer une p-destinée essentielle et exhaustive pour tout $p \geq 1$.

8.2.4 Interprétation du prédicat $\mathcal{Q}$

L'idée est de s'appuyer sur la construction progressive (pas forcément récursive) des sous-arbres de chaque élément, de manière à être sûr qu'ils soient pléthoriques, tout en faisant attention à la propriété d'éloignement dans les cas voulus, c'est-à-dire dans le cas où la racine du sous-arbre est un élément de $\mathcal{M}$. On sépare donc la construction en plusieurs étapes.

8.2.4.1 Interprétation sur $\mathbb{N}^3$

Sur $\mathbb{N}^3$, le prédicat $\mathcal{Q}$ correspond à l'appartenance à $\mathcal{M}$, dans le sens où pour tout $(a, b, c) \in \mathbb{N}^3$, on a :

$$\mathcal{Q}(a, b, c) \Leftrightarrow Q(a, b, c) = 0 \ \ (\text{donc } a \in \mathcal{M}).$$

Cette construction de l'interprétation n'est pas récursive.

8.2.4.2 Interprétation sur $\mathbb{N}_{-}^{*3}$

Sur cette partie de $\mathbb{Z}^3$, la construction effectuée est un processus diagonal, récursif :
- on construit le sous-arbre de hauteur 1 de -1
- on construit le sous-arbre de hauteur 1 de -2
- on construit le sous-arbre de hauteur 2 de -1

- on construit le sous-arbre de hauteur 1 de -3
- on construit le sous-arbre de hauteur 2 de -2
- on construit le sous-arbre de hauteur 3 de -1
- et ainsi de suite

de manière à ce qu'à chaque fois qu'on a besoin d'un "témoin" pour vérifier une conjonction de formules atomiques concernant un nœud, on prend le plus petit (en valeur absolue) qui puisse convenir : on tient compte des relations que l'on a définies auparavant, et dès qu'il y a un élément qui vérifie la conjonction voulue on le prend. Si personne ne convient, on prend le premier élément non encore parcouru pour servir de témoin.

Pour comprendre un peu mieux le principe de la construction, suivons les premières étapes :

Construction du sous-arbre de hauteur 1 de -1 : On pose $\mathcal{Q}(-1,-1,-1)$. On a défini toutes les relations de -1 avec lui-même.

Construction du sous-arbre de hauteur 1 de -2 : On pose $\neg\mathcal{Q}(-2,-2,-2)$. On a défini toutes les relations de -2 avec lui-même.

Construction du sous-arbre de hauteur 2 de -1 : Il faut remplir 2^7 situations. On fait donc appel à 128 nombres différents.

- type 0000000 : -2 ne convient pas car on a $\neg\mathcal{Q}(-2,-2,-2)$ qui contredit le premier 0. On prend donc -3, et par conséquent toutes les relations suivantes sont posées :
  $\mathcal{Q}(-3,-3,-3)$, $\mathcal{Q}(-3,-3,-1)$, $\mathcal{Q}(-3,-1,-3)$, $\mathcal{Q}(-1,-3,-3)$,
  $\mathcal{Q}(-3,-1,-1)$, $\mathcal{Q}(-1,-3,-1)$, $\mathcal{Q}(-1,-1,-3)$
- type 1000000 : -2 convient, donc on prend -2. Toutes les relations suivantes sont posées :
  $\neg\mathcal{Q}(-2,-2,-2)$, $\mathcal{Q}(-2,-2,-1)$, $\mathcal{Q}(-2,-1,-2)$, $\mathcal{Q}(-1,-2,-2)$,
  $\mathcal{Q}(-2,-1,-1)$, $\mathcal{Q}(-1,-2,-1)$, $\mathcal{Q}(-1,-1,-2)$
- type 0100000 : on prend -4
- et ainsi de suite.

On définit ainsi toutes les relations de -1 avec les éléments de l'ensemble $\{-2,\ldots,-129\}$ et de ces nombres chacun avec lui-même.

Construction du sous-arbre de hauteur 1 de -3 : -3 apparaît dans la liste des éléments dont on a déjà défini les relations avec eux-mêmes. Le sous-arbre de hauteur 1 de -3 est donc déjà construit (on sait que $\mathcal{Q}(-3,-3,-3)$).

Construction du sous-arbre de hauteur 2 de -2 : Il faut remplir 2^7 situations. On peut réutiliser certains des nombres $\{-1,-3,\ldots,-129\}$, sachant que les relations entre -1 et -2 sont déjà fixées. On définit les autres dans l'ordre, en tenant compte des relations des nombres avec eux-mêmes. Ici on est obligé d'aller chercher -130 car sinon on manque d'un élément de type $\neg\mathcal{Q}(.,.,.)$.

On poursuit cette construction "jusqu'à l'infini". S'il reste des triplets $(a,b,c)\in\mathbb{N}_{-}^{*3}$ sur lesquels on n'a pas défini $\mathcal{Q}$, on pose $\neg\mathcal{Q}(a,b,c)$.

8.2.4.3 Interprétation de $\mathcal{Q}$ lorsque le premier élément est positif et appartient à $\mathcal{M}$

On réalise une construction analogue à la constuction précédente, à ceci près qu'on pose d'entrée de jeu les contraintes suivantes : si $a \in \mathcal{M}$ et u, v sont des entiers tels que $\|(u,v)\| < \mu(a)$, alors on a $\neg\mathcal{Q}(a,u,v)$. Cette contrainte ne concerne pour chaque $a \in \mathcal{M}$ qu'un nombre fini de couples (u,v), il reste donc "suffisamment" d'entiers pour pouvoir effectuer la construction. On commence par construire le sous-arbre de hauteur 1 de a_0, puis le sous-arbre de hauteur 1 de a_1, le sous-arbre de hauteur 2 de a_0, le sous-arbre de hauteur 1 de a_2, le sous-arbre de hauteur 2 de a_1, le sous-arbre de hauteur 3 de a_0, et ainsi de suite, en veillant à ce qu'ils soient pléthoriques. Cette construction n'est pas récursive.

Remarque 20 *Le rang 1 dans les sous-arbres des a_i est déterminé par la construction du 8.2.4.1.*

De même que précédemment, s'il reste des triplets $(a,b,c) \in \mathcal{M} \times \mathbb{Z}^2$ pour lesquels l'interprétation de $\mathcal{Q}$ n'a pas été définie, on pose $\neg\mathcal{Q}(a,b,c)$.

8.2.4.4 Interprétation dans les autres cas

On construit les sous-arbres des nombres positifs qui ne sont pas dans $\mathcal{M}$ de la même façon : $\mathbb{N}\backslash\mathcal{M}$ est un ensemble dénombrable $\{b_0, b_1, \ldots, b_n, \ldots\}$. On construit le sous-arbre de hauteur 1 de b_0, puis le sous-arbre de hauteur 1 de b_1, le sous-arbre de hauteur 2 de b_0, le sous-arbre de hauteur 1 de b_2, le sous-arbre de hauteur 2 de b_1, le sous-arbre de hauteur 3 de b_0, et ainsi de suite, en veillant à ce qu'ils soient pléthoriques, puis on complète l'interprétation sur les couples $(a,b,c) \in (\mathbb{N}\backslash\mathcal{M}) \times \mathbb{Z}^2$ non atteints par $\neg\mathcal{Q}(a,b,c)$. Cette construction n'est pas récursive.

La construction ainsi effectuée de l'interprétation de $\mathcal{Q}$ possède alors les deux propriétés voulues (destinées pléthoriques et éloignement). En effet, toute p-destinée exhaustive et essentielle ressemblera à la p-destinée pléthorique "formelle" que nous avons construit jusqu'à la profondeur 2 dans le Paragraphe 8.2.2.1. D'autre part, la condition d'éloignement est imposée par construction au Paragraphe 8.2.4.3.

8.2.5 Construction de p-destinées exhaustives et essentielle

La définition de l'interprétation du prédicat $\mathcal{Q}$ nous indique un algorithme de construction immédiat d'une p-destinée exhaustive et essentielle de la structure $\langle\mathbb{Z}, \mathcal{Q}\rangle$. On suit en effet la construction de Paragraphe 8.2.4.2, en donnant comme fils à $\emptyset$ les nœuds de label -1 et -2. On utilise donc seulement la partie récursive de la construction de l'interprétation du prédicat $\mathcal{Q}$.

8.3 Conclusion

La classe des structures H-bornées avec H récursive est strictement incluse dans la classe des structures ayant des destinées récursives. Cependant, les destinées de l'exemple présenté au Paragraphe 8.2 sont très particulières, et cet exemple ne nous donne pas d'indication pour exhiber un algorithme général de construction de destinées n'utilisant pas la borne sur les fils d'un nœud. Une perspective de ce travail consiste donc à trouver un algorithme de construction de destinées pour une classe de structures non H-bornées et cependant décidables.

D'autre part, ce contre-exemple ne remet pas en cause l'intérêt de la comparaison entre les algorithmes de décision de Ferrante et Rackoff d'une part, et de Nézondet d'autre part, dans les structures H-bornées. C'est l'objet du Chapitre 9.

Chapitre 9

Comparaison entre l'algorithme de Nézondet et l'algorithme de Ferrante et Rackoff

9.1 Complexité de l'algorithme de Nézondet

Pour évaluer la complexité de la décision associée à l'algorithme de Nézondet, il est nécessaire de décomposer l'algorithme en deux étapes :

- un algorithme prenant comme entrée p et donnant une p-destinée exhaustive et essentielle (l'essentialité sert à minimiser le nombre de nœuds),
- un algorithme qui se sert d'une p-destinée exhaustive et essentielle, prenant en entrée un énoncé du langage σ de profondeur de quantification p et répondant à la question "Cet énoncé est-il vrai dans la structure ?".

L'étude de la complexité du second algorithme a déjà été effectuée au Paragraphe 4.4 du Chapitre 4. Le temps de calcul est en $\mathcal{O}(\Gamma(p) \times n \times N_p)$, où p est la profondeur de quantification de l'énoncé, $\Gamma(p)$ est la longueur des listes de relations associées aux feuilles, n est la taille de l'énoncé, et N_p est le nombre de nœuds de la p-destinée essentielle et exhaustive. Si on adopte le modèle de calcul des machines de Turing alternantes, nous avons également montré que le second algorithme appartient à la classe de complexité $ATIME(p, \mathcal{O}(n))$. Il nous reste donc à évaluer N_p, et le temps de calcul de la destinée de hauteur p.

9.1.1 Evaluation du nombre de nœuds dans la destinée en fonction de Sup_f

Soit $\mathcal{A}$ une structure vérifiant les mêmes hypothèses que pour la Proposition 11. Dans le pire des cas, les p-destinées exhaustives et essentielles sont "pleines", c'est-à-dire que l'on voit apparaître comme fils d'un nœud x_k, d'ascendants $x_{k-1}, \ldots, x_1, \emptyset$, tous les nœuds d'étiquettes ayant une norme inférieure ou égale à $Sup_f(x_k)$. On note K l'application qui à un entier donnée n associe le nombre d'éléments de A dont la norme est inférieure ou égale à n. On indice les nœuds de la destinée comme au Paragraphe 4.4 du Chapitre 4, à ceci près qu'on est en mesure de préciser les indices $M_{j_1,\ldots,j_{k-1},k}$. Plus exactement :

$$M_{j_1,\ldots,j_{k-1},k} = K(Sup_f(x_{j_1,\ldots,j_{k-2},k-1}).$$

On obtient donc une majoration du nombre total de nœuds dans une p-destinée exhaustive et essentielle :

$$N_p \leq \sum_{j_1=1}^{K(Sup_f(\emptyset))} \ldots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{K(Sup_f(x_{j_1,\ldots,j_{k-2},k-1}))} \ldots \sum_{j_{j_1,\ldots,j_{p-2},p-1}=1}^{K(Sup_f(x_{j_1,\ldots,j_{p-2},p-1}))} 1.$$

9.1.2 Dans le cas H-borné, expression de N_p en fonction de H

Lorsque la structure est H-bornée, on dispose d'une majoration de Sup_f par la fonction H, grâce au Théorème 6. On a donc, pour un nœud α de rang k :

$$Sup_f(\alpha) \leq H(p-k-1,k,m),$$

où m est le maximum des labels du nœud et de ses ancêtres. Cela nous donne une majoration pour le nombre de nœuds N_p :

$$N_p \leq \sum_{j_1=1}^{K(H(p-1,0,0))} \ldots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{K(H(p-k-1,k,m_{j_1,\ldots,j_{k-2},k-1}))} \ldots \sum_{j_{j_1,\ldots,j_{p-2},p-1}=1}^{K(H(0,p-1,m_{j_1,\ldots,j_{p-2},p-1}))} 1.$$

Exemple : Reprenons le cas de l'ordre sur les entiers naturels. Nous avons vu au Chapitre 7, que Sup_f a pour valeur $m + 2^{p-k-1}$, et dans ce cas est égal à $H(p-k-1,k,m)$. Cela nous donne pour le nombre de nœuds de la destinée réduite :

$$N_p \leq \sum_{j_1=1}^{2^{p-1}} \ldots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{m_{j_1,\ldots,j_{k-2},k-1}+2^{p-k-1}} \ldots \sum_{j_{j_1,\ldots,j_{p-2},p-1}=1}^{m_{j_1,\ldots,j_{p-2},p-1}+1} 1.$$

On a déjà majoré une somme semblable, dans le Chapitre 7, par $\mathcal{O}(2^{p^2})$.

9.1.3 Temps de calcul de l'algorithme de construction

Si on considère un modèle de calcul classique, l'algorithme de construction consiste à effectuer quatre parcours de la destinée exhaustive :

- un premier parcours pour la construire ;
- un deuxième parcours pour construire la liste des relations ;
- un troisième parcours pour normaliser la destinée ;
- un dernier parcours pour éliminer les redondances.

Avec cet algorithme, le temps de calcul de la p-destinée est de l'ordre de $\mathcal{O}(N_p \times T_1(p))$, où $T_1(p)$ est le temps mis pour vérifier dans la structure une formule atomique avec p variables.

Cependant, pour pouvoir comparer cette complexité à celle de l'algorithme de Ferrante et Rackoff, il nous faut déterminer la complexité de l'algorithme de construction des destinées en terme de machines de Turing alternantes. L'arbre de calcul associé à la construction de la destinée est de hauteur $(p + T_1(p))$. Il n'y a pas d'alternance de quantifications : l'algorithme de construction est dans $ATIME(0, p + T_1(p))$.

9.1.4 Complexité totale de l'algorithme de Nézondet

Supposons que l'on parte d'un énoncé de profondeur de quantification p, et que l'on doive construire une p-destinée exhaustive et essentielle de la structure avant de décider l'énoncé :

- Le premier algorithme prend un temps $\mathcal{O}(N_p) + T_1(p)$, où :

$$N_p \leq \sum_{j_1=1}^{K(H(p-1,0,0))} \cdots \sum_{j_{j_1,\ldots,j_{k-1},k}=1}^{K(H(p-k-1,k,m_{j_1,\ldots,j_{k-2},k-1}))} \cdots \sum_{j_{j_1,\ldots,j_{p-2},p-1}=1}^{K(H(0,p-1,m_{j_1,\ldots,j_{p-2},p-1}))} 1.$$

- Le deuxième algorithme prend un temps $\mathcal{O}(n \times N_p \times \Gamma_\sigma(p))$, où n est la longueur de la formule.

Dans le modèle de calcul des machines de Turing alternantes, cette décision est dans la classe $ATIME(p, \mathcal{O}(n) + p + T_1(p))$.

Si maintenant on suppose que la destinée est déjà construite, on retrouve la complexité du deuxième algorithme uniquement. Autrement dit, si nous devons décider un nombre M d'énoncés de profondeur de quantification inférieure ou égale à p, le temps total de décision de cet ensemble d'énoncés est de la forme :

- dans le modèle de calcul classique :

$$\mathcal{O}(N_p) + T_1(p) + \mathcal{O}(L \times N_p \times \Gamma_\sigma(p)),$$

  où L est la longueur totale des énoncés ;
- dans le modèle de calcul des machines de Turing alternantes :

$$ATIME(M \times p, \mathcal{O}(L) + p + T_1(p)).$$

9.2 Complexité de l'algorithme de Ferrante et Rackoff

Françoise Maurin présente dans [Mau94] une analyse de la complexité de l'algorithme de Ferrante et Rackoff qui fait intervenir le modèle de calcul des machines de Turing alternantes.

Proposition 11 (Complexité de l'algorithme de Ferrante et Rackoff)
Soit $\mathcal{A}$ une σ-structure, avec σ langage relationnel fini. On suppose que :

- *la structure $\mathcal{A}$ est normée et H-bornée (avec H récursive et vérifiant les restrictions présentées au Chapitre 8, Paragraphe 8.1.2),*
- *pour toute σ-formule sans quantificateur à k variables libres $F(x_1, \ldots, x_k)$, et tout k-uple d'éléments de A, $(a_1, \ldots, a_k)$, il existe une machine de Turing qui décide $\mathcal{A} \models F(a_1, \ldots, a_k)$ en temps $T_1(n)$, où n est la taille de la formule F.*
- *le n-uple d'entiers $(m_1, \ldots, m_n)$ satisfait, pour tout $i \in \{1, \ldots, n-1\}$, $H(n + k - i, i - 1, m_i) \leq m_{i+1}$.*

Alors un énoncé de taille n peut être décidé dans la structure $\mathcal{A}$ en temps $ATIME(n, T(n))$, où :

$$T(n) = 3n \log(n) + T_1(n \log(n) + \sum_{i=1}^{n} m_i).$$

Il paraît difficile a priori de comparer les deux calculs de complexité. En effet, dans l'algorithme de Ferrante et Rackoff, l'énoncé subit une mise sous forme prénexe qui augmente la profondeur de quantification. En fait, cette mise sous forme prénexe est inutile, car on peut réécrire le Théorème 5 pour une formule quelconque. On définit pour cela une relativisation des quantificateurs grâce à la fonction H, de la manière suivante : soit F une formule à n variables libres notées $y_1, \ldots, y_n$ et k variables liées notées $x_1, \ldots, x_k$ dans l'ordre de profondeur de quantification. Soient $m_0, \ldots, m_k$ une suite d'entiers tels que $m_0 \leq \ldots \leq m_k$ et pour tout $i \in \{1, \ldots, k\}$, on ait $H(n+k-i, i-1, m_{i-1}) \leq m_i$. La forme relativisée de F, que l'on notera $\bar{F}$, est la formule F dans laquelle chaque quantification $Q_k x_k G(y_1, \ldots, y_n, x_1, \ldots, x_k)$ devient $(Q_k x_k \leq m_k) G(y_1, \ldots, y_n, x_1, \ldots, x_k)$. On remarque aisément que les opérations booléennes sont compatibles avec cette relativisation.

On peut alors démontrer une version adaptée du Théorème 5 :

Theorème 7
On suppose que $\mathcal{X}$ est une σ-structure H-bornée. On fixe $n, k \in \mathbb{N}$ et soit $F(y_1, \ldots, y_n)$ une formule de profondeur de quantification k, à n variables libres. Soit $(a_1, \ldots, a_n)$ un n-uples d'éléments de $\mathcal{X}$, et $m_0, \ldots, m_p$ une suite d'entiers tels que $m_0 = \max(||a_1||, \ldots, ||a_n||)$, $m_0 \leq \ldots \leq m_p$ et $\forall i \in \{1, \ldots, p\}, H(p - i, i - 1, m_{i-1}) \leq m_i$. Alors :

$$\mathcal{X} \models F(a_1, \ldots, a_n)$$

si et seulement si

$$\mathcal{X} \models \bar{F}(a_1, \ldots, a_n).$$

Preuve : On procède par récurrence sur k.

Si $k = 0$: soit $F(y_1, \ldots, y_n)$ une σ-formule sans quantificateur. On a $\bar{F}(y_1, \ldots, y_n) = F(y_1, \ldots, y_n)$, donc ce cas est trivial.

Montrons que l'hypothèse au rang k implique l'hypothèse au rang $k+1$: soit $F(y_1, \ldots, y_n)$ une σ-formule de profondeur de quantification $k+1$. Soit $(a_1, \ldots, a_n)$ un n-uple d'élements du domaine de $\mathcal{X}$. Puisque la relativisation des quantifications est compatible avec les opérations booléennes, on peut se restreindre au cas où F est de la forme $Q_1 x_1 G(y_1, \ldots, y_n, x_1)$, avec G une σ-formule de profondeur de quantification k.

On suppose que $Q_1 = \exists$ et

$$\mathcal{X} \models F(a_1, \ldots, a_n).$$

Alors, d'après la définition d'une structure H-bornée, on a :

$$\mathcal{X} \models (\exists x_1 \leq H(n, k, m_0))G(a_1, \ldots, a_n, x_1),$$

donc

$$\mathcal{X} \models (\exists x_1 \leq m_1)G(a_1, \ldots, a_n, x_1).$$

Cela signifie qu'il existe un a_{n+1} tel que $||a_{n+1}|| \leq m_1$ et

$$\mathcal{X} \models G(a_1, \ldots, a_n, a_{n+1}).$$

On applique l'hypothèse de récurrence à la formule G (la suite $m_1, \ldots, m_k$ convient), ce qui nous donne :

$$\mathcal{X} \models \bar{G}(a_1, \ldots, a_n, a_{n+1}).$$

On peut alors écrire, puisque $||a_{n+1}|| \leq m_1$:

$$\mathcal{X} \models (\exists x_1 \leq m_1)\bar{G}(a_1, \ldots, a_n, x_1),$$

c'est-à-dire :

$$\mathcal{X} \models \bar{F}(a_1, \ldots, a_n).$$

Inversement, si :

$$\mathcal{X} \models \bar{F}(a_1, \ldots, a_n),$$

alors

$$\mathcal{X} \models (\exists x_1 \leq m_1)\bar{G}(a_1, \ldots, a_n, x_1),$$

ce qui implique qu'il existe un a_{n+1} tel que $||a_{n+1}|| \leq m_1$ et

$$\mathcal{X} \models \bar{G}(a_1, \ldots, a_n, a_{n+1}).$$

On applique l'hypothèse de récurrence (mais dans l'autre sens), ce qui nous donne :

$$\mathcal{X} \models G(a_1, \ldots, a_n, a_{n+1}).$$

On peut alors écrire :

$$\mathcal{X} \models (\exists x_1 \leq m_1)G(a_1, \ldots, a_n, x_1),$$

qui implique bien évidemment :

$$\mathcal{X} \models \exists x_1 G(a_1, \ldots, a_n, x_1),$$

autrement dit :

$$\mathcal{X} \models F(a_1, \ldots, a_n).$$

Dans le cas où $Q_1 = \forall$, on se ramène au cas précédent en considérant $\neg F$.

$\square$

En adaptant l'algorithme de Ferrante et Rackoff à n'importe quel énoncé (pas forcément prénexe), on obtient une complexité en

$$ATIME(p, \mathcal{O}(n) + T_1(p)),$$

où p est la profondeur de quantificateur de l'énoncé, n est la longueur de l'énoncé, et T_1 est le temps mis pour vérifier dans la structure une formule atomique avec p variables.

On peut alors comparer les complexités de l'algorithme de Ferrante et Rackoff, et de l'algorithme de Nézondet. Le résultat apparaît Figure 9.1.

Énoncés de profondeur p	Algorithme de Ferrante et Rackoff	Algorithme de Nézondet
Un énoncé de longueur n	$ATIME(p, \mathcal{O}(n) + T_1(p))$	$ATIME(p, \mathcal{O}(n) + p + T_1(p))$
M énoncés de longueur totale L	$ATIME(M \times p, \mathcal{O}(L) + M \times T_1(p))$	$ATIME(M \times p, \mathcal{O}(L) + p + T_1(p))$

FIG. 9.1 – Comparaison des complexités de l'algorithme de Ferrante et Rackoff et de l'algorithme de Nézondet.

9.3 Conclusion

L'analyse du tableau précédent nous conduit aux trois remarques suivantes :

1. Tout d'abord, la complexité des deux algorithmes n'est pas fondamentalement différente. Ce n'est pas suprenant, étant donné que l'on élimine les quantificateurs de la même façon, à savoir en bornant les quantifications (avec les mêmes bornes).
2. Ensuite, on peut remarquer que dans le cas où l'on a beaucoup d'énoncés à vérifier, l'algorithme de Nézondet devient vite plus intéressant. En effet, le calcul de la satisfaction des formules atomiques sur les éléments de la structure est déjà effectué lors de la construction de la p-destinée exhaustive et essentielle. On n'effectue donc ce calcul qu'une fois, et on se sert de la structure de donnée finie représentant cette

p-destinée, stockée une fois pour toute, pour évaluer tous les énoncés. Dans l'algorithme de Ferrante et Rackoff, on instancie au fur et à mesure, et on évalue les formules atomiques dont on a besoin, pour chaque énoncé à vérifier. Ce n'est donc pas la même philosophie de vérification : l'algorithme de Ferrante et Rackoff est un système "paresseux", au sens informatique du terme (on évalue au fur et à mesure des besoins), tandis que l'algorithme de Nézondet est un algorithme de masse, au sens où plus on a d'énoncés à vérifier, plus on rentabilise la construction de la p-destinée exhaustive et essentielle.

3. Enfin, on peut remarquer un autre phénomène, qui n'a pas été évoqué jusqu'à présent car on s'intéressait uniquement à une complexité dans le pire des cas. Il s'agit du rapport entre le nombre de nœuds de la destinée, et sa majoration grâce à la fonction H. Il peut arriver (et même, on peut conjecturer que cela arrive très souvent) que la destinée soit "creuse", c'est-à-dire qu'il y ait peu de nœuds dont l'étiquette est inférieure à la borne utilisée pour la construction qui soient finalement conservés dans la destinée essentielle.
   Par exemple, dans la 3-destinée réduite de l'ordre (voir Figure 7.6 du Chapitre 7), on s'aperçoit que tous les entiers entre 0 et le plus grand nœud ne sont pas présents à chaque rang. Par exemple, dans le sous-arbre de rang 1 de 3, le sous-arbre de rang 2 de 5 ne comporte pas les feuilles 1 et 2.
   Un exemple plus flagrant est donné dans la 3-destinée de la structure $\langle \mathbb{N}, S, \perp \rangle$, où les tests ont montré qu'il n'y a aucun sous-arbre de rang 1 dont la racine est comprise entre 33554431 et $2^{227} - 1$.
   Cette lacunarité des destinées intervient sur le nombre d'instanciations effectuées lors de la vérification d'un énoncé. Ce nombre d'instanciations n'apparaît pas dans le modèle des machines de Turing alternantes, mais dans la pratique il intervient de manière importante (voir la complexité en temps de l'algorithme de vérification dans le modèle de calcul classique).
   Dans l'algorithme de Ferrante et Rackoff, on ne connaît pas ces endroits où il y a un "trou" (c'est-à-dire aucune instanciation pouvant mener à une situation nouvelle), donc on parcourt l'ensemble des instanciations dans un pavé donné.
   En conclusion, plus la p-destinée est lacunaire, et plus l'algorithme de Nézondet est intéressant.

Conclusion et perspectives de la Partie II

Conclusion

Cette Partie II s'articule en deux axes.

- Premier axe : nous nous sommes fondé sur deux exemples de structures arithmétiques, $\langle \mathbb{N}, S, \perp \rangle$, dont la théorie est indécidable, et $\langle \mathbb{N}, \leq \rangle$, dont la théorie est décidable, pour récolter quelques observations au sujet de la construction des destinées dans les structures infinies. Nous avons observé que dans le premier cas, la 3-destinée réduite de la structure est "difficile à construire", car nous nous heurtons à des problèmes ouverts en théorie des nombres et en arithmétique (Chapitre 6). Cette "difficulté" n'est pas surprenante, car dans le cas d'une théorie indécidable, il n'y a pas d'algorithme de construction des destinées de la structure. Le deuxième exemple nous confirme le lien entre décidabilité de la théorie de la structure et construction automatique des destinées, car dans ce cas, nous sommes en mesure de décrire un algorithme de construction automatique de destinées exhaustives et essentielles de la structure pour toute hauteur (Chapitre 7). Nous avons même la possibilité de construire, pour tout n-uple d'éléments de $\mathbb{N}$, une p-destinée exhaustive et essentielle de ce n-uple. La question s'est alors naturellement posée de caractériser la classe des structures pour lesquelles il existe un algorithme de construction de destinées exhaustives et essentielles, que nous pouvons définir comme les structures "à destinées récursives".

> **Définition 58 (Structure à destinées récursives)**
> *Soit $\mathcal{A}$ une structure sur un langage relationnel fini σ. On dit que la structure $\mathcal{A}$ est* **à destinées récursives** *s'il existe un algorithme prenant en entrée un entier p et donnant en sortie une p-destinée exhaustive et essentielle de la structure $\mathcal{A}$.*

- Deuxième axe : dans le souci de caractériser la classe des structures à destinées récursives, nous nous sommes appuyé sur une notion déjà existante, celle des structures H-bornées. Nous avons démontré dans un premier temps que les structures H-bornées sont à destinées récursives, et que l'inclusion inverse est fausse (Chapitre 8). Puis nous avons comparé deux méthodes de décision dans les structures

H-bornées, celle proposée par Ferrante et Rackoff, et celle proposée par Nézondet (Chapitre 9).

Nous pouvons étendre les travaux présentés dans cette partie, d'une part en s'intéressant à ce qu'il se passe pour des structures "très infinies", c'est-à-dire les structures ordinales, et d'autre part en poursuivant la recherche d'une caractérisation de la classe des structures à destinées récursives. On s'intéresse notamment au lien entre cette classe et la classe des structures dont la théorie est décidable.

Généralisation à l'addition ordinale

Dans [Mau94], Maurin étend les notions définies par Ferrante et Rackoff ([FR79]) aux ordinaux, et démontre ainsi des résultats de complexité pour l'addition ordinale. Ces résultats nous offrent une possibilité de prolonger le travail effectué aux Chapitres 8 et 9. La définition de destinée d'une structure est très générale, et ne fait pas intervenir la cardinalité du domaine. Une p-destinée exhaustive et essentielle est un objet fini, même si le domaine est de cardinalité supérieure à ω. Il est donc possible de produire un "algorithme" de Nézondet généralisé aux structures ordinales (on calcule à l'aide d'automates à temps ordinal). La comparaison entre cet algorithme de Nézondet généralisé et l'algorithme de décision de Ferrante et Rackoff dans ces structures est analogue à celle effectuée au Chapitre 9.

Caractérisation de la récursivité des destinées

Nous avons constaté que la classe des structures H-bornées pour une fonction H récursive ne capture pas la classe des structures à destinées récursives. Comment, dès lors, caractériser cette dernière ? Une réponse partielle est fournie par Patrick Cegielski dans [Ceg01]. Il démontre en effet le résultat suivant :

Theorème 8
Soit $R_1, \ldots, R_k$ des symboles de relation, et $\langle \mathbb{N}, R_1, \ldots, R_k \rangle$ une structure telle que :
– *la théorie complète de la structure $\langle \mathbb{N}, R_1, \ldots, R_k, (i)_{i \in \mathbb{N}} \rangle$ est décidable (tout élément de $\mathbb{N}$ est considéré comme une constante) ;*
– *les relations R_i sont récursives.*
Alors on a un algorithme qui calcule une p-destinée exhaustive et essentielle de la structure pour tout p.

On a donc une réponse partielle à notre question de caractérisation. Le résultat de Cegielski porte sur des structures de domaine $\mathbb{N}$ ayant des propriétés assez contraignantes.

Définition 59 (Structure de Cegielski)
Soit $R_1, \ldots, R_k$ des symboles de relation. La structure $\langle \mathbb{N}, R_1, \ldots, R_k \rangle$ est une **structure de Cegielski** *si :*
- *la théorie complète de la structure $\langle \mathbb{N}, R_1, \ldots, R_k, (i)_{i \in \mathbb{N}} \rangle$ est décidable (tout élément de $\mathbb{N}$ est considéré comme une constante) ;*
- *les relations R_i sont récursives.*

On a donc une inclusion de la classe des structures de Cegielski dans la classe des structures à destinées récursives. Les conditions de la définition d'une structure de Cegielski sont plus fortes que la simple possibilité de construire une p-destinée exhaustive et essentielle de la structure. En fait , on montre que l'on peut alors construire des p-destinées exhaustives et essentielle de tout n-uple d'éléments de la structure, si c'est une structure de Cegielski. Cela nous amène à considérer une autre notion de structure à destinées récursives, celle de structure à destinées fortement récursives :

Définition 60 (Structure à destinées fortement récursives)
Soit $\mathcal{A}$ une structure sur un langage relationnel fini σ. La structure $\mathcal{A}$ est une **structure à destinées fortement récursives** *s'il existe un algorithme qui prend en entrée un entier p et un n-uple $(a_1, \ldots, a_n)$ d'éléments du domaine de $\mathcal{A}$ et qui produit une p-destinée exhaustive et essentielle du n-uple $(a_1, \ldots, a_n)$.*

Nous noterons :
- **DR** la classe des structures à destinées récursives,
- **DFR** la classe des structures à destinées fortement récursives,
- **CEG** la classe des structures de Cegielski,
- **HB** la classe des structures H-bornées telles que (mêmes restrictions qu'au Chapitre 8) :
  - il n'y a qu'un nombre fini d'éléments de la structure de norme inférieure à un entier donné ;
  - il est possible de les énumérer récursivement ;
  - la fonction H est récursive.

Nous connaissons déjà quelques inclusions entre ces classes, présentées dans le schéma de la Figure 9.2.

Nous pouvons pousser ces résultats un peu plus loin. Notamment, nous montrons que :
- $\mathbf{HB} = \mathbf{DFR}$;
- $\mathbf{CEG} \subsetneq \mathbf{HB} = \mathbf{DFR}$;
- $\mathbf{CEG} = \mathbf{DFR} \cap \{\text{Structures arithmétiques}\}$;
- $\mathbf{DFR} \subsetneq \mathbf{DR}$.

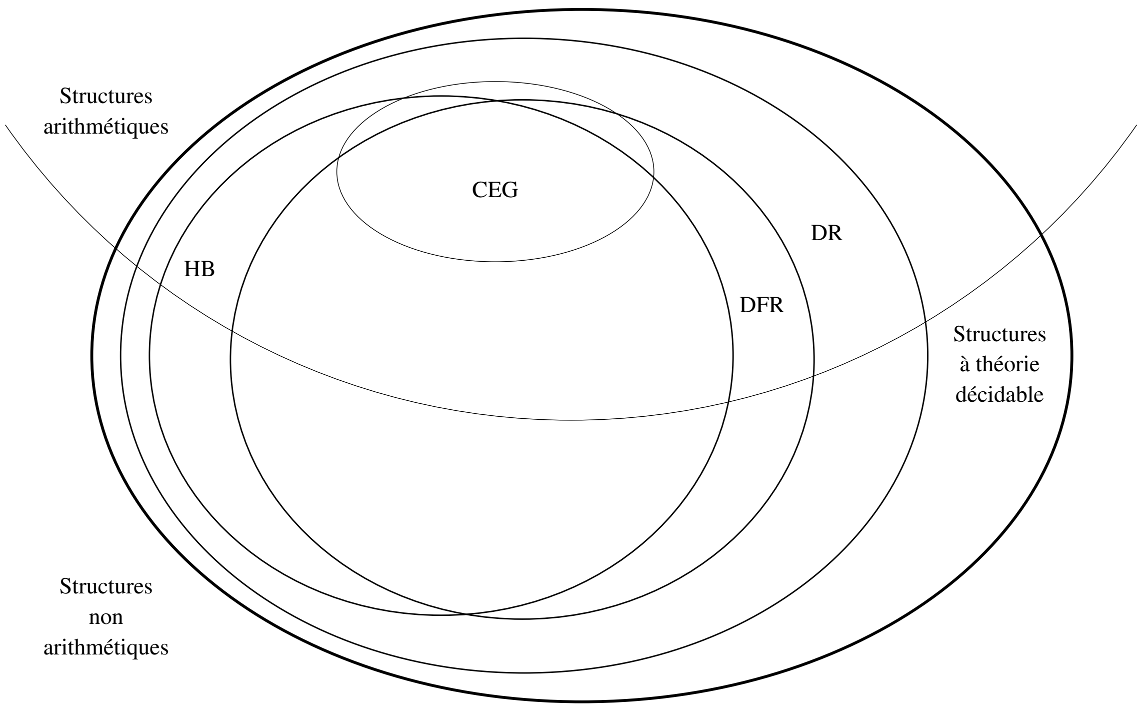


FIG. 9.2 – Inclusions entre classes de structures.

Proposition 12 (DFR $\subsetneq$ DR)
La classe des structures à destinées fortement récursives est strictement incluse dans la classe des structures à destinées récursives.

Preuve : L'inclusion est évidente. Le caractère strict est donné par le contre-exemple du Chapitre 8. En effet, dans cet exemple, il est impossible de construire une p-destinée du singleton (a) pour un a dans l'ensemble indécidable $\mathcal{M}$.

□

Proposition 13 (HB = DFR)
La classe des structures à destinées fortement récursives coïncide avec la classe des structures H-bornées avec H récursive (sous les conditions restrictives sur la norme).

Preuve : Soit $\mathcal{A}$ une structure H-bornée, avec les restrictions ci-dessus. Soit $(a_1, \ldots, a_p)$ un p-uple d'éléments du domaine de $\mathcal{A}$, et $m_0 = \max(||a_1||, \ldots, ||a_p||)$. La structure $(\mathcal{A}, a_1, \ldots, a_p)$ est $H_{m_0,p}$-bornée pour la fonction :

$$H_{m_0,p}(n, k, m) = H(n, k + p, \max(m, m_0)).$$

L'algorithme de construction des destinées des p-uples se présente alors sous la forme :

Entrée : un p-uple $(a_1, \ldots, a_p)$, une profondeur l
Sortie : une l-destinée exhaustive et essentielle du p-uple $(a_1, \ldots, a_p)$

1. Calculer $m_0 = \max(||a_1||, \ldots, ||a_p||)$
2. Construire l'arbre complet de hauteur l en utilisant comme borne sur les fils d'un nœud la fonction $H_{m_0,p}(n, k, m)$
3. Ajouter les listes de relations
4. Normaliser
5. Éliminer les redondances

Soit $\mathcal{A}$ une structure à destinées fortement récursives. On fixe m, k et n des entiers. On note :

$$U_{k,m} = \{(a_1, \ldots, a_k) \in A^k / \text{ Pour tout } i \in \{1, \ldots, k\},\ ||a_i|| \leq m\}.$$

Cet ensemble est fini d'après les restrictions effectuées sur la norme.
On appelle $D_n(a_1, \ldots, a_k)$ une n-destinée réduite du k-uple $(a_1, \ldots, a_k)$. On pose la fonction H :

$$H(n, k, m) = \max_{U_{k,m}}(Sup_{f_{D_n(a_1,\ldots,a_k)}}(\emptyset)).$$

Cette fonction H est récursive, puisque l'on peut calculer récursivement des n-destinées de tout k-uple. La structure $\mathcal{A}$ est H-bornée : soit $(a_1, \ldots, a_k)$ un élément de $U_{k,m}$, et $F(x_1, \ldots, x_k)$ une formule de profondeur de quantification n. On suppose que :

$$\mathcal{A} \models \exists x_{k+1} F(a_1, \ldots, a_k, x_{k+1}).$$

Alors :

$$D_n(a_1, \ldots, a_k) \models \exists x_{k+1}(P(\emptyset, x_{k+1}) \wedge \tilde{F}(a_1, \ldots, a_k, x_{k+1})),$$

et d'après la définition de Sup_f, on a un nœud α fils de $\emptyset$ dans cette destinée tel que $||l(\alpha)|| \leq Sup_f(\emptyset) \leq H(n, k, m)$, et :

$$D_n(a_1, \ldots, a_k) \models \tilde{F}(a_1, \ldots, a_k, \alpha),$$

ce qui implique :

$$\mathcal{A} \models F(a_1, \ldots, a_k, l(\alpha)).$$

Par conséquent, on a un élément de $\mathcal{A}$, qui est $a_{k+1} = l(\alpha)$, tel que $||a_{k+1}|| \leq H(n, k, m)$ et :

$$\mathcal{A} \models F(a_1, \ldots, a_k, a_{k+1}).$$

Ceci démontre que la structure $\mathcal{A}$ est H-bornée, pour une fonction H récursive.

□

Proposition 14 (CEG $\subsetneq$ HB = DFR)
La classe des structures de Cegielski est strictement incluse dans la classe des structures à destinées fortement récursives.

Preuve : On montre que les structures de Cegielski sont à destinées fortement récursives, en adaptant la preuve de Cegielski dans [Ceg01]. En effet, si la structure $\langle \mathbb{N}, R_1, \ldots, R_k \rangle$ est une structure de Cegielski, la structure $\langle \mathbb{N}, R_1, \ldots, R_k, a_1, \ldots, a_n \rangle$ l'est également, et on applique le même processus que dans [Ceg01] pour construire les p-destinées de cette structure.

Cette inclusion est stricte car on peut trouver des structures H-bornées dont le domaine n'est pas $\mathbb{N}$, par exemple la structure 2SEL (2 Successors, Equal Length) présentée dans [FR79] (pages 102-111). Cette structure a pour domaine l'ensemble des mots sur l'alphabet $\{0, 1\}$, avec deux successeurs r_0, r_1, et la relation “avoir la même longueur”.

$\square$

Proposition 15 (CEG = DFR$\cap${Structures arithmétiques})
La classe des structures de Cegielski est la classe des structures de domaine $\mathbb{N}$ à destinées fortement récursives.

Preuve : On a déjà une inclusion (toute structure de Cegielski est une structure de domaine $\mathbb{N}$ à destinées fortement récursives), il nous reste à montrer l'inclusion inverse. Soit $\langle \mathbb{N}, R_1, \ldots, R_k \rangle$ une structure de domaine $\mathbb{N}$ à destinées fortement récursives. Alors, nécessairement, chaque relation R_i est récursive, et on peut lire la satisfaction d'un énoncé de profondeur de quantification p où interviennent des constantes $a_1, \ldots, a_n$ sur une p-destinée exhaustive et essentielle du n-uple $(a_1, \ldots, a_n)$.

$\square$

Le schéma d'inclusion résultant des propositions précédentes entre les classes de structures est celui de la Figure 9.3 ($F(\mathbb{R}^2)$ est l'ensemble des fermés de $\mathbb{R}^2$ dans la topologie usuelle). Il nous reste dès lors deux indéterminations : existe-t-il des structures décidables qui ne sont pas à destinées récursives, respectivement arithmétiques et non arithmétiques ?

On peut également remarquer que nous disposons maintenant d'une caractérisation en termes de destinées des structures H-bornées avec H récursive (et restrictions sur la norme).

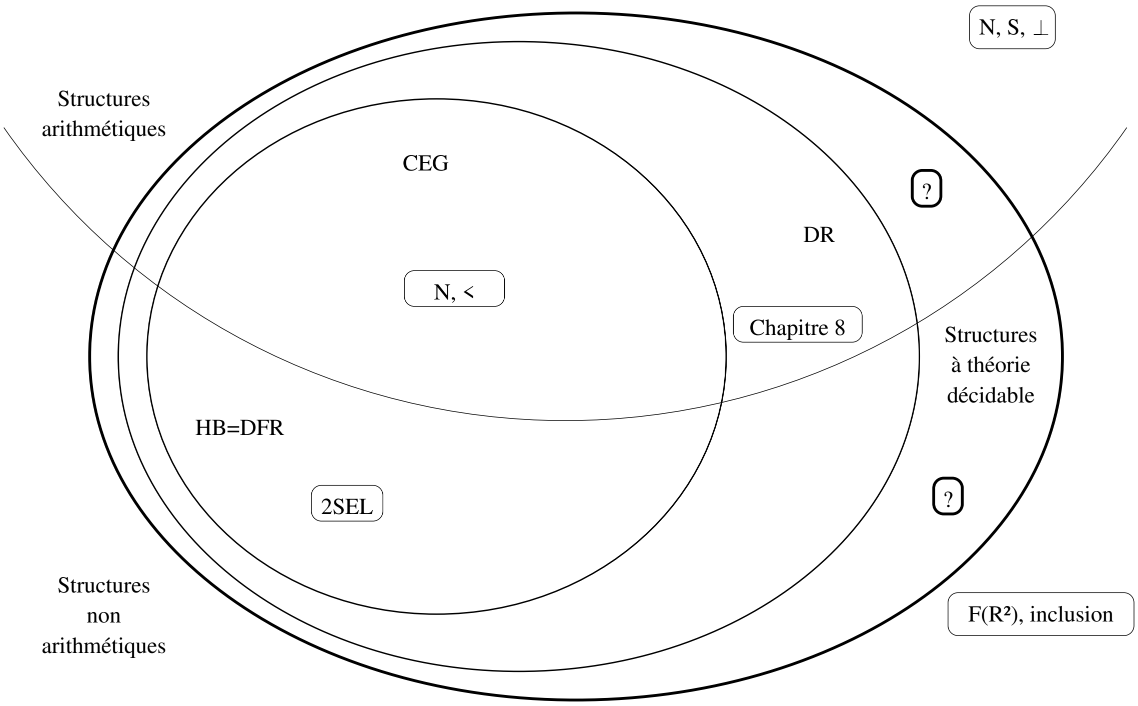


FIG. 9.3 – Inclusions entre classes de structures.

Troisième partie

Structures finies

Introduction

Cette troisième partie est née d'une suggestion d'Etienne Grandjean[1], qui à la lecture de quelques travaux sur les destinées, a vu un lien possible avec une conjecture en complexité, la conjecture NE $\stackrel{?}{=}$ CoNE. Le destin de cette conjecture est lié à celui de la célèbre conjecture P $\stackrel{?}{=}$ NP, car si NE$\neq$CoNE, alors P$\neq$NP. Dans cette partie, nous considérons l'équivalent en théorie des modèles finis de la conjecture NE $\stackrel{?}{=}$ CoNE, appelé "conjecture du Spectre". Cette conjecture est énoncée par Asser dans [Ass55]. Le spectre d'un énoncé est l'ensemble des cardinaux des modèles finis de l'énoncé, et la conjecture consiste en la question "Le complémentaire d'un spectre est-il un spectre ?". Nous donnons quelques pistes pour attaquer la conjecture du Spectre, puis exposons quelques cas particuliers où cette conjecture est résolue.

Le Chapitre 10 présente le contexte général du problème NE $\stackrel{?}{=}$ CoNE, ses liens avec le problème NP $\stackrel{?}{=}$ CoNP, ainsi que la conjecture du Spectre. Ce chapitre énonce également une conjecture, proposée par Ash dans [Ash94], qui implique la conjecture du Spectre. Nous en présentons une nouvelle version, affaiblie, qui est exactement équivalente à la conjecture du Spectre.

Dans le Chapitre 11, nous exposons quelques idées générales pour attaquer la conjecture de Ash et la conjecture du Spectre, à l'aide de restrictions syntaxiques et sémantiques sur les énoncés considérés. Nous présentons également quelques pistes susceptibles de faire avancer la résolution de la conjecture du Spectre. Nous distinguons trois grandes orientations, qui font l'objet des trois derniers chapitres.

La première piste est exploitée au Chapitre 12. Elle consiste à étudier précisément ce qu'il se passe pour les énoncés de profondeur de quantification 2. On montre dans ce cas que la conjecture de Ash est vérifiée. Cela implique que le complémentaire d'un spectre d'énoncé de profondeur de quantification 2 est un spectre. En particulier, il s'agit d'un spectre d'énoncé de profondeur de quantification 2.

La deuxième piste consiste à s'intéresser à différentes restrictions sémantiques, pour un langage constitué d'une relation binaire. C'est l'objet du Chapitre 13. Fagin, dans [Fag75], montre que le cas général découle de la conjecture du Spectre pour le cas d'une relation de graphe. Dans ce chapitre, on part d'un exemple où les modèles sont simples à décrire, à savoir un graphe de bijection. Dans ce cas, on montre que la conjecture du Spectre est vérifiée, mais pas la conjecture de Ash (seulement une de ses versions affaiblies). On élargit ensuite les possibilités du langage afin de se rapprocher du cas d'une relation de graphe. On montre ainsi que la conjecture du Spectre est vérifiée pour un graphe de fonction,

[1] GREYC, Caen

éventuellement avec des relations unaires, pour un graphe de degré total 2, orienté ou non orienté, avec ou sans relations unaires.

La troisième piste vise également à se rapprocher du cas d'une relation de graphe, mais par l'intermédiaire d'un codage : il est possible de coder une relation de graphe à l'aide de deux relations d'équivalence et l'égalité. Nous montrons que la conjecture du spectre est vérifiée pour une relation d'équivalence et l'égalité, pour deux relations d'équivalence imbriquées et l'égalité, puis nous décrivons le codage d'un graphe en utilisant deux relations d'équivalence (non imbriquées) et l'égalité. C'est l'objet du Chapitre 14.

Chapitre 10

Le problème NE $\stackrel{?}{=}$ CoNE et la conjecture du Spectre

La théorie des modèles finis s'intéresse aux classes de complexité, et notamment à leurs liens avec la logique. Ce chapitre présente un problème de complexité, le problème NE $\stackrel{?}{=}$ CoNE, et une approche fondée sur les outils de la théorie des modèles finis. Dans un premier temps, nous rappelons le contexte de ce problème, puis nous en présentons au Paragraphe 10.2 la version modèles finis : la *conjecture du Spectre*. Nous nous ramenons ensuite, au Paragraphe 10.3, à un problème équivalent où intervient le nombre de classes de k-isomorphisme (au sens de Fraïssé) de structures d'un cardinal donné. Ce problème, appelé *conjecture de Ash "ultrafaible"*, est la généralisation d'une conjecture énoncée par Ash en 1994.

10.1 Contexte du problème NE $\stackrel{?}{=}$ CoNE

10.1.1 Quelques définitions, et un récapitulatif

Nous ne rappelons pas ici ce qu'est une machine de Turing, on pourra pour cela se référer à [Pap94], [VL90] ou [BDG90]. Nous rappelons en revanche la définition de quelques classes de complexité pour ce modèle de calcul.

Définition 61 (Classe de complexité P)
La classe P est la classe des langages reconnaissables par une machine de Turing déterministe en temps polynomial.

Définition 62 (Classe de complexité NP)
La classe NP est la classe des langages reconnaissables par une machine de Turing non déterministe en temps polynomial.

Définition 63 (Classe de complexité CoNP)
La classe CoNP est la classe des langages dont le complémentaire est dans NP.

Définition 64 (Classe de complexité EXP)
La classe EXP est la classe des langages reconnaissables par une machine de Turing déterministe en temps $\mathcal{O}(2^{cn^k})$, où c, k sont des constantes et n la taille de l'entrée.

Définition 65 (Classe de complexité NEXP)
La classe NEXP est la classe des langages reconnaissables par une machine de Turing non déterministe en temps $\mathcal{O}(2^{cn^k})$, où c, k sont des constantes et n la taille de l'entrée.

Définition 66 (Classe de complexité CoNEXP)
La classe CoNEXP est la classe des langages dont le complémentaire est dans NEXP.

Définition 67 (Classe de complexité E)
La classe E est la classe des langages reconnaissables par une machine de Turing déterministe en temps $\mathcal{O}(2^{cn})$, où c est une constante et n la taille de l'entrée.

Définition 68 (Classe de complexité NE)
La classe NE est la classe des langages reconnaissables par une machine de Turing non déterministe en temps $\mathcal{O}(2^{cn})$, où c est une constante et n la taille de l'entrée.

Définition 69 (Classe de complexité CoNE)
La classe CoNE est la classe des langages dont le complémentaire est dans NE.

La Figure 10.1 présente quelques relations d'inclusion entre différentes classes de complexité. Parmi ces inclusions, on peut remarquer que certaines d'entre elles sont strictes :

- NEXP $\subsetneq$ EXPSPACE ([VL90]) ;
- PSPACE $\subsetneq$ EXP ([HO02]).

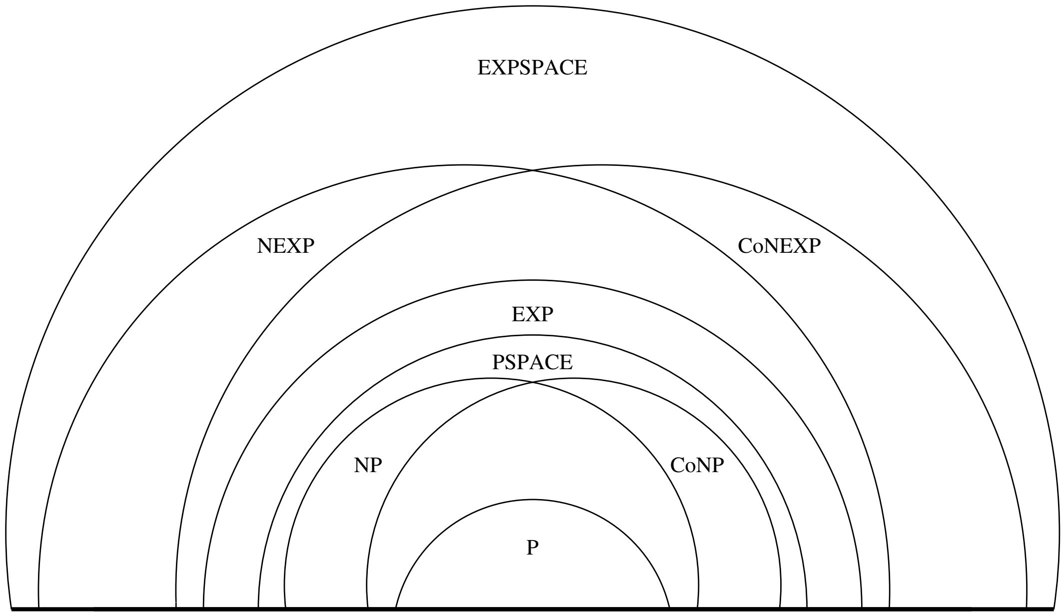


Fig. 10.1 – Inclusions entre classes de complexité.

Le problème P $\stackrel{?}{=}$ NP est l'un des problèmes les plus connus de l'informatique théorique. Il s'agit, en très résumé, de savoir si nous devons brûler nos cartes de crédits. En conséquence, ce problème a été et est très étudié. Un des moyens d'attaque de cette question est de s'intéresser à la clôture par complément des classes de complexité NE et NP. Ce qu'on sait de la clôture par complément des classes de complexité peut se résumer ainsi :

- les classes déterministes, en temps ou en espace, sont closes par complément ([Pap94]) ;
- les classes non déterministes en espace sont closes par complément à partir de NSPACE($\log^2(n)$) ([Pap94], p.142) ;
- la classe de complexité NL (=NSPACE($\log(n)$)) est close par complément (Théorème d'Immerman-Szelepcsényi, voir [Imm88] et [Sze87]) ;
- on ne dispose pas de théorème général concernant les classes non déterministes en temps. En particulier pour NE et NP la question reste ouverte.

Or, nous avons le schéma d'implications suivant ([VL90]) :

$$\begin{array}{ccccccc} P & = & NP & \Rightarrow & NP & = & CoNP \\ & \Downarrow & & & & & \\ E & = & NE & \Rightarrow & NE & = & CoNE \end{array}$$

Une réponse négative à l'une des questions NP $\stackrel{?}{=}$ CoNP et NE $\stackrel{?}{=}$ CoNE implique donc que P $\neq$ NP. En revanche, une réponse positive à ces deux problèmes ne nous apprend rien de probant sur P $\stackrel{?}{=}$ NP.

10.1.2 Approches du problème NP $\stackrel{?}{=}$ CoNP

De nombreux travaux ont été effectués autour de ce problème, notamment en terme de complexité descriptive. Fagin, dans [Fag90], donne un résumé très clair de ces différents travaux, et du cheminement des idées depuis l'introduction de la notion de spectre par Scholz en 1952 ([Sch52]). Nous reviendrons sur ces travaux au Paragraphe 10.2.

On peut également relativiser les classes de complexité mises en jeu par l'emploi d'un oracle.

> **Définition 70 (Machine de Turing avec oracle)**
> *Soit A un langage. Une* **machine de Turing avec oracle** M^A *est une machine de Turing disposant d'un ruban supplémentaire appelé* **ruban de requête**, *et de trois états supplémentaires $q_?$, q_y, et q_n. Quand la machine entre dans l'état $q_?$, on effectue une requête à l'oracle : si le contenu du ruban de requête est dans l'ensemble A accepté par l'oracle, on passe à l'état q_y ; sinon on passe à l'état q_n.*

Un oracle est donc un moyen d'avoir une réponse instantanée à une ensemble de questions défini à l'avance. Ce modèle de calcul permet de savoir ce qu'il se passe dans un “univers différent”, afin de se donner des intuitions sur ce qu'il se passe dans le modèle de calcul classique. Malheureusement, nous avons des résultats discordants concernant les problèmes P $\stackrel{?}{=}$ NP et NP $\stackrel{?}{=}$ CoNP :

- il existe un oracle A pour lequel $\mathrm{P}^A = \mathrm{NP}^A$, et il existe un oracle B pour lequel $\mathrm{P}^B \neq \mathrm{NP}^B$ ([Pap94]) ;
- il existe des oracles C, D, E pour lesquels $\mathrm{NP}^C = \mathrm{CoNP}^C$ et $\mathrm{P}^C \neq \mathrm{NP}^C$, $\mathrm{NP}^D \neq \mathrm{CoNP}^D$ et $\mathrm{P}^D = \mathrm{NP}^D \cap \mathrm{CoNP}^D$, $\mathrm{NP}^E \neq \mathrm{CoNP}^E$ et $\mathrm{P}^E \neq \mathrm{NP}^E \cap \mathrm{CoNP}^E$ ([BGS75]).

Ces résultats nous confirment qu'une preuve de NP = CoNP ou de NP $\neq$ CoNP est certainement “difficile”, au sens où l'on doit s'affranchir du modèle de calcul. Cela nous incite à penser que la voie de la complexité descriptive est peut-être mieux adaptée, puisque, justement on se place dans un cadre logique par nature indépendant du modèle de calcul.

10.1.3 Approches du problème NE $\stackrel{?}{=}$ CoNE

L'approche utilisant les oracles est, comme dans le cas précédent, vouée à l'échec ([HIS85]) : on trouve en effet un oracle A tel que $\mathrm{NP}^A = \mathrm{CoNP}^A$ mais $\mathrm{NE}^A \neq \mathrm{CoNE}^A$, alors que l'on a l'implication NP = CoNP $\Rightarrow$ NE = CoNE.

Nous pouvons cependant espérer que l'approche “complexité descriptive” donne quelques résultats. C'est l'objet de cette partie, dans laquelle nous présentons des résultats compatibles avec la clôture par complément de la classe NE.

10.2 Version modèles finis : la conjecture du Spectre

10.2.1 Définition d'un spectre

En 1952, Scholz introduit la notion de spectre d'une formule ([Sch52]), et s'interroge sur une caractérisation possible de la classe des spectres.

Définition 71 (Spectre)
On appelle **spectre** *d'un énoncé du premier ordre Φ sur un langage relationnel fini, et on note $Sp(\Phi)$ l'ensemble des cardinaux des modèles finis de l'énoncé Φ.*

Exemples :

1. $\sigma = \{=\}$, $\varphi := \exists x \exists y[\neg(x = y) \wedge \forall z((z = x) \vee (z = y))]$. Tous les modèles de φ sont nécessairement de taille 2, donc $Sp(\varphi) = \{2\}$. Le spectre de l'énoncé φ est représenté Figure 10.2.

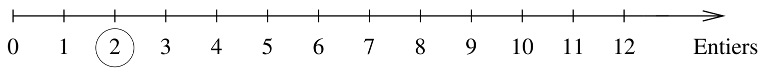


FIG. 10.2 – Spectre de $\varphi := \exists x \exists y[\neg(x = y) \wedge \forall z((z = x) \vee (z = y))]$.

2. $\sigma = \{<\}$, $\varphi := \exists x \exists y \exists z[(x < y) \wedge (y < z)]$. Tous les modèles de φ ont au moins 3 éléments, et on a un modèle de φ de cardinal n pour tout $n \geq 3$. Le spectre de l'énoncé φ est représenté Figure 10.3.

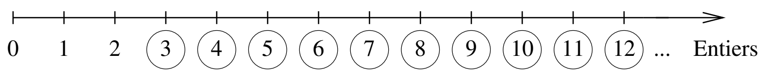


FIG. 10.3 – Spectre de $\varphi := \exists x \exists y \exists z[(x < y) \wedge (y < z)]$.

3. $\sigma = \{+, \times, 0, 1\}$, $\varphi :=$ Axiomes des corps finis. Les cardinaux des modèles de φ sont tous les entiers de la forme p^α avec p un nombre premier et $\alpha \geq 1$. Le spectre de l'énoncé φ est représenté Figure 10.4.

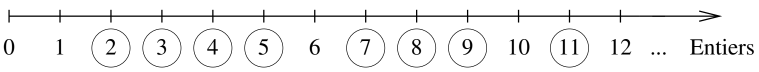


FIG. 10.4 – Spectre de la théorie des corps finis.

10.2.2 Énoncé de la conjecture du Spectre

La conjecture du Spectre est née de la question d'Asser, en 1955 ([Ass55], problème 3 page 263) : "Est-ce que le complémentaire d'un spectre est un spectre ?". La conjecture s'énonce alors :

Conjecture 1 (Conjecture du Spectre)
Soit Φ un énoncé du premier ordre sur un langage relationnel fini σ, et $Sp(\Phi)$ son spectre. Il existe un langage τ et un énoncé Ψ du premier ordre sur ce langage telle que :

$$Sp(\Psi) = \mathbb{N} \backslash Sp(\Phi).$$

10.2.3 Lien entre les spectres et NE

Suite à la question de Scholz ([Sch52]) sur une possible caractérisation des spectres, Asser a montré que les spectres sont des ensembles récursifs, mais ne couvrent pas tous les ensembles récursifs ([Ass55]). Il pose également la question du complémentaire (*i.e.* la conjecture du Spectre). Les travaux de Mostowski et d'Asser permettent de donner quelques exemples d'ensembles d'entiers qui sont des spectres, comme l'ensemble des nombres premiers, l'ensemble des multiples d'un entier k donné ou bien encore l'ensemble image de la fonction factorielle ([Mos56]). Quelques années plus tard, Bennett ([Ben62]) poursuit ces travaux, notamment en situant la classe des spectres dans la hiérarchie de Grzegorczyk ([Grz53]). L'équivalence entre les spectres de formule et la classe de complexité NE a été établie par Jones et Selman en 1974 dans [JS74], suite à tous les travaux antérieurs. Ils définissent des *automates de spectre*, dont le calcul est ramené à celui d'une machine de Turing non déterministe en temps exponentiel. Leur résultat est le suivant :

Theorème 9 (Équivalence entre les spectres et la classe NE)
Soit S une partie de $\mathbb{N}$. Les assertions suivantes sont équivalentes :
- *S est le spectre d'un énoncé du premier ordre avec égalité ;*
- *S est accepté par un automate de spectre (en écrivant les entiers en binaire) ;*
- *S est accepté par une machine de Turing non déterministe, en temps $\mathcal{O}(2^{cn})$, où c est une constante et n la taille de l'entrée (en écrivant les entiers en binaire).*

Une conséquence de cette caractérisation est la suivante : le problème NE $\stackrel{?}{=}$ CoNE est équivalent à la résolution de la conjecture du Spectre.

10.2.4 Parallèle avec le problème NP $\stackrel{?}{=}$ CoNP

On peut comparer les travaux effectués sur le problème NE $\stackrel{?}{=}$ CoNE à ceux effectués sur le problème P $\stackrel{?}{=}$ NP, notamment ceux de Fagin. Celui-ci définit une notion de **spectre généralisé**, inspirée de la notion de spectre, mais qui diffère sur deux points :

- tout d'abord, les formules considérées sont des formules Σ_1^1, autrement dit des formules de la forme :

  $$\exists Q_1 \ldots \exists Q_k F(P_1, \ldots, P_s, Q_1, \ldots, Q_k),$$

  où F un énoncé du premier ordre, les P_i sont des variables de prédicats appelés **prédéfinis** et les Q_j sont des variables de prédicats dits **extra-prédicats**. On a donc quitté le premier ordre ;
- de plus, les ensembles appelés spectres généralisés sont les ensembles de structures finies vérifiant une propriété Σ_1^1, et non l'ensemble de leurs cardinalités.

Fagin démontre dans [Fag74] un résultat analogue à celui de Jones et Selman, mais pour les spectres généralisés (Théorème 10). Il définit un encodage Enc, qui transforme les structures finies en mots sur un alphabet fini, permettant de faire le lien entre les spectres généralisés et la classe NP.

Theorème 10 (Équivalence entre les spectres généralisés et NP)
Soit σ un langage non vide, et $\mathcal{Q}$ un ensemble de σ-structures clos par isomorphisme. Alors $\mathcal{Q}$ est un spectre généralisé si et seulement si son encodage $Enc(\mathcal{Q})$ est dans NP.

Dans le cadre des spectres généralisés, on capture donc la classe de complexité NP. Cela peut paraître à première vue paradoxal qu'avec des ensembles plus compliqués que les spectres (*i.e.* un ensemble de structures finies au lieu d'un ensemble de nombres), on capture une classe plus petite que NE. Ce paradoxe disparaît quand on se rappelle que le temps de calcul est estimé en fonction de la taille de l'entrée. Dans le cas des spectres, la taille de l'entrée correspond à la taille d'un entier (donc si m est l'entrée, la taille de l'entrée est de l'ordre de $\log(m)$), alors que dans le cas des spectres généralisés, l'entrée est le codage d'une structure finie (donc si m est le cardinal du domaine de la structure, le codage de la structure est intuitivement de taille m^k, où k dépend de l'arité des relations). Comme dans le deuxième cas, la taille de l'entrée est beaucoup plus grosse que dans le premier cas, le temps de calcul – par rapport à la taille de l'entrée – s'en trouve diminué d'autant, relativement au temps de calcul dans le cas des spectres.

L'approche modèles finis du problème NP $\stackrel{?}{=}$ CoNP n'a pas encore abouti, mais on connaît des résultats intermédiaires. Par exemple, Ajtai, dans [Ajt83], montre que si l'on se restreint au second ordre monadique, on n'a pas la clôture par complément :

Theorème 11 (MonNP $\neq$ MonCoNP)
Il existe une propriété exprimable dans MonCoNP, mais pas dans MonNP, même en présence de prédicats prédéfinis arbitraires.

On peut trouver dans [Sch96] une preuve détaillée de ce résultat.

10.3 La conjecture de Ash

En 1994, Ash propose dans [Ash94] une conjecture qui implique la conjecture du Spectre. Nous présentons ici différentes versions de cette conjecture, et notamment une version, établie par nos soins, qui non seulement implique la conjecture du Spectre, mais est aussi une condition nécessaire à celle-ci.

10.3.1 Notations

Il nous faut tout d'abord préciser le cadre : on notera
- σ un langage relationnel fini ;
- $k \geq 0$ une profondeur de quantification ;
- n le cardinal du domaine d'une σ-structure.

Le nombre de classes de k-isomorphisme de σ-structures finies est fini ([Poi85]) :

Définition 72 (Nombre de classes de k-isomorphisme)
Soit σ un langage relationnel fini, et $k \leq 0$. Le nombre de classes de k-isomorphisme de σ-structures finies est noté $M_{\sigma,k}$.

On définit alors une fonction de comptage des classes de k-isomorphisme de structures de cardinal donné, que l'on appelle *fonction de Ash* :

Définition 73 (Fonction de Ash $N_{\sigma,k}$)
La fonction $N_{\sigma,k}$ est la fonction qui, à un cardinal donné n, associe le nombre de classes de k-isomorphisme de σ-structures qui ont un représentant de cardinal n. Cette fonction est appelée la **fonction de Ash** *du langage σ au rang k.*

Remarque 21 *La fonction définie ci-dessus est bornée par $M_{\sigma,k}$.*

10.3.2 La conjecture de Ash

La conjecture de Ash s'énonce de la façon suivante :

> **Conjecture 2 (Conjecture de Ash)**
> *Pour tout langage σ relationnel fini égalitaire, et tout $k \geq 1$, la fonction de Ash, $N_{\sigma,k}$, est ultimement constante.*

Cette conjecture suppose donc un comportement asymptotique très lisse des structures finies vis-à-vis de la k-équivalence. L'intuition sous-jacente est la suivante : quand les structures deviennent très grandes, on ne peut pas représenter de nouvelles propriétés à profondeur de quantification k fixée. La conjecture de Ash implique la conjecture du Spectre, la preuve est donnée dans [Ash94]. Nous ne donnons pas la preuve ici, car elle ressemble beaucoup à celle que nous donnons du Théorème 12.

10.3.3 La conjecture de Ash périodique

Une version affaiblie de la conjecture de Ash est la conjecture de Ash périodique, qui suppose un comportement un peu moins lisse des structures finies :

> **Conjecture 3 (Conjecture de Ash périodique)**
> *Pour tout langage σ relationnel fini égalitaire, et tout $k \geq 1$, la fonction de Ash, $N_{\sigma,k}$, est ultimement périodique.*

La version affaiblie implique également la conjecture du Spectre, et la preuve est directement adaptée de celle de Ash. Cependant elle n'est pas une condition nécessaire de la conjecture du Spectre.

10.3.4 La conjecture de Ash "ultrafaible"

Nous avons établi au cours de la thèse une nouvelle conjecture, directement inspirée des hypothèses de Ash et de l'observation de l'équivalence entre la classe NE et celle des spectres. Elle fait intervenir les ensembles antécédents d'un entier par la fonction $N_{\sigma,k}$.

> **Conjecture 4 (Conjecture de Ash ultrafaible)**
> *Soit un langage σ et $k \geq 1$. On note pour tout $i \in \mathbb{N}$:*
>
> $$F_i = \{n \in \mathbb{N} / N_{\sigma,k}(n) = i\}.$$
>
> *Pour tout $i \in \{0, \ldots, M_{\sigma,k}\}$, l'ensemble F_i est dans NE.*

Remarque 22 *Les F_i forment une partition de $\mathbb{N}$, et ils sont vides pour $i > M_{\sigma,k}$.*

10.3.5 Équivalence entre la conjecture de Ash ultrafaible et la conjecture du Spectre

Il nous reste à démontrer à présent l'équivalence entre la conjecture de Ash ultrafaible et la conjecture du Spectre, c'est-à-dire montrer le Théorème 12.

Theorème 12
La conjecture du Spectre est vraie si et seulement si la conjecture de Ash ultrafaible est vraie.

Preuve :

Conjecture de Ash ultrafaible $\Rightarrow$ Conjecture du Spectre

La preuve est grandement inspirée de la preuve de Ash dans [Ash94]. On se donne un langage σ, et on fixe φ une σ-formule de profondeur de quantification k. On va montrer que $\mathbb{N} \backslash Sp(\varphi)$ est un spectre.

On considère les classes $(\mathcal{C}_i)_{i \in \{1,\ldots,M_{\sigma,k}\}}$ de k-isomorphisme de σ-structures finies. Pour chacune de ces classes, il existe un énoncé Ψ_i qui le caractérise (c'est-à-dire qu'une structure finie est dans $\mathcal{C}_i$ si et seulement si elle satisfait Ψ_i). Ce fait est justifié par le Théorème 3 du Chapitre 5.

Pour $j \in \{0, \ldots, M_{\sigma,k}\}$, on pose :

$$X_j = \{S \text{ ensemble de classes de } k\text{-isomorphisme } / \ |S| = j \text{ et } \forall \mathcal{C} \in S, \mathcal{C} \models \neg\varphi\}.$$

Autrement dit, parmi les classes de k-isomorphisme de σ-structures, on s'intéresse à celles qui ne satisfont pas φ (comme φ est de profondeur de quantification k, les structures dans une même classe de k-isomorphisme satisfont φ en même temps, ou satisfont $\neg\varphi$ en même temps, et on peut donc parler de la satisfaction de φ ou $\neg\varphi$ par toute une classe de k-isomorphisme). Parmi les classes ne satisfaisant pas la formule φ, on considère les ensembles constitués d'un nombre j de ces classes (Figure 10.5).

Pour tout $i \in \{1, \ldots, M_{\sigma,k}\}$, on appelle Ψ'_i l'énoncé construit à partir de Ψ_i en y remplaçant chaque symbole de σ par un symbole correspondant de σ_i, avec σ_i un langage dupliqué de σ tel que, pour tout $i \in \{1, \ldots, M_{\sigma,k}\}$ et tout $j \in \{1, \ldots, M_{\sigma,k}\}$, on ait $i \neq j \Rightarrow \sigma_i \cap \sigma_j \subseteq \{=\}$.

On pose à présent, pour tout $j \in \{1, \ldots, M_{\sigma,k}\}$:

$$\theta_j = \bigvee_{S \in X_j} \bigwedge_{\mathcal{C}_i \in S} \Psi'_i.$$

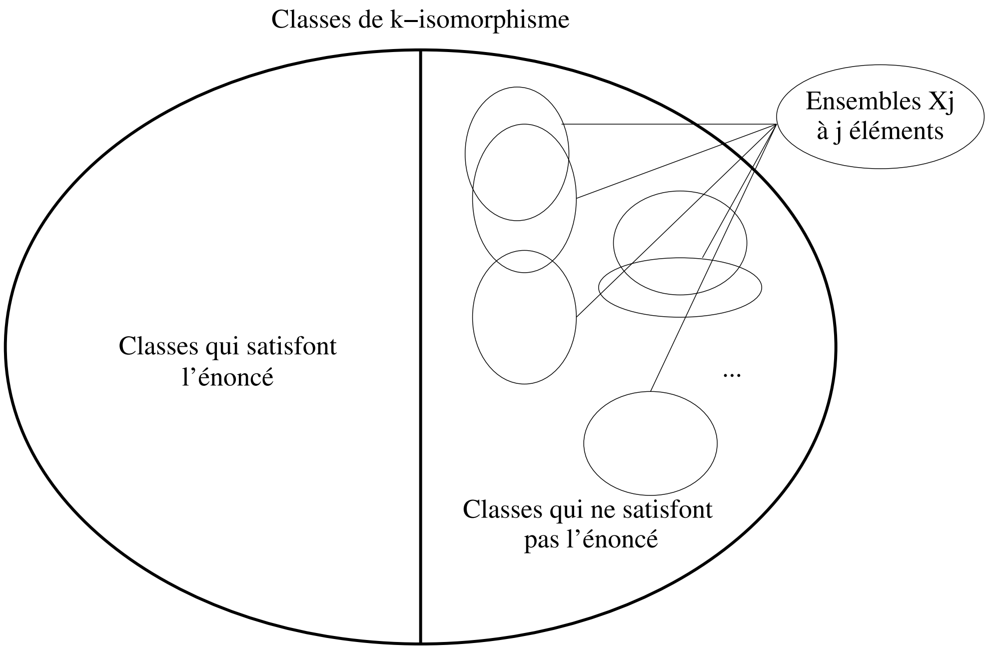


FIG. 10.5 – Ensembles X_j.

Intuitivement, cet énoncé θ_j signifie qu'on peut trouver un ensemble de j classes de k-isomorphisme de structures qui ne satisfont pas φ. Plus précisément, il signifie qu'un modèle de θ_j peut être "vu" de j façons différentes (selon le langage σ'_i que l'on considère), qui impliquent toutes la non-satisfaction de φ. Autrement dit, l'existence d'un modèle de θ_j de cardinal n implique l'existence de représentants de cardinal n pour au moins j des classes $\mathcal{C}_i$ qui satisfont $\neg\varphi$.

Le lemme suivant nous donne une caractérisation du complémentaire du spectre de φ grâce à ces formules θ_j et aux ensembles d'antécédents F_j de la fonction $N_{\sigma,k}$.

Lemme 6

$$\mathbb{N}\backslash Sp(\varphi) = \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap F_j).$$

Preuve : (du lemme)

$\boxed{\subseteq :}$ Soit $n \notin Sp(\varphi)$. Soit $j_0 \in \{1, \ldots, M_{\sigma,k}\}$ tel que $n \in F_{j_0}$ (ce j_0 est unique). On a alors un ensemble S_0 de j_0 classes de k-isomorphisme $\mathcal{C}_i$ qui ont chacune au moins un représentant de taille n, par définition de F_{j_0}.
Puisque $n \notin Sp(\varphi)$, on ne peut pas avoir, pour aucun des $\mathcal{C}_i$ de l'ensemble S_0, $\mathcal{C}_i \models \varphi$. Donc pour tout $\mathcal{C}_i$ de l'ensemble S_0, on a $\mathcal{C}_i \models \neg\varphi$.
Autrement dit, si M est une structure de taille n, telle que, pour tout i tel que $\mathcal{C}_i \in S_0$,

M est un modèle de $\mathcal{C}_i$ pour une interprétation du langage σ'_i, alors M est un modèle de taille n de θ_{j_0}. Or un tel modèle existe, puisque toutes les classes $\mathcal{C}_i \in S_{j_0}$ ont au moins un représentant de taille n. Cela signifie que $n \in Sp(\theta_{j_0})$. Comme n est également dans F_{j_0}, on a :

$$n \in \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap F_j).$$

On vient donc de montrer :

$$\mathbb{N} \backslash Sp(\varphi) \subseteq \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap F_j).$$

$\boxed{\supseteq :}$ Soit n dans $\bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap F_j)$. Alors il existe un j_0 tel que $n \in Sp(\theta_{j_0}) \cap F_{j_0}$. Les F_j étant disjoints, ce j_0 est unique. L'entier n est dans $Sp(\theta_{j_0})$, donc il existe un ensemble S_0 de classes de k-équivalence satisfaisant $\neg\varphi$ et ayant au moins un représentant de taille n. Si $n \in Sp(\varphi)$, on aurait une classe $\mathcal{C}$ ayant au moins un représentant de taille n et satisfaisant φ. Donc $\mathcal{C} \notin S_0$. Cela implique $N_{\sigma,k}(n) \geq j_0 + 1$, ce qui est contradictoire avec $j_0 \in F_{j_0}$. Donc $n \notin Sp(\varphi)$. D'où :

$$\mathbb{N} \backslash Sp(\varphi) \supseteq \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap F_j).$$

$\square$

Retour à la preuve du théorème : si chacun des ensembles $(F_j)_{j \in \{0,\ldots,M_{\sigma,k}\}}$ est reconnaissable en temps exponentiel non déterministe, alors chaque F_j est un spectre. Dès lors, l'ensemble $\bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap F_j)$ est un spectre. On a donc montré que la conjecture de Ash ultrafaible implique la conjecture du Spectre.

$\boxed{\text{Conjecture du Spectre} \Rightarrow \text{Conjecture de Ash ultrafaible}}$

On se fixe σ et $k \geq 1$. On se donne φ une anti-tautologie de profondeur de quantification k (on peut toujours en construire). On considère les formules $(\theta_j)_{j \in \{0,\ldots,M_{\sigma,k}\}}$, construites comme précédemment :

$$\theta_j = \bigvee_{S \in X_j} \bigwedge_{\mathcal{C}_i \in S} \Psi'_i.$$

Pour tout $j \in \{0, \ldots, M_{\sigma,k}\}$, on pose :

$$A_j = \{n \in \mathbb{N} / N_{\sigma,k} \geq j\}.$$

On remarque que $A_j = \bigcup_{i=1}^{j} F_i$, ou encore, $F_j = A_j \backslash A_{j+1}$.

On s'appuie sur le lemme suivant :

Lemme 7

$$A_j = Sp(\theta_j).$$

Preuve : (du lemme)

$\boxed{\supseteq :}$ Si $n \in Sp(\theta_j)$, alors il existe un ensemble S_0 de j classes de k-isomorphisme qui satisfont $\neg\varphi$ et ont au moins un représentant de taille n. Il y a donc au moins j classes de k-isomorphisme ayant un représentant de taille n. C'est-à-dire $n \in A_j$.

$\boxed{\subseteq :}$ Si $n \in A_j$, il y a au moins j classes de k-isomorphisme ayant un modèle de taille n. Soit S_0 un ensemble de j de ces classes. Comme φ est une anti-tautologie, chacune des classes de S_0 satisfont $\neg\varphi$, et donc θ_j a un modèle de taille n. C'est-à-dire $n \in Sp(\theta_j)$.

□

Retour à la preuve du théorème : on a

$$F_j = Sp(\theta_j) \backslash Sp(\theta_{j+1}),$$

ce qui s'écrit également

$$F_j = Sp(\theta_j) \cap (\mathbb{N} \backslash Sp(\theta_{j+1})).$$

Or d'après la conjecture du Spectre, $\mathbb{N} \backslash Sp(\theta_{j+1})$ est un spectre, donc F_j est un spectre et par conséquent il appartient à la classe NE, d'après le Théorème 9.

□

La preuve de ce théorème nous donne des précisions supplémentaires sur les ensembles F_j : si la conjecture de Ash ultrafaible est vraie, ces ensembles F_j sont des spectres.

Lorsque la conjecture de Ash constante est vérifiée, cela signifie qu'il y a un des F_j qui est de la forme $\Delta \cup \{n/n \geq n_0\}$, et les autres F_j sont finis. Plus précisément, si la fonction de Ash est ultimement constante et de valeur r, alors

$$\mathbb{N} \backslash Sp(\varphi) = Sp(\theta_r).$$

Cela nous permet de préciser dès lors que le complémentaire du spectre de φ est :
- un spectre,
- d'un énoncé de profondeur de quantification égale à celle de φ,
- sur un langage qui est une multiplication du langage de φ, c'est-à-dire ne contient que des relations de mêmes arités que celles du langage de φ.

Lorsque la conjecture de Ash périodique est vérifiée, cela signifie qu'il y a un nombre donné de F_j qui sont des ensembles arithmétiques (de la forme $a\mathbb{N} + b$), et les autres sont finis. Si $r_1, \ldots, r_s$ sont les valeurs atteintes par la fonction de Ash lorsqu'elle devient périodique, on a

$$\mathbb{N} \backslash Sp(\varphi) = \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap (a_j\mathbb{N} + b_j)).$$

Il arrive que l'on puisse considérer les ensembles arithmétiques comme des spectres du même langage que φ, c'est le cas notamment pour un graphe de bijection ou une équivalence. Dans ce cas, on peut conclure sur la nature du complémentaire du spectre de φ de manière analogue au cas constant.

Chapitre 11

Angles d'attaque de la conjecture de Ash et de la conjecture du Spectre

Ce chapitre présente différentes observations et résultats pouvant orienter la recherche d'une confirmation ou d'une infirmation de la conjecture de Ash ultrafaible, et donc de la conjecture du Spectre.

Les conjectures de Ash supposent que le langage est quelconque, ainsi que la profondeur de quantification. On peut donc commencer par étudier la conjecture pour l'un de ces paramètres fixé. Lorsque nous fixons k, on a peu de moyens d'étudier le cas général, pour tout langage, sauf éventuellement pour les cas $k = 1$ et $k = 2$ (voir Chapitre 12). Lorsque nous fixons les arités des relations du langage σ, nous effectuons une restriction **syntaxique** sur les cas étudiés. C'est l'objet du Paragraphe 11.1.

Etant donnée l'ampleur de la tâche restant à accomplir, même après restriction syntaxique, nous pouvons également effectuer des restrictions **sémantiques**, c'est-à-dire intégrer à notre étude certaines propriétés des relations du langage, et ainsi se munir d'une théorie $\mathcal{T}$. Au lieu d'étudier le comportement asymptotique du nombre de classes de k-isomorphisme de σ-structures de cardinal n, on étudie le comportement asymptotique du nombre de classes de k-isomorphisme de σ-structures de cardinal *n qui sont modèles de cette théorie $\mathcal{T}$*. C'est l'objet du Paragraphe 11.2.

Certaines de ces restrictions sémantiques ou syntaxiques ne sont pas limitantes, dans le sens où une réponse positive à la conjecture de Ash ou du Spectre, dans ce cas restreint, implique une réponse positive dans le cas général. Au Paragraphe 11.3, nous présentons les pistes qui nous ont semblées "abordables". Les résultats intermédiaires jalonnant ces pistes font l'objet des Chapitres 12, 13 et 14.

11.1 Restrictions syntaxiques

Commençons par analyser les restrictions syntaxiques. Ash règle le cas des relations unaires dans [Ash94]. Dans ce cas, la fonction de Ash est ultimement constante. Malheureusement, on ne peut pas exprimer beaucoup de propriétés avec uniquement des relations unaires. Nous considérons donc également le cas où le langage est constitué de relations binaires. Ce cas n'est pas résolu, car il est possible de ramener le cas général à ce cas binaire ([Fag75]), comme nous le verrons au Paragraphe 11.1.2.

11.1.1 Le cas unaire

Ash ([Ash94]) montre le résultat suivant :

Theorème 13
Soit σ un langage fini constitué de symboles de relations unaires et de constantes. Alors la fonction de Ash $n \mapsto N_{\sigma,k}(n)$ est ultimement constante.

Preuve : On suppose que le langage σ est constitué de p symboles de relations unaires et de l symboles de constantes. Toute σ-structure $\mathcal{A}$ est partitionnée en au plus $2^p + l$ composantes dont les éléments sont $(k-1)$-isomorphes. On pose $K = k \times (2^p + l)$, et on suppose qu'il y a $h = N_{\sigma,k}(m)$ structures $\mathcal{A}_1, \ldots, \mathcal{A}_h$ de cardinal $m \geq K$ qui sont non-k-isomorphes. Pour tout $i \in \{1, \ldots, h\}$, il y a au moins une composante C de $\mathcal{A}_i$ qui a plus de k éléments, et on construit une nouvelle structure $\mathcal{B}_i$ en ajoutant un élément à cette composante. Montrons, en utilisant un jeu d'Ehrenfeucht en k coups, que $\mathcal{A}_i$ et $\mathcal{B}_i$ sont k-isomorphes. On définit une stratégie gagnante pour Joueur II de la façon suivante :
- Si Joueur I choisit un élément de $\mathcal{A}_i$ qui est dans une composante distincte de C, alors Joueur II choisit le même dans $\mathcal{B}_i$ (et inversement).
- Si Joueur I choisit un élément de $\mathcal{A}_i$ qui est dans C et qui n'a pas encore été choisi, alors Joueur II choisit dans $\mathcal{B}_i$ un élément de C qui n'a pas été choisi (et inversement). Ce choix est toujours possible car on joue en k coups et il y a plus de k éléments dans la composante C, dans $\mathcal{A}_i$ comme dans $\mathcal{B}_i$.

C'est une stratégie gagnante puisque les éléments d'une même composante sont $(k-1)$-isomorphes.

Donc $\mathcal{A}_i$ et $\mathcal{B}_i$ sont k-isomorphes, et par conséquent il y a au moins h structures non k-isomorphes deux à deux et de cardinal $m+1$. On a donc $N_{\sigma,k}(m) \leq N_{\sigma,k}(m+1)$. La suite d'entiers $(N_{\sigma,k}(m))_{m \geq M}$ est donc croissante, et majorée par $M_{\sigma,k}$, elle est donc ultimement constante.

$\square$

La conséquence de ce théorème est que la conjecture du Spectre est vraie pour les langages unaires.

Proposition 16
Soit σ un langage fini constitué de symboles de relations unaires et de constantes. Soit φ un énoncé du langage σ. Le complémentaire de $Sp(\varphi)$ est un spectre de même type.

11.1.2 Le cas binaire

Dans le cas où le langage σ est constitué de relations binaires quelconques, on ne peut pas conclure en utilisant les mêmes techniques que pour le cas unaire. D'autre part, on dispose du Théorème 14, dû à Fagin ([Fag75]), qui nous dispense de considérer les autres cas que le cas binaire, et même plus précisément, que le cas d'une seule relation binaire.

Theorème 14
Soit σ un langage relationnel fini dont tous les prédicats sont d'arité k. Si S est le spectre d'un σ-énoncé, alors $\{n^k/n \in S\}$ est le spectre d'un $\{BIN, =\}$-énoncé, où BIN est un symbole de relation binaire.

Fagin montre en fait un résultat plus général qui s'applique aux spectres généralisés, et dont la preuve apparaît dans [Fag75]. Il montre également le résultat suivant :

Proposition 17
Soit $k \geq 1$ et S un spectre. Alors l'ensemble $\{n/n^k \in S\}$ est un spectre.

La preuve donnée par Fagin consiste en une exhibition de l'énoncé dont $\{n/n^k \in S\}$ est le spectre, mais la caractérisation de Jones et Selman ([JS74]) rend cette proposition immédiate : pour une entrée n donnée, on calcule n^k puis on reconnaît si n^k est dans S, le tout se faisant dans NE. Alors l'ensemble $\{n/n^k \in S\}$ est dans NE. C'est donc un spectre.

La conséquence de ces affirmations est la suivante : si on arrive à montrer que, pour toute profondeur de quantification et pour tout langage constitué d'un seul prédicat binaire, la conjecture de Ash ultrafaible est vraie, alors cela signifie que le complémentaire d'un spectre de $\{BIN, =\}$-énoncé est un spectre, d'après le Théorème 12. Par ailleurs, si S est un spectre quelconque, on peut se ramener au cas où tous les prédicats du langage sont de même arité k. Alors, l'ensemble $B = \{n^k/n \in S\}$ est un spectre de $\{BIN, =\}$-énoncé et le complémentaire de cet ensemble est un spectre. Le complémentaire de S est l'ensemble : $\{n/n^k \notin B\}$. On reconnaît cet ensemble en calculant n^k puis en testant si $n^k \in \mathbb{N} \backslash B$. Le tout se fait dans NE, on a donc un spectre.

On vient de montrer que le cas général pouvait se ramener au cas d'un seul prédicat binaire. Cela signifie que, d'une part si nous montrons la conjecture du Spectre dans le cas binaire, nous la montrons également pour le cas général, et d'autre part s'il existe un contre-exemple (un spectre tel que son complémentaire n'est pas un spectre), alors il existe un contre-exemple dans BIN.

On considèrera donc dans la suite des langages comportant uniquement des prédicats d'arité 2.

11.2 Restrictions sémantiques

Les restrictions sémantiques nécessitent un léger remaniement des définitions : on se fixe une théorie $\mathcal{T}$ sur un langage donné σ, et k une profondeur de quantification donnée. On considère la restriction $\mathcal{T}_k$ de $\mathcal{T}$ aux énoncés de profondeur de quantification k. On compte les classes de k-isomorphisme de modèles de $\mathcal{T}_k$ de cardinal n. La fonction de Ash compte donc à présent des modèles de la théorie à profondeur de quantification donnée et non pas des structures quelconques. Nous sommes donc amenés à décrire les classes de k-isomorphisme de modèles de cette théorie à profondeur de quantification k. Ensuite, pour chaque classe de k-isomorphisme, il va nous falloir regarder quels sont les cardinaux des éléments de cette classe. Enfin, pour chaque cardinal n, on regarde quelles sont les classes représentées par un modèle de cardinal n.

11.2.1 Fonction de Ash d'une théorie et conjecture du spectre

Définition 74 (Fonction de Ash $N_{\mathcal{T},k}$ d'une théorie)
Soit $\mathcal{T}$ une théorie sur un langage relationnel fini σ. La fonction $N_{\mathcal{T},k}$ est la fonction qui, à un cardinal donné n, associe le nombre de classes de k-isomorphisme de σ-structures qui ont un représentant de cardinal n et qui sont modèles de $\mathcal{T}_k$. Cette fonction est appelée la **fonction de Ash** *de la théorie $\mathcal{T}$ au rang k.*

On peut énoncer de la même manière que dans le Chapitre 10 des conjectures sur le comportement asymptotique de cette fonction, mais nous devons préciser les conséquences que ces conjectures peuvent avoir sur la conjecture du Spectre.

Conjecture 5 (Conjecture de Ash ultrafaible pour une théorie)
Soit un langage σ, $k \geq 1$, et $\mathcal{T}$ une théorie sur le langage σ. On note pour tout $i \in \mathbb{N}$:

$$F_{\mathcal{T},i} = \{n \in \mathbb{N} / N_{\mathcal{T},k}(n) = i\}.$$

Pour tout $i \in \{0, \ldots, M_{\sigma,k}\}$, l'ensemble $F_{\mathcal{T},i}$ est dans NE.

Remarque 23 *A condition que la théorie admette au rang k un modèle de taille n pour tout $n \in \mathbb{N}$, les $F_{\mathcal{T},i}$ forment une partition de $\mathbb{N}$. Ils sont vides pour $i > M_{\sigma,k}$.*

On suppose désormais que la théorie $\mathcal{T}$ admet au rang k un modèle de taille n pour tout $n \in \mathbb{N}$. On fixe un énoncé de profondeur de quantification k sur le langage σ, et impliquant $\mathcal{T}_k$. On définit les énoncés $\theta_{\mathcal{T},j}$ de façon analogue aux énoncés θ_j du Chapitre 10, à ceci près qu'on impose que les classes de k-isomorphisme contenues dans les ensembles X_j satisfont $\mathcal{T}_k$.

Pour $j \in \{0, \ldots, M_{\sigma,k}\}$, on pose :

$X_{\mathcal{T},j} = \{S$ ensemble de classes de k-isomorphisme / $|S| = j$ et $\forall \mathcal{C} \in S, \mathcal{C} \models \mathcal{T}_k \cup \{\neg\varphi\}\}$.

On pose à présent, pour tout $j \in \{1, \ldots, M_{\sigma,k}\}$:

$$\theta_{\mathcal{T},j} = \bigvee_{S \in X_{\mathcal{T},j}} \bigwedge_{\mathcal{C}_i \in S} \Psi'_i.$$

On est alors en mesure de montrer le lemme analogue au Lemme 6 du Chapitre 10.

Lemme 8

$$\mathbb{N} \backslash Sp(\varphi) = \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_{\mathcal{T},j}) \cap F_{\mathcal{T},j}).$$

Preuve : (du lemme)

$\boxed{\subseteq :}$ Soit $n \notin Sp(\varphi)$. Soit $j_0 \in \{1, \ldots, M_{\sigma,k}\}$ tel que $n \in F_{\mathcal{T},j_0}$ (ce j_0 existe et est unique). On a alors un ensemble S_0 de j_0 classes de k-isomorphisme $\mathcal{C}_i$ qui ont chacune au moins un représentant de taille n et dont tous les représentants sont modèles de $\mathcal{T}_k$, par définition de $F_{\mathcal{T},j_0}$.
Puisque $n \notin Sp(\varphi)$, on ne peut pas avoir, pour aucun des $\mathcal{C}_i$ de l'ensemble S_0, $\mathcal{C}_i \models \varphi$. Donc pour tout $\mathcal{C}_i$ de l'ensemble S_0, on a $\mathcal{C}_i \models \neg\varphi$.
Autrement dit, si M est une structure de taille n, telle que, pour tout i tel que $\mathcal{C}_i \in S_0$, M est un modèle de $\mathcal{C}_i$ pour une interprétation du langage σ'_i, alors M est un modèle de taille n de $\theta_{\mathcal{T},j_0}$. Or un tel modèle existe, puisque toutes les classes $\mathcal{C}_i \in S_{j_0}$ ont au moins un représentant de taille n et tous leurs représentants sont modèles de $\mathcal{T}_k$. Cela signifie que $n \in Sp(\theta_{\mathcal{T},j_0})$. Comme n est également dans $F_{\mathcal{T},j_0}$, on a :

$$n \in \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_{\mathcal{T},j}) \cap F_{\mathcal{T},j}).$$

On vient donc de montrer :

$$\mathbb{N} \backslash Sp(\varphi) \subseteq \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_{\mathcal{T},j}) \cap F_{\mathcal{T},j}).$$

$\boxed{\supseteq :}$ Soit n dans $\bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_{\mathcal{T},j}) \cap F_{\mathcal{T},j})$. Alors il existe un j_0 tel que $n \in Sp(\theta_{\mathcal{T},j_0}) \cap F_{\mathcal{T},j_0}$. Les $F_{\mathcal{T},j}$ étant disjoints, ce j_0 est unique. L'entier n est dans $Sp(\theta_{\mathcal{T},j_0})$, donc il existe un ensemble S_0 de classes de k-équivalence satisfaisant $\neg\varphi$, dont tous les représentants sont modèles de $\mathcal{T}_k$, et ayant au moins un représentant de taille n. Si $n \in Sp(\varphi)$, on aurait une classe $\mathcal{C}$, dont tous les représentants sont modèles de $\mathcal{T}_k$, ayant au moins un représentant de taille n et satisfaisant φ. Donc $\mathcal{C} \notin S_0$. Cela implique $N_{\mathcal{T},k}(n) \geq j_0 + 1$, ce qui est contradictoire avec $j_0 \in F_{\mathcal{T},j_0}$. Donc $n \notin Sp(\varphi)$. D'où :

$$\mathbb{N} \backslash Sp(\varphi) \supseteq \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_{\mathcal{T},j}) \cap F_{\mathcal{T},j}).$$

□

Ce lemme nous permet de démontrer un analogue du Théorème 12, dans le cas restreint à une théorie.

Theorème 15
Soit $\mathcal{T}$ une théorie sur un langage σ qui admet au rang k des modèles de cardinal n pour tout $n \in \mathbb{N}$. La conjecture de Ash ultrafaible au rang k pour la théorie $\mathcal{T}$ est vraie si et seulement si pour tout énoncé φ de profondeur de quantification inférieure ou égale à k impliquant $\mathcal{T}_k$, le complémentaire de $Sp(\varphi)$ est un spectre.

Preuve : La preuve est en tout point analogue à celle du Théorème 12.

□

On peut faire les mêmes remarques que dans le cas du Théorème 12 : si le comportement de la fonction de Ash est relativement régulier (asymptotiquement constante ou asymptotiquement périodique), cela nous donne des informations supplémentaires sur le complémentaire du spectre. Notamment, lorsque la fonction de Ash est ultimement constante et de valeur r, alors :

$$\mathbb{N} \backslash Sp(\varphi) = Sp(\theta_{\mathcal{T},r}).$$

Cela nous permet de préciser dès lors que le complémentaire du spectre de φ est :
- un spectre,
- d'un énoncé ψ de profondeur de quantification égale à celle de φ,
- sur un langage qui est une multiplication du langage de φ, c'est-à-dire qu'il ne contient que des relations de mêmes arités que celles du langage de φ,
- et cet énoncé est cohérent avec la théorie $\mathcal{T}_k$ par exemple, si la théorie exprime que les relations sont des relations d'équivalence, alors les relations dans l'énoncé ψ sont également des relations d'équivalence.

Lorsque la conjecture de Ash périodique est vérifiée, on a

$$\mathbb{N} \backslash Sp(\varphi) = \bigcup_{j=1}^{M_{\sigma,k}} (Sp(\theta_j) \cap (a_j \mathbb{N} + b_j)),$$

et cela nous permet de conclure dans certains cas sur la nature du spectre complémentaire.

Revenons quelques instants sur le résultat de Fagin, énoncé dans le Théorème 14 du Paragraphe 11.1.2. La relation BIN est une relation de graphe non orienté. Cela signifie qu'il s'agit d'une relation symétrique et irréflexive, ce qui s'exprime par la théorie :

$$\mathcal{T}_{Graphe} = Th(\forall x(\neg R(x,x) \wedge \forall y[(R(x,y) \rightarrow R(y,x)])).$$

Le Théorème 14 et le Théorème 15 signifient que, si l'on arrive à montrer que la fonction de Ash de cette théorie satisfait la conjecture de Ash ultrafaible, le complémentaire d'un spectre de BIN-énoncé est un spectre, et par conséquent, le complémentaire de tout spectre est un spectre. C'est pourquoi on va s'intéresser à des théories qui se rapprochent de la théorie d'un graphe non orienté.

On peut remarquer de plus que lorsque la théorie est constituée d'un seul énoncé, les restrictions au rang de quantification k de la théorie n'ont plus lieu d'être, pour k supérieur ou égal à la profondeur de quantification de la théorie. C'est le cas de la théorie d'un graphe non orienté, et de tous les exemples que nous étudions aux Chapitres 13 et 14.

11.2.2 Rôle des destinées

Les destinées interviennent dans ce problème de la conjecture de Ash, de manière assez naturelle, dans l'étape de description des classes de k-isomorphisme. Nous utilisons également beaucoup les jeux d'Ehrenfeucht pour démontrer les critères de caractérisation des classes de k-isomorphisme. Les destinées nous permettent d'avoir une intuition sur ces critères : la description via les destinées des configurations potentielles nous permet de décrire les objets que l'on peut retrouver dans les structures, par exemple les cycles dans le cas de la théorie d'un graphe de bijection. Dans tous les exemples qui sont présentés par la suite, les destinées nous ont été utiles pour déterminer ce qu'il est possible d'exprimer à un rang de quantification donnée, par exemple la longueur des cycles pour un graphe de bijection, le nombre de cycles que l'on peut discriminer au rang k, et ainsi de suite. Elles nous ont également permis de déterminer une méthodologie d'attaque de ces problèmes, grâce à leur étude exhaustive aux rang 3, 4, ..., jusqu'à l'explosion combinatoire trop importante du nombre de cas. En revanche, pour ce qui est des preuves de la caractérisation des classes de k-isomorphisme, les jeux nous ont paru mieux adaptés, étant donnée la simplicité relative de la description d'un jeu entre deux structures en comparaison de la description d'une destinées exhaustive et essentielle de chacune des structures. On peut résumer en disant que les destinées sont un excellent outil d'intuition, et les jeux un excellent outil de preuve.

Voyons un exemple : on considère, pour $k = 2$, le langage $\sigma = \{<, =\}$, et la théorie $\mathcal{T}$ d'un ordre total. On cherche, pour un cardinal n donné, combien il y a de classes de 2-isomorphisme de modèles de $\mathcal{T}$ de taille n. Pour cela, on décrit les classes de 2-isomorphisme, en donnant les 2-destinées non isomorphes possibles, puis en étudiant les conséquences de ce qui est écrit dans la destinée sur le cardinal du modèle. Pour cela, on essaie de construire toutes les 2-destinées (réduites) non-isomorphes de σ-structures finies.

Une 2-destinée d'une telle structure est un arbre de hauteur 3, de racine $\emptyset$, et les fils de $\emptyset$ sont représentés par une variable x. On a, pour un fils x de $\emptyset$, quatre possibilités de sous-arbres de rang 1, représentées Figure 11.1.

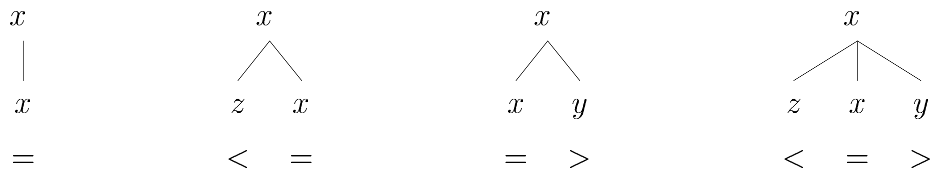


FIG. 11.1 – Possibilités de sous-arbres de rang 1 dans la 2-destinée réduite d'une $\{<,=\}$-structure finie.

Ensuite, dans la destinée, certains de ces sous-arbres peuvent apparaître ou non. Par exemple, le premier sous-arbre exclut tous les autres (il signifie qu'il n'y a qu'un seul élément dans la structure, puisque l'ordre est total). D'autre part, dans une structure finie de cardinal au moins 2, munie d'un ordre total, il y a toujours un plus petit élément (troisième sous-arbre) et un plus grand élément (deuxième sous-arbre). Ce sont les seuls éléments si la structure est de cardinal 2. Si la structure est de cardinal supérieur, il y a également des éléments intermédiaires (quatrième sous-arbre). Il y a donc 3 destinées pour la hauteur 2 qui sont des destinées de structures finies (voir Figure 11.2).

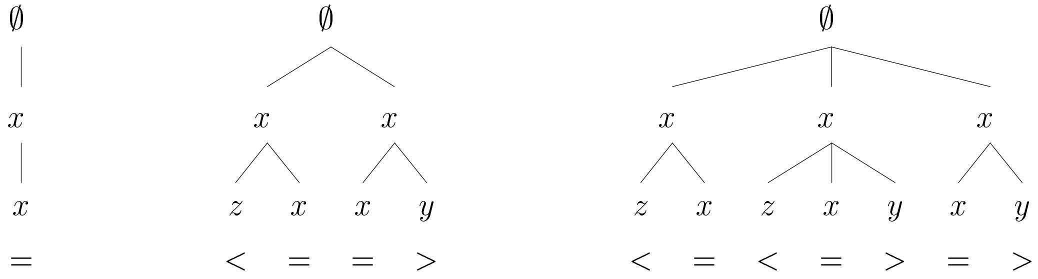


FIG. 11.2 – Possibilités de 2-destinées réduites d'une $\{<,=\}$-structure finie.

On a donc trois classes de 2-isomorphisme pour les $\{<,=\}$-structures finies, représentées chacune par une de ces trois 2-destinées réduites. Il reste à étudier la cardinalité des représentants de ces classes :

- la première classe est celle d'une structure nécessairement à un seul élément, il n'y a donc dans cette classe que des structures de cardinal 1 ;
- la deuxième classe est celle d'une structure nécessairement à deux éléments, il n'y a donc dans cette classe que des structures de cardinal 2 ;

– la troisième classe est celle de structures ayant au moins trois éléments, et il y a des représentants de cette classe pour tout cardinal $n \geq 3$.

Pour tout $n \geq 1$, on a donc exactement une classe de 2-isomorphisme de modèles de $\mathcal{T}$ ayant des représentants de cardinal n, la fonction $N_{\mathcal{T},k}(n)$ est donc trivialement ultimement constante (à partir de $n = 1$) et de valeur 1. Dans cette théorie et à profondeur de quantification 2, on vérifie donc la conjecture de Ash initiale. Cela ne nous donne évidemment aucune indication sur sa validité dans le cas général.

11.2.3 Rôle de l'égalité

L'énoncé des conjectures de Ash considère toujours un langage égalitaire. En fait, le cas des langages non égalitaires est assez trivial, à condition qu'on ne puisse pas définir l'égalité dans la théorie considérée (par exemple, on peut définir l'égalité avec une bijection, ou avec la relation de divisibilité).

Proposition 18
Soit σ un langage ne contenant pas l'égalité. Alors pour tout $k \geq 1$, la fonction de Ash

$$n \mapsto N_{\sigma,k}(n)$$

est ultimement constante.

Preuve : Soit σ un langage non égalitaire, composé de s relations $R_1, \ldots, R_s$ d'arités respectives $r_1 \leq \ldots \leq r_s$. Il y a un nombre fini de classes de $(k-1)$-isomorphisme de singletons, notons ce nombre S. Soit $\mathcal{A}$ une σ-structure finie de cardinal $n > 2S$. Ainsi, il y a au moins deux éléments de $\mathcal{A}$ qui sont $(k-1)$-isomorphes. Notons $x_1, \ldots, x_t$ les éléments de cette classe de $(k-1)$-isomorphisme. On construit la σ-structure $\mathcal{B}$ de cardinal $n+1$ en ajoutant un élément x_0 à $\mathcal{A}$ de la façon suivante : pour tout $i \in \{1, \ldots, s\}$, si $y_1, \ldots, y_{r_i}$ sont des éléments de $\mathcal{A}$, alors, pour toute façon de placer une variable ou plusieurs parmi r_i, on a $R_i(y_1, \ldots, x_0, \ldots, y_{r_i})$ si et seulement si on a $R_i(y_1, \ldots, x_1, \ldots, y_{r_i})$. Le nouvel élément x_0 satisfait donc les mêmes relations avec les éléments du domaine que x_1. On ne peut pas distinguer x_1 et x_0, puisqu'ils ont le même type de $(k-1)$-isomorphisme et que le langage ne contient pas l'égalité. On peut donc définir une stratégie gagnante pour Joueur II dans le jeu d'Ehrenfeucht en k coups entre les structures $\mathcal{A}$ et $\mathcal{B}$:
– Au premier coup : si Joueur I choisit un élément y de $\mathcal{A}$, Joueur II choisit le même élément dans $\mathcal{B}$. Si Joueur I choisit un élément $y \neq x_0$ dans $\mathcal{B}$, Joueur II choisit le même élément dans $\mathcal{A}$. Si Joueur I choisit x_0 dans $\mathcal{B}$, Joueur II choisit x_1 dans $\mathcal{A}$.
– ...
– Au coup k : si Joueur I choisit un élément y de $\mathcal{A}$, Joueur II choisit le même élément dans $\mathcal{B}$. Si Joueur I choisit un élément $y \neq x_0$ dans $\mathcal{B}$, Joueur II choisit le même élément dans $\mathcal{A}$. Si Joueur I choisit x_0 dans $\mathcal{B}$, Joueur II choisit x_1 dans $\mathcal{A}$.

Les k-uples choisis respectivement dans $\mathcal{A}$ et $\mathcal{B}$ satisfont les mêmes σ-formules sans quantificateurs, étant donné qu'on ne peut pas distinguer x_0 de x_1 et que l'on a définit les relations de x_0 avec les éléments de $\mathcal{A}$ de la même façon que les relations de x_1 avec ces mêmes éléments.

Alors, les structures $\mathcal{A}$ et $\mathcal{B}$ sont k-isomorphes, et $\mathcal{B}$ est de cardinal égal au cardinal de $\mathcal{A}$ plus un. La fonction de Ash est croissante et bornée, donc ultimement constante.

□

Lorsque l'on rajoute une restriction sémantique, ce résultat reste partiellement vrai. En effet, pour les théories dans lesquelles l'égalité est définissable, par exemple grâce à un énoncé de profondeur de quantification k, on ne peut pas garantir la constance de la fonction de Ash pour des profondeurs supérieures ou égales à k. En revanche, lorsque l'égalité n'est pas définissable, le résultat reste valable. Nous en verrons un exemple au Paragraphe 14.1 du Chapitre 14.

11.2.4 Une intuition

En considérant un certain nombre d'exemples, nous avons dégagé l'intuition suivante : moins la théorie est expressive, (*i.e.* elle est plus contrainte, donc les axiomes fixent davantage l'interprétation du langage), plus la fonction de Ash est irrégulière. A l'inverse, plus la théorie est expressive (*i.e.* elle est moins contrainte, c'est-à-dire qu'on peut parler au sein de cette théorie de nombreuses autres choses que ce dont semblent parler les axiomes a priori), et plus le comportement de la fonction de Ash est lisse.

Par exemple, si on considère la théorie des algèbres de Boole finies : pour un cardinal n donné, on a au plus une classe de k-isomorphisme (il y a catégoricité de la théorie des algèbres de Boole finies). Si n est une puissance de deux, il y a une classe, si n n'est pas une puissance de 2, il n'y a aucune classe. La fonction de Ash correspond alors à la fonction caractéristique de l'ensemble des puissances de 2 (Figure 11.3).

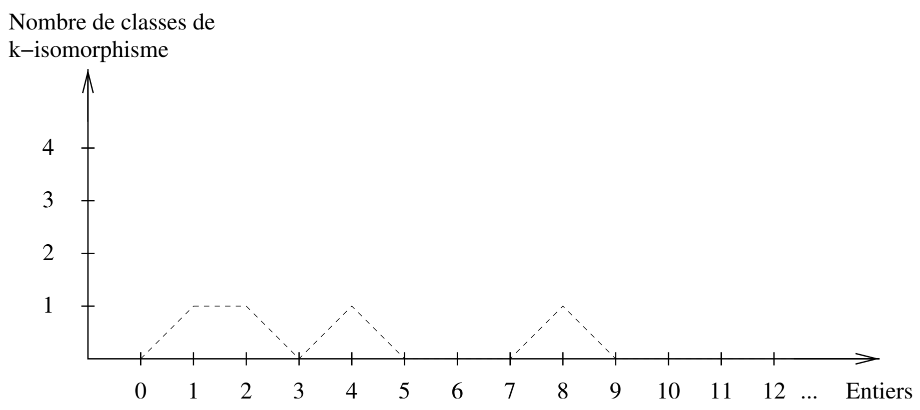


FIG. 11.3 – Fonction de Ash pour la théorie des algèbres de Boole finies.

Cette fonction de Ash vérifie alors la conjecture de Ash ultrafaible, mais ni la conjecture de Ash périodique, ni (encore moins) la conjecture de Ash originelle. En revanche, si on contraint moins le langage, on peut s'autoriser d'autres formes de modèles non isomorphes qui peuvent "compenser" ce comportement, dans une certaine mesure.

On peut pousser plus loin le raisonnement et se demander dans quel cas une fonction peut être une fonction de Ash. Nous avons dégagé la condition suffisante suivante :

Proposition 19
Soit f une fonction de $\mathbb{N}$ dans $\mathbb{N}$ telle que :
- *la fonction f est bornée par une constante M_f,*
- *la fonction f est calculable dans E.*

Alors il existe une signature σ et une profondeur de quantification k telles que $f = N_{\sigma,k}$.

Preuve : On pose, pour tout $i \in \mathbb{N}$:

$$F_i = \{n \in \mathbb{N}/f(n) = i\}.$$

Les F_i forment une partition de $\mathbb{N}$, et sont vides pour $i > M_f$. On a, pour tout $i \leq M_f$, $F_i \in$ E. En effet, $n \in F_i$ si et seulement si $f(n) = i$, donc pour reconnaître si $n \in F_i$, on calcule $f(n)$ et on compare le résultat à i, opération qui se fait dans E.

De plus, Fagin montre dans [Fag74] que tout ensemble de la classe de complexité E est un spectre d'énoncé catégorique. C'est-à-dire que pour tout $i \leq M_f$, il existe un énoncé θ_i tel que :
- $F_i = Sp(\theta_i)$,
- pour tout $n \in \mathbb{N}$, il existe au plus un modèle de θ_i (à isomorphisme près) de cardinal n.

En notant σ_i le langage associé à θ_i (on suppose que pour tout $i \neq j$, on a $\sigma_i \cap \sigma_j \subseteq \{=\}$), on pose :

$$\sigma = (\bigcup_{i \leq M_f} \sigma_i) \cup \{B\},$$

où B est une relation unaire qui n'apparaît dans aucun des σ_i. On pose ensuite, pour tout $i \leq M_f$, et tout $j \leq i$:

$$\varphi_{ij} = \theta_i \wedge Card(B, j),$$

où $Card(B, j)$ est un énoncé qui dit qu'il y a exactement j éléments satisfaisant la relation B. On pose enfin :

$$\varphi_i = \bigvee_{j=1}^{i} \varphi_{ij},$$

et

$$\varphi = \bigvee_{i=1}^{M_f} \varphi_i.$$

On montre alors ci-dessous que
$$N_{\varphi,K} = f,$$
où K est le maximum de la profondeur des θ_i et de M_f.

Soit $n \in \mathbb{N}$. La fonction de Ash $N_{\varphi,K}$ est le nombre de modèles de φ de taille n qui ne sont pas K-isomorphes. On remarque la chose suivante : si $\mathcal{M}$ est un modèle de φ de taille n, alors il existe un unique i et un unique j tels que $\mathcal{M}$ est modèle de φ_{ij}. En effet, comme les F_i sont disjoints et correspondent respectivement aux spectres des θ_i, nécessairement les cardinaux des modèles des θ_i sont disjoints. Donc pour un cardinal donné, il existe un unique i tel que $\mathcal{M}$ soit modèle de θ_i, et c'est le i tel que $n \in F_i$. Donc si $\mathcal{M}$ est modèle de φ, il existe un unique i tel que $\mathcal{M}$ est modèle de φ_i. De plus, les φ_{ij} se distinguent par le nombre exact d'éléments qui satisfont la relation B, et ne peuvent donc avoir un modèle commun. Il existe donc un unique j tel que $\mathcal{M}$ est un modèle de φ_{ij}. Par définition, si $n \in F_i$, on a $f(n) = i$. Il reste donc à montrer que $N_{\varphi,K}(n) = i$. Comme θ_i est catégorique, il existe exactement un modèle de θ_i de cardinal n. Soit $\mathcal{M}_i$ ce modèle. On peut interpréter le prédicat B de i façons différentes pour donner des modèles non K-isomorphes de φ_i, donc de φ. Donc $N_{\varphi,K}(n) \geq i$. D'autre part, on a vu que si $\mathcal{M}$ est un modèle de φ, c'est obligatoirement l'un des modèles des φ_{ij}, et donc correspond au modèle $\mathcal{M}_i$ avec une des interprétations de B. Donc $N_{\varphi,K}(n) = i = f(n)$.

$\square$

Ce résultat nous dit donc que, pour toute fonction "raisonnablement compliquée" (bornée et dans E), on peut trouver une théorie pour laquelle c'est la fonction de Ash. Ceci dit, la théorie en question est loin d'être naturelle, et on ne sait rien dire de son expressivité, si ce n'est qu'elle est réduite. Ce résultat n'infirme en rien notre intuition. Pour transformer le domaine de l'intuitif en certitude, il faudrait par exemple montrer que la théorie que nous produisons dans cette preuve est strictement moins expressive qu'une théorie de référence, par exemple la théorie de deux fonctions ou la théorie des graphes. Ou bien, montrer qu'on peut trouver un exemple de théorie naturelle expressive dont le comportement de la fonction de Ash est aussi irrégulier que l'on veut (tout en restant dans NE si possible).

11.3 Pistes suivies pour attaquer la conjecture de Ash

En fonction de ces intuitions et observations, nous nous sommes attaqué à plusieurs pistes parallèles pouvant aboutir soit à une confirmation de ces intuitions, soit à un contre-exemple suffisamment significatif pour remettre en question la validité de la conjecture de Ash.

11.3.1 La piste : $k = 2$

La première piste est née de l'observation suivante : il y a une explosion combinatoire dans la construction des destinées entre la hauteur 2 et la hauteur 3, et donc une multiplication du nombre de classes de k-isomorphisme très importante à partir du rang 3, ce qui rend plus difficile l'étude de la fonction de comptage. Il est donc raisonnable de penser que le cas $k = 2$ est beaucoup plus facile à résoudre que les cas 'à profondeur de quantification supérieure. De plus, le pouvoir d'expressivité au rang 2 est généralement assez faible : on peut donc difficilement “parler” du cardinal du modèle avec des énoncés de rang de quantification 2. Nous montrons au Chapitre 12 que notre intuition concernant le cas $k = 2$ est fondée : la conjecture de Ash constante est vérifiée dans ce cas. Il reste la question de savoir si l'on peut généraliser les méthodes utilisées pour $k = 2$ à des profondeurs de quantifications supérieures. Cela est loin d'être évident.

11.3.2 La piste : une relation binaire et l'égalité

La deuxième piste est née du Théorème 14, qui nous dit que tout spectre peut se ramener (par une transformation qui est dans NE) à un spectre d'une seule relation de graphe. D'autre part, une sous-piste possible est donnée par l'observation suivante, faite par Durand, Fagin et Loescher dans [DFL98] : si S est un spectre, alors il existe deux entiers positifs h et k tels que $hS^k = \{hn^k/n \in S\}$ est le spectre d'un énoncé sur un langage composé de deux fonctions (et l'égalité). Le cas de deux fonctions et l'égalité est donc également une piste possible pour conclure sur le cas général. Les différentes sous-pistes suivies sont résumées dans la Figure 11.4. L'étude de ces différentes pistes est présentée au Chapitre 13. Le signe “?” signifie qu'on ne connaît pas le comportement de la fonction de Ash.

11.3.3 La piste : deux relations d'équivalence et l'égalité

La troisième piste est née du fait que tout spectre peut se ramener à un spectre de deux relations d'équivalence et l'égalité, en procédant à un rembourrage polynomial d'un graphe dans un modèle de deux relations d'équivalence. Ce résultat est donné au Chapitre 14. Les sous-pistes suivies sont résumées dans la Figure 11.5. Le signe “?” signifie qu'on ne connaît pas le comportement de la fonction de Ash.

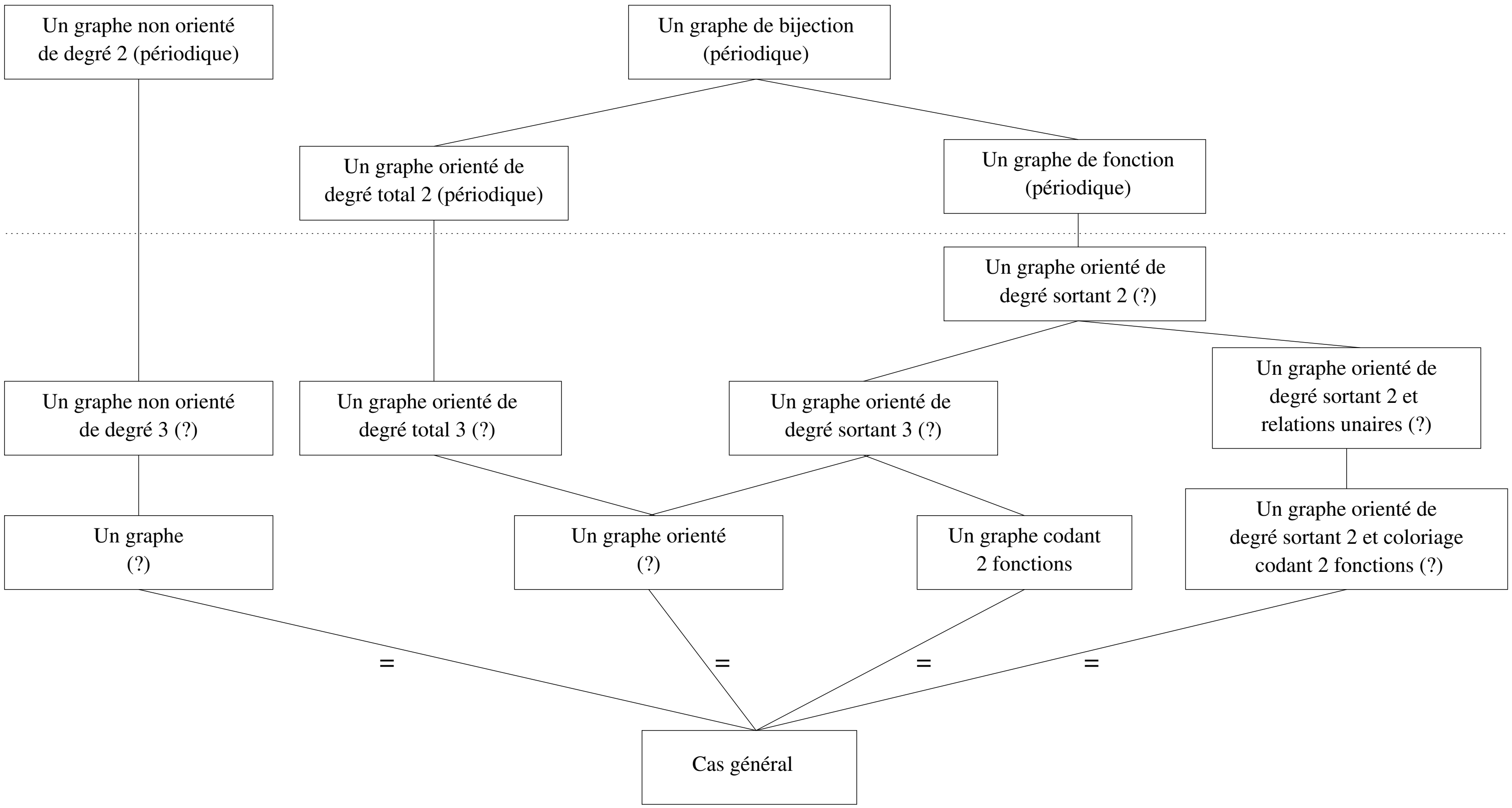


FIG. 11.4 – Les pistes suivies pour l'étude du cas binaire.

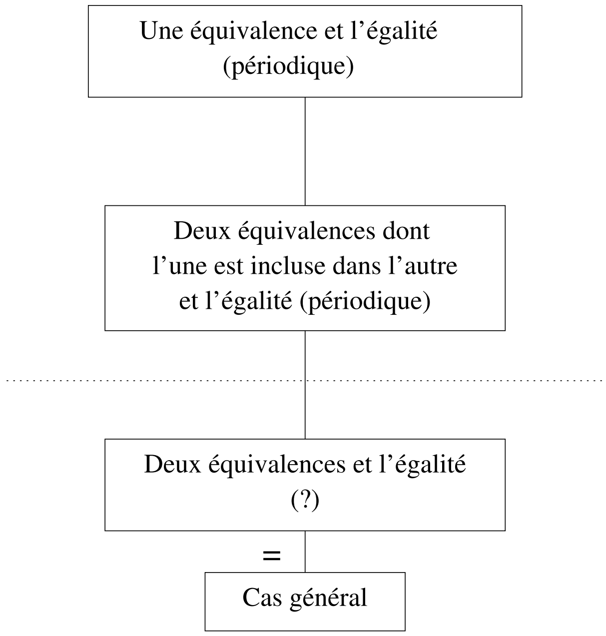


FIG. 11.5 – Les pistes suivies menant à deux équivalences et l'égalité.

Chapitre 12

Le cas $k = 2$

Dans ce chapitre, nous confirmons l'intuition selon laquelle la conjecture de Ash est vérifiée pour $k = 2$. Pour cela, on commence par montrer, au Paragraphe 12.1, que le cas d'un langage d'arité quelconque se ramène au cas de plusieurs relations binaires, à savoir qu'un spectre d'énoncé quelconque de profondeur de quantification 2 est un spectre d'énoncé de relations binaires de profondeur de quantification 2. Ensuite, nous examinons en détail le cas d'une seule relation binaire au Paragraphe 12.2, avant de nous intéresser au cas d'un nombre quelconque de relations binaires, qui nous permet de conclure. Ceci fait l'objet du Paragraphe 12.3.

12.1 Pour $k = 2$, le cas général se ramène au cas de plusieurs relations binaires

Nous commençons par montrer la proposition suivante :

Proposition 20
Soit σ un langage relationnel fini et F un σ-énoncé de profondeur de quantification 2. Alors il existe un énoncé G de profondeur de quantification 2 sur un langage composé uniquement de symboles de relations binaires tel que

$$Sp(G) = Sp(F).$$

Preuve : On suppose que le langage σ est composé de m symboles de constantes $c_1, \ldots, c_m$, de n symboles de relations unaires $u_1, \ldots, u_n$, et de p symboles de relations d'arité supérieure ou égale à 2, notées $R_1, \ldots, R_p$, d'arités respectives $r_1, \ldots, r_p$. On décrit le langage σ' composé de relations binaires uniquement (et de constantes), de la façon suivante :

- il y a m constantes $c_1, \ldots, c_m$;
- il y a n relations binaires $U_1, \ldots, U_n$;
- pour tout $i \in \{1, \ldots, p\}$, il y a $2^{r_i - 1}$ relations binaires $R_i^1, \ldots, R_i^{2^{r_i-1}}$.

Pour tout $i \in \{1, \ldots, p\}$, il y a au plus $2^{r_i - 1}$ façons d'écrire une relation d'arité r_i avec deux variables x et y. A chacune de ces façons, on associe une fois pour toutes un numéro entre 1 et 2^{r_i-1}.

Soit F un énoncé de profondeur de quantification 2. On construit l'énoncé G à partir de F en transformant les sous-formules atomiques de F de la façon suivante :
- si la formule atomique est de la forme $u_i(x)$, on la remplace par $U_i(x, x)$;
- si la formule atomique est de la forme $R_i(., \ldots, .)$, avec pour numéro $j \in \{1, \ldots, 2^{r_i-1}\}$, on la remplace par $R_i^j(., .)$ avec les noms des variables qui correspondent.

Montrons que $Sp(F) = Sp(G)$, par une double inclusion.

Si $n \in Sp(F)$, il existe un modèle de F de cardinal n. Notons $\mathcal{A}$ cette structure. On construit un modèle $\mathcal{B}$ de G de la façon suivante :
- $\mathcal{A}$ et $\mathcal{B}$ ont même domaine, et les constantes sont interprétées de la même façon.
- Si a est un élément du domaine, on a :

$$U_i^{\mathcal{B}}(a, a) \text{ si et seulement si } u_i^{\mathcal{A}}(a),$$

  et pour tous éléments $a \neq b$ du domaine, on a $\neg U_i^{\mathcal{B}}(a, b)$.
- Si a et b sont deux éléments du domaine, on a :

$$R_i^{j\,\mathcal{B}}(a, b) \text{ si et seulement si } R_i{}^{\mathcal{A}}(., \ldots, .),$$

  où l'emplacement des variables a et b dans la relation R_i est celui numéroté j.

Il est clair que $\mathcal{B}$ est un modèle de G.

Si $n \in Sp(G)$, on exécute la manœuvre inverse pour construire un modèle de F de taille n à partir d'un modèle de G de taille n.

□

12.2 Le cas d'une relation binaire et l'égalité

On veut montrer que dans le cas d'un langage restreint à une seule relation binaire (quelconque), la fonction de comptage pour $k = 2$ est ultimement constante, autrement dit que ce cas satisfait la conjecture de Ash.

On se donne $\sigma = \{R, =\}$ avec R une relation binaire quelconque. Calculons le nombre de classes de 2-isomorphisme de modèles, en utilisant les destinées. Un nœud x de rang 1 dans une 2-destinée d'une $\{R\}$-structure finie a un fils "obligatoire" et huit fils "optionnels" :
- le fils obligatoire a même étiquette que le nœud x, et satisfait les mêmes relations avec lui-même que le nœud x avec lui-même ;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $R(y, y)$ et aucune relation avec x ;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $\neg R(y, y)$ et aucune relation avec x ;

- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $R(y, y)$ et $R(x, y)$ mais pas $R(y, x)$;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $\neg R(y, y)$ et $R(x, y)$ mais pas $R(y, x)$;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $R(y, y)$ et $R(y, x)$ mais pas $R(x, y)$;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $\neg R(y, y)$ et $R(y, x)$ mais pas $R(x, y)$;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $R(y, y)$, $R(y, x)$ et $R(x, y)$;
- un fils optionnel y, d'étiquette distincte de celle de x, satisfait $\neg R(y, y)$, $R(y, x)$ et $R(x, y)$.

Le nœud x peut satisfaire $R(x, x)$ ou bien $\neg R(x, x)$. Cela nous donne donc, pour un x donné, $2 \times 2^8 = 2^9$ possibilités. Il y a donc 512 classes de 1-isomorphisme d'éléments. Dans la destinée, chaque sous-arbre possible peut apparaître ou non, mais il y a au moins un sous-arbre. Cela nous donne donc $2^{2^9} - 1$ potentielles 2-destinées, donc autant de classes de 2-isomorphisme possibles. Autrement dit, on a :

$$M_{\{R,=\},2} = 2^{512} - 1.$$

Nous avons la possibilité d'étudier, pour chacune de ces classes, quels sont les cardinaux de leurs représentants. Mais il va sans dire que cela risque de prendre quelques millénaires. Nous choisissons donc une autre méthode : on se donne une classe de 2-isomorphisme, et l'un de ses représentants, puis on montre qu'on peut obtenir une $\{R, =\}$-structure 2-équivalente en ajoutant un seul élément, à condition que le cardinal du représentant initial soit suffisamment grand.

Soit $\mathcal{A}$ une $\{R\}$-structure finie, de cardinal n strictement supérieur à 512, de sorte que l'on est sûr qu'il y a au moins deux éléments de la structure qui sont 1-isomorphes. On énumère les éléments de $\mathcal{A}$: $x_1, \ldots, x_n$, et on suppose que les éléments x_1 à x_s constituent les éléments d'une même classe de 1-isomorphisme entre singletons, avec $s \geq 2$.

On construit la structure $\mathcal{B}$ comme suit : on ajoute au domaine de $\mathcal{A}$ un élément x_0 distinct des x_i, et on interprète la relation R sur le domaine étendu :
- si $j > s$, alors $R(x_0, x_j)$ si et seulement si $R(x_1, x_j)$, et $R(x_j, x_0)$ si et seulement si $R(x_j, x_1)$;
- si $j \leq s$, alors $R(x_0, x_j)$ et $R(x_j, x_0)$ si et seulement si $R(x_1, x_2)$, et $R(x_0, x_0)$ si et seulement si $R(x_1, x_1)$.

Il reste à montrer que la $\{R\}$-structure $\mathcal{B}$ ainsi construite est 2-isomorphe à la structure $\mathcal{A}$.

On définit une stratégie gagnante pour Joueur II dans le jeu entre $\mathcal{A}$ et $\mathcal{B}$ en deux coups :
- Au premier coup :
  - si Joueur I choisit un des x_i avec $i > 0$ dans l'une des structures, Joueur II choisit le même x_i dans l'autre structure,
  - si Joueur I choisit x_0 dans la structure $\mathcal{B}$, Joueur II choisit x_1 dans la structure $\mathcal{A}$.

- Au deuxième coup :
  - si Joueur I choisit un des x_i, sauf x_1 ou x_0, dans l'une des structures, alors Joueur II choisit le même x_i dans l'autre structure,
  - si Joueur I choisit x_0 dans la structure $\mathcal{B}$, alors Joueur II choisit le premier x_i pour $i \leq s$ qui n'a pas été encore choisi dans la structure $\mathcal{A}$ (il en reste au moins un),
  - si Joueur I choisit x_1 dans l'une des structures, alors Joueur II choisit le premier x_i pour $i \leq s$ qui n'a pas été encore choisi dans l'autre structure (il en reste au moins un).

Montrons qu'il s'agit d'une stratégie gagnante. Les couples choisis dans l'une et l'autre structure après le jeu sont de la forme :

Joueur I	Joueur II	Couple choisi dans $\mathcal{A}$	Couple choisi dans $\mathcal{B}$
$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{B}}, {x_j}^{\mathcal{B}})$	$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{B}}, {x_j}^{\mathcal{B}})$
$({x_{i\neq 0}}^{\mathcal{B}}, {x_{j\neq 0}}^{\mathcal{B}})$	$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{B}}, {x_j}^{\mathcal{B}})$
$({x_i}^{\mathcal{A}}, {x_{j\neq 0}}^{\mathcal{B}})$	$({x_i}^{\mathcal{B}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{B}}, {x_j}^{\mathcal{B}})$
$({x_{i\neq 0}}^{\mathcal{B}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{B}})$	$({x_i}^{\mathcal{A}}, {x_j}^{\mathcal{A}})$	$({x_i}^{\mathcal{B}}, {x_j}^{\mathcal{B}})$
$({x_{i\neq 1}}^{\mathcal{A}}, x_0^{\mathcal{B}})$	$({x_{i\neq 1}}^{\mathcal{B}}, {x_1}^{\mathcal{A}})$	$({x_{i\neq 1}}^{\mathcal{A}}, x_1^{\mathcal{A}})$	$({x_{i\neq 1}}^{\mathcal{B}}, {x_0}^{\mathcal{B}})$
$(x_0^{\mathcal{B}}, {x_{i\neq 1}}^{\mathcal{A}})$	$({x_1}^{\mathcal{A}}, {x_{i\neq 1}}^{\mathcal{B}})$	$({x_1}^{\mathcal{A}}, {x_{i\neq 1}}^{\mathcal{A}})$	$(x_0^{\mathcal{B}}, {x_{i\neq 1}}^{\mathcal{B}})$
$({x_{i\neq 0,1}}^{\mathcal{B}}, x_0^{\mathcal{B}})$	$({x_{i\neq 1}}^{\mathcal{A}}, {x_1}^{\mathcal{A}})$	$({x_{i\neq 1}}^{\mathcal{A}}, x_1^{\mathcal{A}})$	$({x_{i\neq 1}}^{\mathcal{B}}, {x_0}^{\mathcal{B}})$
$(x_0^{\mathcal{B}}, {x_{i\neq 0,1}}^{\mathcal{B}})$	$({x_1}^{\mathcal{A}}, {x_{i\neq 1}}^{\mathcal{A}})$	$({x_1}^{\mathcal{A}}, {x_{i\neq 1}}^{\mathcal{A}})$	$(x_0^{\mathcal{B}}, {x_{i\neq 1}}^{\mathcal{B}})$
$({x_1}^{\mathcal{A}}, x_0^{\mathcal{B}})$	$({x_1}^{\mathcal{B}}, {x_2}^{\mathcal{A}})$	$({x_1}^{\mathcal{A}}, x_2^{\mathcal{A}})$	$({x_1}^{\mathcal{B}}, {x_0}^{\mathcal{B}})$
$({x_1}^{\mathcal{B}}, x_0^{\mathcal{B}})$	$({x_1}^{\mathcal{A}}, {x_2}^{\mathcal{A}})$	$({x_1}^{\mathcal{A}}, x_2^{\mathcal{A}})$	$({x_1}^{\mathcal{B}}, {x_0}^{\mathcal{B}})$
$(x_0^{\mathcal{B}}, {x_1}^{\mathcal{A}})$	$({x_1}^{\mathcal{A}}, {x_0}^{\mathcal{B}})$	$({x_1}^{\mathcal{A}}, x_1^{\mathcal{A}})$	$({x_0}^{\mathcal{B}}, x_0^{\mathcal{B}})$
$(x_0^{\mathcal{B}}, {x_1}^{\mathcal{B}})$	$({x_1}^{\mathcal{A}}, {x_2}^{\mathcal{A}})$	$({x_1}^{\mathcal{A}}, x_2^{\mathcal{A}})$	$({x_0}^{\mathcal{B}}, {x_1}^{\mathcal{B}})$

Dans chacun de ces cas, il est clair que les mêmes relations sont satisfaites dans $\mathcal{A}$ et dans $\mathcal{B}$ par les couples choisis.

On vient donc de montrer que si l'on dispose, pour une classe de 2-isomorphisme, d'un représentant de taille $n > 512$, alors on peut exhiber un représentant de cette même classe de taille $n + 1$. Autrement dit, la fonction de comptage $n \mapsto N_{\{R,=\},2}(n)$ est croissante. Comme c'est une fonction à valeurs entières qui est bornée (par $2^{512} - 1$), elle est ultimement constante.

12.3 Le cas de plusieurs relations binaires et l'égalité

Estimons de la même manière le nombre de cas qu'il y aurait à étudier, en fonction du nombre de relations binaires. On suppose qu'il y a dans le langage s relations binaires notées $R_1, \ldots, R_s$. Regardons les sous-arbres de hauteur 1 possibles : pour un nœud de rang 1 donné x, il y a un fils "obligatoire" ayant même étiquette que x, et des fils "optionnels", ayant une étiquette différente de celle de x. Pour les fils optionnel y de x, et chaque relation R_i, on peut avoir les situations :

– $R_i(y,y)$ et aucune relation avec x ;
– $\neg R_i(y,y)$ et aucune relation avec x ;
– $R_i(y,y)$ et $R_i(x,y)$ mais pas $R_i(y,x)$;
– $\neg R_i(y,y)$ et $R_i(x,y)$ mais pas $R_i(y,x)$;
– $R_i(y,y)$ et $R_i(y,x)$ mais pas $R_i(x,y)$;
– $\neg R_i(y,y)$ et $R_i(y,x)$ mais pas $R_i(x,y)$;
– $R_i(y,y)$, $R_i(y,x)$ et $R_i(x,y)$;
– $\neg R_i(y,y)$, $R_i(y,x)$ et $R_i(x,y)$;

On a donc, pour chaque i, huit possibilités pour les fils optionnels. Cela nous donne en tout 8^s possibilités de fils optionnels, qui peuvent être soit présents, soit absents. Comme le nœud x, pour chaque i, peut satisfaire soit $R_i(x,x)$, soit $\neg R_i(x,x)$, cela nous donne $2^s \times 2^{8^s} = 2^{s+8^s}$ possibilités de sous-arbres de rang 1. (Autrement dit, 2^{s+8^s} classes de 1-isomorphisme). Dans la destinée, chaque sous-arbre possible peut apparaître ou non, mais il y a au moins un sous-arbre. Cela nous donne donc $2^{2^{s+8^s}} - 1$ destinées possibles, soit autant de classes de 2-isomorphisme de structures.

On adopte la même méthode que précédemment, à savoir qu'on prend une structure de taille supérieure strictement à 2^{s+8^s}, de manière à pouvoir asurer l'existence d'une classe de 1-isomorphisme de singletons ayant au moins deux éléments dans la structure, puis on ajoute un élément à ce modèle en lui faisant partager les mêmes relations que les éléments de cette classe "nombreuse". On montre enfin, grâce à l'exhibition d'une stratégie gagnante (en fait, la même que celle décrite pour une seule relation binaire), que le nouveau modèle est 2-isomorphe à l'ancien.

On montre donc que si on dispose pour une classe de 2-isomorphisme d'un représentant de taille $n > 2^{s+8^s}$, alors on peut exhiber un représentant de taille $n+1$. Autrement dit, la fonction de comptage $n \mapsto N_{\{R_1,\ldots,R_s,=\},2}(n)$ est croissante. Comme c'est une fonction à valeurs entières qui est bornée (par $2^{2^{s+8^s}} - 1$), elle est ultimement constante.

12.4 Conclusion et perspectives

Nous sommes donc en mesure d'énoncer le résultat suivant :

Theorème 16 (Théorème du Spectre pour $k = 2$)
Soit σ un langage relationnel fini. Soit φ un σ-énoncé de profondeur de quantification au plus 2. Alors le complémentaire du spectre $Sp(\varphi)$ est un spectre d'énoncé de profondeur de quantification 2 sur un langage de relations binaires.

Mortimer précise dans [Mor75] la forme des spectres d'énoncés de profondeur de quantification 2 : ils sont soit finis soit cofinis. On peut également consulter [LB98].

Il est difficile de songer à une généralisation de ce résultat en dimension 3. En réalité, losrque l'on passe au rang de quantification 3, on commence à faire intervenir de manière forte des notions comme le nombre d'éléments dans une classe, et lorsque l'on ajoute un élément à une structure, il faut parfois rajouter d'autres éléments en relation avec celui-ci pour pouvoir rester dans la même classe de 3-isomorphisme. Il est compliqué de déterminer, selon le cas, combien d'éléments il faut ajouter, et si ce nombre est variable ou peut être plus ou moins fixé.

Comment dès lors, aborder la question des profondeurs de quantification supérieures ? Une première question à se poser est la suivante : existe-t'il une profondeur de quantification k telle qu'une réponse positive à la conjecture du Spectre pour cette profondeur de quantification implique un résultat positif pour toutes les profondeurs de quantification supérieures ?

D'autre part, on peut se demander quel lien il y a entre le caractère décidable et la facilité avec laquelle on résoud la conjecture du Spectre. Dans le cas $k = 2$, la théorie FO_2 est décidable ([Mor75]), c'est-à-dire qu'il existe un algorithme qui, pour tout énoncé de profondeur de quantification 2, dit si cet énoncé admet un modèle. En revanche il existe un ensemble d'énoncés de profondeur de quantification 3 qui est indécidable ([Gol84]).

Chapitre 13

Résultats sur le cas binaire

Dans la perspective d'étudier des théories d'un langage composé d'une relation binaire et de l'égalité, nous commençons par nous intéresser au cas d'un graphe de bijection. Cet exemple est intéressant à plus d'un titre :

- tout d'abord, les structures finies qui sont modèles d'un graphe de bijection sont simples à décrire. Il s'agit en effet de réunions de cycles disjoints, chaque cycle constituant une orbite pour la bijection ;
- la théorie d'un graphe de bijection admet un modèle de cardinal n pour tout $n \geq 1$;
- la théorie d'un graphe de bijection est constituée d'une seule formule de profondeur de quantification 3 ;
- ensuite, nous le verrons au Paragraphe 13.1.5, la conjecture de Ash constante n'est pas vérifiée pour cette théorie, alors que la conjecture de Ash périodique est vérifiée (Paragraphe 13.1.2). On peut donc se servir de cette théorie comme point de départ pour essayer d'établir une frontière entre les théories pour lesquelles la conjecture de Ash constante est vérifiée, et celles pour lesquelles elle ne l'est pas (et de même pour établir une frontière entre les théories pour lesquelles la conjecture de Ash périodique est vérifiée, et celles pour lesquelles elle ne l'est pas) ;
- il est facile d'élargir ce cas à des théories un peu moins contraintes, ce que nous faisons au Paragraphe 13.2.

13.1 Un graphe de bijection et l'égalité

Dans ce paragraphe, nous commençons par donner un exemple complet avec le cas de la profondeur de quantification $k = 2$ (Paragraphe 13.1.1), puis nous montrons, à l'aide d'une méthode d'"élaguage de cycle", que pour tout $k \geq 1$, la conjecture de Ash périodique est vérifiée. Nous décrivons ensuite les classes de k-isomorphisme (Paragraphe 13.1.3), puis les cardinaux des représentants des classes de k-isomorphisme (Paragraphe 13.1.4), afin de montrer que la conjecture de Ash constante n'est pas vérifiée pour $k \geq 3$ (Paragraphe 13.1.5).

On fixe le langage $\sigma = \{F, =\}$, où F est un prédicat binaire. On fixe également la théorie d'un graphe de bijection :

$$T = \{\forall x(\exists y[F(x,y) \wedge \forall z(F(x,z) \rightarrow (z = y))]) \wedge \forall x(\exists y[F(y,x) \wedge \forall z(F(z,x) \rightarrow (z = y))])\}.$$

On appele f la fonction associée au prédicat F, c'est-à-dire telle que $y = f(x)$ si et seulement si $F(x, y)$.

On note $N_{T,k}(n)$ le nombre de classes de k-isomorphisme de modèles de T qui ont un représentant de cardinal n. On montre le résultat suivant :

Theorème 17 (Conjecture de Ash pour un graphe de bijection)
Soit $k \geq 3$. La fonction de Ash $n \mapsto N_{T,k}(n)$ est ultimement périodique, mais non ultimement constante.

13.1.1 Le cas $k = 2$

On fixe $k = 2$. On s'intéresse au comportement de la fonction de Ash. On procède de la façon suivante :
- on décrit les classes de 1-isomorphisme de singletons dans un modèle de T,
- puis on décrit les classes de 2-isomorphisme de modèles de T,
- on étudie quelles sont les cardinalités possibles pour les représentants de chaque classe,
- et enfin on calcule $N_{T,2}(n)$ pour tout n.

13.1.1.1 Classes de 1-isomorphisme de singletons dans un modèle de T

On suppose qu'on est dans un modèle fini de T, noté $\mathcal{A}$. On va décrire les sous-arbres potentiels de rang 1 d'une 2-destinée exhaustive et essentielle de $\mathcal{A}$. Soit x un nœud de rang 1 dans la 2-destinée réduite de $\mathcal{A}$. Le nœud x peut satisfaire, avec lui-même :
- soit $F(x, x)$,
- soit $\neg F(x, x)$.

Le nœud x a un fils obligatoire, de même étiquette, et des fils optionnels d'étiquettes distinctes de celle de x. Un fils optionnel y peut satisfaire les situations :
- $F(y, y)$ et $F(y, x)$ et $F(x, y)$,
- $F(y, y)$ et $F(y, x)$ et $\neg F(x, y)$,
- $F(y, y)$ et $\neg F(y, x)$ et $F(x, y)$,
- $F(y, y)$ et $\neg F(y, x)$ et $\neg F(x, y)$,
- $\neg F(y, y)$ et $F(y, x)$ et $F(x, y)$,
- $\neg F(y, y)$ et $F(y, x)$ et $\neg F(x, y)$,
- $\neg F(y, y)$ et $\neg F(y, x)$ et $F(x, y)$,
- $\neg F(y, y)$ et $\neg F(y, x)$ et $\neg F(x, y)$,

On aurait donc, a priori, 2^9 possibilités de sous-arbres de rang 1. En réalité, du fait de la restriction sémantique, certains de ces cas ne peuvent pas se produire, par exemple à cause de l'unicité de l'image et de l'antécédent. Les 12 sous-arbres pouvant apparaître (ceux qui sont cohérents avec la théorie T) sont dessinés Figure 13.1.

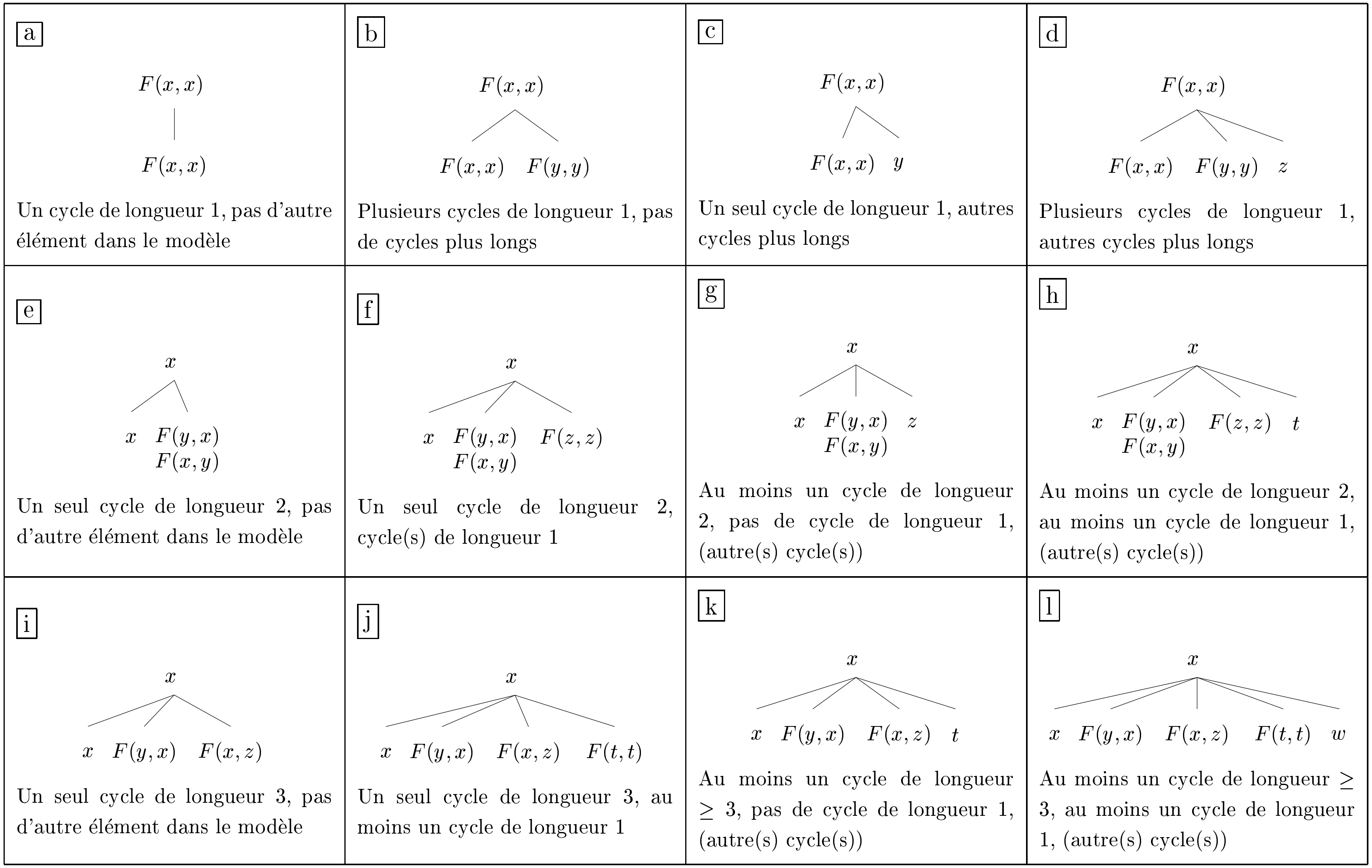


FIG. 13.1 – Sous-arbres de rang 1 potentiels dans une 2-destinée d'un modèle de T.

Le tableau précédent nous permet de mettre en évidence ce qui sert à décrire le modèle au rang 2 : le nombre de cycles et leur longueur. Il nous reste maintenant à combiner les sous-arbres de rang 1 pour décrire toutes les classes de 2-isomorphisme de modèles de T.

13.1.1.2 Classes de 2-isomorphisme de modèles de T

Pour décrire les classes de 2-isomorphisme de modèles de T, on décrit les cycles qui apparaissent dans le modèle. On voit, d'après la Figure 13.1, qu'on peut discriminer les cycles de longueur 1, 2 et ≥ 3. Une configuration du modèle sera donc une description du nombre de cycles de longueur 1, de cycles de longueur 2, et de cycles de longueur ≥ 3. Les configurations menant à des classes de 2-isomorphisme distinctes sont décrites ci-dessous :

1. <u>Cycles d'une seule sorte de longueur</u>

   Cycles de longueur 1
   - Un seul cycle de longueur 1 ($\boxed{\text{a}}$) : modèle de taille **1** ;
   - Plusieurs cycles de longueur 1 ($\boxed{\text{b}}$) : modèle de taille n **pour tout** $n \geq 2$;

   Cycles de longueur 2
   - Un seul cycle de longueur 2 ($\boxed{\text{e}}$) : modèle de taille **2** ;
   - Plusieurs cycles de longueur 2 ($\boxed{\text{g}}$) : modèle de taille **paire** pour tout nombre pair ≥ 4 ;

   Cycles de longueur ≥ 3
   - Un seul cycle de longueur 3 ($\boxed{\text{i}}$) : modèle de taille **3** ;
   - Plusieurs cycles de longueur ≥ 3 ($\boxed{\text{k}}$) : modèle de taille n **pour tout** $n \geq 6$;

2. <u>Cycles de deux sortes de longueurs</u>

   Cycles de longueurs 1 et 2
   - Un seul cycle de longueur 1, un seul cycle de longueur 2 ($\boxed{\text{c}}$ et $\boxed{\text{f}}$) : modèle de taille **3** ;
   - Un seul cycle de longueur 1, plusieurs cycles de longueur 2 ($\boxed{\text{c}}$ et $\boxed{\text{h}}$) : modèle de taille **impair** pour tout nombre impair ≥ 5 ;
   - Plusieurs cycles de longueur 1, un seul cycle de longueur 2 ($\boxed{\text{d}}$ et $\boxed{\text{f}}$) : modèle de taille n **pour tout** $n \geq 4$;
   - Plusieurs cycles de longueur 1, plusieurs cycles de longueur 2 ($\boxed{\text{d}}$ et $\boxed{\text{h}}$) : modèle de taille n **pour tout** $n \geq 6$;

   Cycles de longueurs 1 et ≥ 3
   - Un seul cycle de longueur 1, un seul cycle de longueur 3 ($\boxed{\text{c}}$ et $\boxed{\text{j}}$) : modèle de taille **4** ;
   - Un seul cycle de longueur 1, plusieurs cycles de longueur ≥ 3 ($\boxed{\text{c}}$ et $\boxed{\text{l}}$) : modèle de taille n **pour tout** $n \geq 7$;
   - Plusieurs cycles de longueur 1, un seul cycle de longueur ≥ 3 ($\boxed{\text{d}}$ et $\boxed{\text{j}}$) : modèle de taille n **pour tout** $n \geq 5$;
   - Plusieurs cycles de longueur 1, plusieurs cycles de longueur ≥ 3 ($\boxed{\text{d}}$ et $\boxed{\text{l}}$) : modèle de taille n **pour tout** $n \geq 8$;

   Cycles de longueurs 2 et ≥ 3
   - ($\boxed{\text{g}}$ et $\boxed{\text{k}}$) : modèle de taille n **pour tout** $n \geq 5$;

3. Cycles de trois sortes de longueurs

   Cycles de longueurs 1, 2 et ≥ 3
   - Un seul cycle de longueur 1 ($\boxed{\text{c}}$, $\boxed{\text{h}}$ et $\boxed{\text{l}}$) : modèle de taille n **pour tout** $n \geq 6$;
   - Plusieurs cycles de longueur 1 ($\boxed{\text{d}}$, $\boxed{\text{h}}$ et $\boxed{\text{l}}$) : modèle de taille n **pour tout** $n \geq 7$;

13.1.1.3 Cardinaux des représentants de chaque classe

On observe 17 classes de modèles pour la 2-isomorphie, mais celles dont la taille du modèle est bornée ne nous intéressent pas, étant donné que nous cherchons à déterminer le comportement asymptotique de la fonction de Ash.

Nous observons les restrictions sur la taille suivantes :
- Plusieurs cycles de longueur 2, pas d'autres cycles : modèle de taille **paire** ;
- Un seul cycle de longueur 1, plusieurs cycles de longueur 2, pas d'autres cycles : modèle de taille **impaire** ;

Pour n pair ≥ 8, on a ainsi 11 classes d'équivalence de modèle ayant un représentant de taille n, et de même pour n impair ≥ 7, mais ce ne sont pas les mêmes classes (il y en a 10 en commun, une classe qui n'apparaît que dans le cas pair, et une classe qui n'apparaît que dans le cas impair).

13.1.1.4 Conclusion

On a donc finalement

$$n \mapsto N_{T,k}(n)$$

ultimement constante de valeur 11 à partir de $n = 7$.

Le cas $k = 2$ ne rentre donc pas dans le cadre du Théorème 17, mais nous a permis de voir qu'interviennent dans le comptage des classes de modèles :
- la longueur des cycles,
- leur absence ou leur présence,
- leur nombre.

13.1.2 La conjecture de Ash périodique est vérifiée

Nous montrons à présent, sans décrire précisément les classes de k-isomorphisme, que la conjecture de Ash périodique est vérifiée. Pour cela, on montre que toute structure est k-isomorphe à une structure ne comportant que des "petits cycles", en un sens que l'on précisera. Par conséquent, si elle de cardinal "assez grand", en un sens que l'on précisera également, elle est k-isomorphe à une structure ayant des petits cycles "nombreux", structure à laquelle on peut ajouter un nombre arbitraire de "petits cycles" sans changer de classe de k-isomorphisme. Nous allons formaliser toutes ces notions. Cette méthode de découpage est inspirée la méthode employée par Durand, Fagin et Loescher dans [DFL98].

On fixe $k \geq 3$. On peut remarquer que deux éléments d'un même cycle sont nécessairement k-isomorphes. D'autre part, on commence par montrer que deux cycles de longeurs strictement supérieures à $2^{k-1}+2$ sont k-isomorphes (Proposition 21 ci-dessous). Ensuite, on montrera que toute structure est k-isomorphe à une structure dont les longueurs des cycles sont inférieures ou égales à 2^k+6.

13.1.2.1 Longueurs des cycles

> **Proposition 21 (Longueurs des cycles)**
> *Soit $\mathcal{M}$ une σ-structure, modèle de T, telle qu'il existe au moins un cycle de longueur $l \geq 2^{k-1}+3$. Alors $\mathcal{M}$ est k-isomorphe à la structure $\mathcal{M}'$ obtenue à partir de $\mathcal{M}$ en transformant l'un des cycles de longueur l en cycle de longueur $l+1$.*

Preuve : On définit le voisinage au rang k d'un élément :

> **Définition 75 (Voisinage)**
> *Soit $\mathcal{A}$ un graphe de bijection fini. Soit x un élément de $\mathcal{A}$. Le* **voisinage positif** *de x au rang k est l'ensemble :*
>
> $$V_k(x) = \{f^n(x)/n \in \{-2^{k-1}, \ldots, 2^{k-1}\}\}.$$

Le voisinage au rang k d'un élément constitue l'ensemble des éléments du cycle que l'on peut décrire avec une profondeur de quantification k.

On énumère les cycles de la structure $\mathcal{M}$: $C_0, \ldots, C_m$. On suppose que le cycle C_0 est le cycle de longueur $l \geq 2^{k-1}+3$ auquel on ajoute un élément pour obtenir la structure $\mathcal{M}'$. Celle-ci est constituée des cycles $C'_0, C'_1, \ldots, C'_m$, où C'_0 est un cycle de longueur $l+1$ et $C'_i = C_i$ pour tout $i \in \{1, \ldots, m\}$.

Pour montrer que les structures $\mathcal{M}$ et $\mathcal{M}'$ sont k-isomorphes, on décrit une stratégie gagnante pour Joueur II dans le jeu d'Ehrenfeucht en k coups entre les deux structures.

- Au premier coup : si Joueur I choisit un élément du cycle C_i dans $\mathcal{M}$, Joueur II choisit un élément du cycle C'_i dans $\mathcal{M}'$, et inversement.
- On suppose qu'on a joué $p < k$ coup, et qu'à chaque coup $i \leq p$, on a, si $(a_1, \ldots, a_i)$ et $(b_1, \ldots, b_i)$ sont les i-uples d'éléments choisis respectivement dans $\mathcal{M}$ et $\mathcal{M}'$:
  - il existe un isomorphisme local entre $(a_1, \ldots, a_i)$ et $(b_1, \ldots, b_i)$;
  - si a_i est dans une intersection de voisinages au rang $k-i+1$ des a_j pour $j < i$, alors b_i est dans la même situation vis-à-vis des voisinages au rang $k-i+1$ des b_j pour $j < i$;
  - si a_i n'est dans aucun voisinage au rang $k-i+1$ des a_j pour $j < i$, alors b_i est dans la même situation vis-à-vis des voisinages au rang $k-i+1$ des b_j pour $j < i$.

Remarquons que si $y \notin V_k(x)$ alors $V_{k-1}(x) \cap V_{k-1}(x) = \emptyset$. On veut jouer le $(p+1)$ième coup, et propager les trois conditions ci-dessus.

- Si Joueur I choisit un élément d'un des cycles C_i ou C'_i pour $i \geq 1$, alors Joueur II choisit le même élément dans l'autre structure. Les trois conditions restent vraies.
- Si Joueur I choisit un élément a_{p+1} de C_0 dans $\mathcal{M}$. Alors on a deux situations possibles :
  - soit a_{p+1} est dans le voisinage au rang $k-p$ d'un ou plusieurs des a_i pour $i \leq p$. Alors Joueur II choisit le b_{p+1} qui est au même endroit dans les voisinages au rang $k-p$ des b_i pour $i \leq p$. Ceci est possible grâce aux trois conditions, qui assurent que les intersections sont les mêmes dans les deux structures. Les trois conditions restent vraies.
  - soit a_{p+1} n'est dans aucun voisinage au rang $k-p$ des a_i pour $i \leq p$. Il faut qu'il reste sur C'_0 au moins un élément qui n'est dans aucun voisinage des b_i pour $i \leq p$. C'est le cas, car les voisinages des b_i pour $i \leq p$ recouvrent au maximum $p(2^{k-p}+1)$ éléments, et $p(2^{k-p}+1) \leq 2^{k-1}+3 \leq l$ pour tout $p \leq k$. Joueur II choisit un élément b_{p+1} qui n'est dans aucun voisinage des b_i pour $i \leq p$. Les trois conditions restent vraies.
- Si Joueur I choisit un élément b_{p+1} de C'_0 dans $\mathcal{M}$, on procède comme ci-dessu en échangeant les rôles de $\mathcal{M}$ et $\mathcal{M}'$.

Les deux structures $\mathcal{M}$ et $\mathcal{M}'$ sont k-isomorphes.

□

13.1.2.2 Découpage des cycles

Proposition 22 (Découpage des cycles)
Soit $\mathcal{A}$ un graphe de bijection fini. Alors $\mathcal{A}$ est k-isomorphe à un graphe de bijection fini $\mathcal{B}$ qui ne comporte que des cycles de longueur inférieure ou égale à 2^k+6.

Preuve : On procède par découpage des grands cycles de $\mathcal{A}$. Si $\mathcal{A}$ ne comporte que des cycles de longueur inférieure ou égale à 2^k+6, la proposition est évidente. Sinon, $\mathcal{A}$ comporte $r \geq 1$ cycles, notés $C_1, \ldots, C_r$, qui sont de longueur strictement supérieure à 2^k+6. On découpe chacun de ces cycles de la façon suivante : sur le cycle C_i, il y a au moins deux éléments a_i et b_i qui sont séparés par $2^{k-1}+2$ éléments, d'un côté comme de l'autre du cycle. On intervertit les antécédents de a_i et b_i, comme sur la Figure 13.2, pour obtenir deux cycles D_{a_i} et D_{b_i}, tous deux de longueur supérieure ou égale à $2^{k-1}+3$.

On définit la structure $\mathcal{B}$ comme étant la structure de même domaine que $\mathcal{A}$ mais dans laquelle tous les cycles $C_1, \ldots, C_r$ ont subi un ou plusieurs découpages, jusqu'à ce qu'ils soient découpés en cycles de longueurs inférieures à 2^k+6. Il est évident que $\mathcal{B}$ est modèle de T, et qu'elle ne comporte que des cycles de longueur inférieure ou égale à 2^k+6. Il reste à

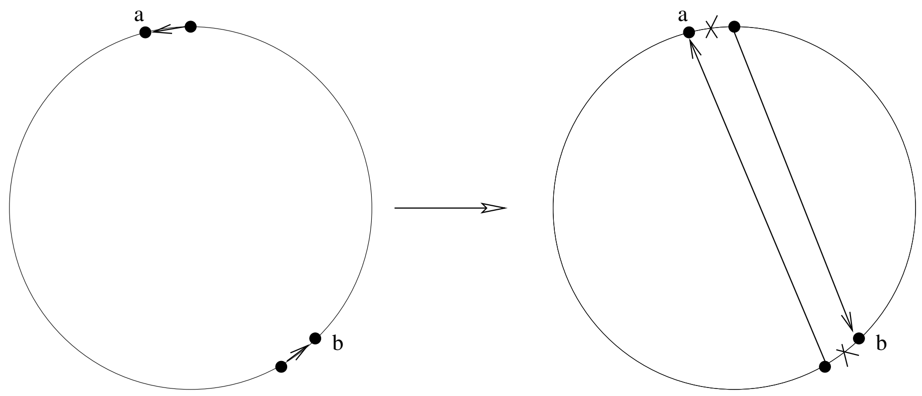


FIG. 13.2 – Découpage d'un cycle long en deux cycles courts

démontrer que les deux structures $\mathcal{A}$ et $\mathcal{B}$ sont k-isomorphes. Il suffit pour cela de montrer qu'un cycle C_i est k-isomorphe à la réunion des deux cycles issus du découpage, D_{a_i} et D_{b_i}. On adapte la stratégie gagnante utilisée précédemment, en mettant en correspondance les voisinages de a_i dans C_i avec le cycle D_{a_i}, et les voisinages de b_i dans C_i avec le cycle D_{b_i}.

□

13.1.2.3 Conclusion sur la périodicité de la fonction de comptage

On dispose maintenant de tous les éléments pour montrer le résultat suivant :

Theorème 18 (Ash périodique)
Pour tout $k \geq 3$, la fonction de Ash :

$$n \mapsto N_{T,k}(n)$$

est asymptotiquement périodique.

Preuve : Soit $\mathcal{C}$ une classe de k-isomorphisme modèle de T, telle que $\mathcal{C}$ a un représentant $\mathcal{A}$ de cardinal $n > k \times (2^k + 7)(2^{k-1} + 3)$. D'après la Proposition 22, la structure $\mathcal{A}$ est k-isomorphe à une structure $\mathcal{B}$ de même cardinal que $\mathcal{A}$, et ne comportant que des cycles de longueurs inférieures ou égales à $2^k + 6$. Comme le cardinal de $\mathcal{B}$ est $n > k \times (2^k+7)(2^{k-1}+3)$, cela signifie qu'il y a au moins une longueur de cycle $l \leq 2^k+6$ pour laquelle la structure $\mathcal{B}$ contient au moins k cycles de longueur l. Alors $\mathcal{B}$ est k-isomorphe aux structures $\mathcal{B}_s$ de cardinaux respectifs $n+s\times l$ construites en ajoutant à $\mathcal{B}$ un nombre s de cycles de longueur l. Cela signifie que la classe $\mathcal{C}$ est comptée pour tous les cardinaux de la forme $n + s \times l$.

Si l'on réalise la même opération pour tous les représentants de $\mathcal{C}$ qui ont un cardinal strictement supérieur à $k\times(2^k+7)(2^{k-1}+3)$, alors il existe un certain nombre de longueurs $l_1, \ldots, l_r$ toutes inférieures ou égales à $2^k + 6$, et un certain nombre de cardinaux initiaux

$n_1, \ldots, n_r$ tels que l'ensemble des cardinaux des représentants de $\mathcal{C}$ qui sont strictement supérieurs à $k(2^k+7)(2^{k-1}+3)$ est :

$$\bigcup_{i=1}^{r}(n_i + l_i\mathbb{N}).$$

Autrement dit, la classe $\mathcal{C}$ est comptée périodiquement dans $N_{T,k}(n)$ avec une période $\mathrm{ppcm}(l_i)_{i=1\ldots r}$ qui divise $\mathrm{ppcm}(i)_{i\leq 2^k+6}$.

Comme c'est le cas de chaque classe ayant des représentants de cardinal strictement supérieur à $k(2^k+7)(2^{k-1}+3)$, on en déduit que $n \mapsto N_{T,k}(n)$ est périodique de période divisant $\mathrm{ppcm}(i)_{i\leq 2^k+6}$.

□

13.1.3 Description des classes de k-isomorphisme

13.1.3.1 Notion de k-configuration

Afin de montrer que la conjecture de Ash constante n'est pas vérifiée par la théorie T, nous allons devoir caractériser de manière très précise les classes de k-isomorphisme de modèles finis de T. Nous avons vu, grâce à la Proposition 21, que les cycles de longueur $l \geq 2^{k-1}+3$ sont k-isomorphes. Un modèle fini de T peut donc être décrit, à k-isomorphisme près, par la donnée du nombre de cycles de longueur 1, du nombre de cycles de longueur 2, ..., du nombre de cycles de longueur 2^{k-1}, du nombre de cycles de longueure $2^{k-1}+1$, du nombre de cycles de longueure $2^{k-1}+2$ et du nombre de cycles de longueur supérieure ou égale à $2^{k-1}+3$. On s'appuie alors sur la notion de *k-configuration* :

Définition 76 (k-configuration)
Une k-configuration est la donnée d'un $(2^{k-1}+3)$-uple d'entiers $(n_1, \ldots, n_{2^{k-1}+3})$.

A chaque modèle de T est associée une k-configuration.

Définition 77 (Petit cycle)
Soit $\mathcal{M}$ un modèle de T. Les **petits cycles** *au rang k de $\mathcal{M}$ sont les cycles de longueur $l \leq 2^{k-1}$.*

13.1.3.2 Définition et caractérisation au rang k des petits cycles

Les petits cycles sont caractérisables par des énoncés de profondeur de quantification $\leq k$.

Proposition 23 (Définition des petits cycles)

1. *Pour tout* $n \geq 0$*, il existe une formule* $F_n(x,y)$ *à deux paramètres, de profondeur de quantification* $\lceil \log_2 n \rceil$ *qui exprime que* $y = f^n(x)$*.*
2. *Pour tout* $l \geq 1$*, il existe une formule* $C_l(x)$ *à un paramètre, de profondeur de quantification* $\lceil \log_2 l \rceil$ *qui exprime que* x *est dans un cycle de longueur* l*.*
3. *Pour tout* $i \geq 0$*, il existe une formule* $V_i(x,y)$ *à deux paramètres, de profondeur de quantification* $i-1$ *qui exprime que* y *est dans le voisinage au rang* i *de* x*.*

Preuve :

1. On procède par récurrence sur $\lceil \log_2 n \rceil$. On définit :

$$F_0(x,y) := (x = y)$$
$$F_1(x,y) := F(x,y)$$

   On suppose que les formules $F_i(x,y)$ sont définies pour $\lceil \log_2 i \rceil \leq \lceil \log_2 n \rceil$, on veut les définir pour $j \in \{2^{\lceil \log_2 n \rceil} + 1, \ldots, 2^{\lceil \log_2 n \rceil + 1}\}$. Pour un tel j, on a $j = 2^{\lceil \log_2 n \rceil} + r$, avec $0 < r \leq 2^{\lceil \log_2 n \rceil}$. On définit :

$$F_j(x,y) := \exists z (F_{2^{\lceil \log_2 n \rceil}}(x,z) \wedge F_r(z,y)).$$

   Cette formule est de profondeur de quantification égale au maximum des profondeurs de quantification de $F_{2^{\lceil \log_2 n \rceil}}$ et F_r plus un, c'est-à-dire $\lceil \log_2 n \rceil + 1 = \lceil \log_2 j \rceil$. Elle exprime que $y = f^r(f^{2^{\lceil \log_2 n \rceil}}(x) = f^j(x)$. C'est bien ce qu'on voulait.
2. On pose $C_l(x) := F_l(x,x)$.
3. On pose

$$V_i(x,y) := F_0(x,y) \vee \bigvee_{j=1}^{2^{i-1}} F_j(x,y) \vee \bigvee_{j=1}^{2^{i-1}} F_j(y,x).$$

□

On distingue maintenant trois sortes de cycles :
- les cycles de longueur $l \leq 2^{k-1}$ (les petits cycles) ;
- les cycles de longueur $l \geq 2^{k-1} + 3$ (les grands cycles) ;
- les cycles de longueur $2^{k-1}+1$ et $2^{k-1}+2$, qui ne sont ni grands, ni petits. Appelons-les cycles moyens.

13.1.3.3 Le cas des cycles moyens

Analysons en détail le cas des cycles moyens. Ces cycles sont discernables des grands cycles, mais seulement dans certaines structures.

Un cycle de longueur $2^{k-1}+1$

Un cycle de longueur $2^{k-1}+1$ est definissable à profondeur de quantification k si la structure ne comporte qu'un seul cycle de longueur $> 2^{k-2}$. En effet, si c'est le cas, la formule suivante :

$$G_{2^{k-1}+1}(x) := \neg C_1(x) \wedge \ldots \wedge \neg C_{2^{k-1}}(x) \wedge \forall y(C_1(y) \vee \ldots \vee C_{2^{k-2}}(y) \vee V_{k-1}(x,y)),$$

de profondeur de quantification $k-1$, exprime :
- que le cycle de x est de longueur supérieure ou égale à $2^{k-1}+1$,
- que tous les éléments de la structure sont soit dans des cycles de longueur $\leq 2^{k-2}$, soit dans le voisinage au rang $k-1$ de x. Cela implique que le cycle de x est de longueur exactement $2^{k-1}+1$ et qu'il est le seul cycle de longueur $> 2^{k-2}$.

Un cycle de longueur $2^{k-1}+2$

Un cycle de longueur $2^{k-1}+2$ est definissable à profondeur de quantification k si la structure ne comporte qu'un seul cycle de longueur $> 2^{k-3}$. En effet, si c'est le cas, la formule suivante :

$$\begin{aligned} G_{2^{k-1}+2}(x) := &\neg C_1(x) \wedge \ldots \wedge \neg C_{2^{k-1}}(x) \wedge \exists y(\neg C_1(y) \wedge \ldots \wedge \neg C_{2^{k-2}}(y) \wedge \neg V_{k-1}(x,y) \\ &\wedge \forall z[C_1(z) \vee \ldots C_{2^{k-3}}(z) \vee V_{k-2}(x,z) \vee V_{k-2}(y,z)]), \end{aligned}$$

de profondeur de quantification $k-1$, exprime :
- que le cycle de x est de longueur supérieure ou égale à $2^{k-1}+1$,
- que le cycle de y est de longueur supérieure ou égale à $2^{k-2}+1$,
- que tous les éléments de la structure sont soit dans des cycles de longueur $\leq 2^{k-3}$, soit dans le voisinage au rang $k-2$ de x, soit dans le voisinage au rang $k-2$ de y. Cela implique que y est sur le cycle de x (sinon, étant donnée la longueur du cycle de x, il y aurait des éléments sur ce cycle hors de $V_{k-2}(x)$, et hors de $V_{k-2}(y)$ qui serait sur un autre cycle). Comme y n'est pas dans le voisinage au rang $k-1$ de x, cela signifie que le cycle de x est de longueur $2^{k-1}+2$. Comme les deux voisinages au rang $k-2$ de x et y suffisent à recouvrir tout ce cycle, cela signifie que le cycle de x est exactement de longueur $2^{k-1}+2$. De plus, c'est le seul cycle à être de longueur $> 2^{k-3}$.

En dehors de ces conditions très particulières, ces cycles moyens deviennent grands, c'est ce qu'exprime la proposition ci-dessous.

Proposition 24 (Quand les cycles moyens deviennent grands)

1. *Soit $\mathcal{M}$ une structure qui contient un cycle de longueur $2^{k-1}+1$ et au moins un autre cycle de longueur $l > 2^{k-2}$. Alors $\mathcal{M}$ est isomorphe à la structure $\mathcal{M}'$ obtenue à partir de $\mathcal{M}$ en transformant un cycle de longueur $2^{k-1}+1$ en cycle de longueur $2^{k-1}+3$.*
2. *Soit $\mathcal{M}$ une structure qui contient un cycle de longueur $2^{k-1}+2$ et au moins un autre cycle de longueur $l > 2^{k-3}$. Alors $\mathcal{M}$ est isomorphe à la structure $\mathcal{M}'$ obtenue à partir de $\mathcal{M}$ en transformant un cycle de longueur $2^{k-1}+2$ en cycle de longueur $2^{k-1}+3$.*

Preuve : On énumère les cycles de $\mathcal{M}$: $C_0, \dots, C_m$, et on suppose que c'est le cycle C_0 qui est modifié pour donner la structure $\mathcal{M}'$, contenant les cycles C'_i correspondants aux cycles C_i. On suppose de plus que le cycle C_1 est de longueur $> 2^{k-2}$ dans le cas (1.), et de longueur $> 2^{k-3}$ dans le cas (2.). On exécute la même stratégie gagnante que pour la Proposition 21, en changeant seulement la façon de jouer le $(p+1)$ième coup :

- Si Joueur I choisit un élément dans un cycle C_i pour $i = 2, \dots, m$, alors Joueur II choisit le même élément dans C'_i. Les trois conditions sont encore vérifiées.
- Si Joueur I choisit un élément a_{p+1} dans le cycle C_0 de $\mathcal{M}$, alors Joueur II peut choisir un élément b_{p+1} dans le cycle C'_0 de longueur $2^{k-1}+3$ de $\mathcal{M}'$. Ceci est possible car pour tout $p \leq k$, on a $p(2^{k-p}+1) < 2^{k-1}+3$. Les trois conditions sont encore vraies.
- Si Joueur I choisit un élément b_{p+1} dans le cycle C_0, tel que b_{p+1} est dans une intersection de voisinages au rang $k-p$ des b_i pour $i = 1, \dots, p$, alors Joueur II choisit a_{p+1} dans l'intersection de voisinages au rang $k-p$ qui correspond dans C_0.
- Si Joueur I choisit un élément b_{p+1} dans le cycle C_0, tel que b_{p+1} n'est dans aucun voisinage au rang $k-p$ des b_i pour $i = 1, \dots, p$, montrons que Joueur II peut choisir un a_{p+1} qui propage les conditions.
  1. On suppose qu'on est dans le cas (1.) (c'est-à-dire que C_0 est de longueur $2^{k-1}+1$).
     - Si les voisinages au rang $k-p$ des a_i recouvrent le cycle de longueur $2^{k-1}+1$, cela signifie que $p = 1$ et c'est donc pour jouer le deuxième coup qu'il se pose éventuellement un problème. On a joué un seul coup, et on a a_1 choisi dans C_0, et b_1 choisi dans C'_0. Joueur Ia choisi b_2 dans C'_0, en dehors du voisinage au rang $k-1$ de b_1. Alors Joueur II choisit un élément a_2 de C_1. On poursuit alors le jeu normalement en utilisant au choix C_1 ou C_0 lorsque Joueur I choisit un élément de C'_1 ou C'_0. C'est toujours possible, car pour tout $p \in \{2, \dots, k\}$, on a $p(2^{k-1}+1) \leq 2^{k-2}+2^{k-1}+1$, donc strictement inférieur à la longueur de C_1 plus la longueur de C_0.
     - Sinon, Joueur II choisit un élément en dehors des voisinages au rang $k-p$ des a_i, pour $i = 1, \dots$.
  2. On suppose qu'on est dans le cas (2.) (c'est-à-dire que C_0 est de longueur $2^{k-1}+2$).

- Si les voisinages au rang $k-p$ des a_i recouvrent le cycle de longueur $2^{k-1}+1$, cela signifie que $p=2$ et c'est donc pour jouer le troisième coup qu'il se pose éventuellement un problème. On a joué deux coups, et on a (a_1, a_2) choisis dans C_0, et (b_1, b_2) choisis dans C'_0. Joueur Ia choisi b_3 dans C'_0, en dehors des voisinages au rang $k-2$ de b_1 et b_2. Alors Joueur II choisit un élément a_3 de C_1. On poursuit alors le jeu normalement en utilisant au choix C_1 ou C_0 lorsque Joueur I choisit un élément de C'_1 ou C'_0. C'est toujours possible, car pour tout $p \in \{3, \ldots, k\}$, on a $p(2^{k-1}+1) \leq 2^{k-3}+2^{k-1}+1$, donc strictement inférieur à la longueur de C_1 plus la longueur de C_0.
- Sinon, Joueur II choisit un élément en dehors des voisinages au rang $k-p$ des a_i, pour $i = 1, \ldots p$.

- Si Joueur I choisit un élément de C_1, ou bien un élément de C'_1 et à aucun moment dans le jeu les voisinages des a_i n'ont recouvert C_0, alors Joueur II choisit le même élément dans l'autre structure.
- Si Joueur I choisit un élément de C'_1 et qu'on a eu recours à C_1 pour représenter l'analogue d'un élément de C'_0, alors :
  - si b_{p+1} est dans une intersection de voisinages des b_i pour $i = 1, \ldots, p$, alors Joueur II choisit l'élément correspondant dans l'intersection correspondante dans $\mathcal{M}$,
  - si b_{p+1} n'est dans aucune intersection de voisinages des b_i pour $i = 1, \ldots, p$, alors Joueur II choisit dans C_1 ou C_0 un élément qui n'est dans aucun voisinage des a_i pour $i = 1, \ldots, p$. C'est possible d'après ce qui précède.

  Dans tous les cas, les trois conditions restent vérifiées.

□

13.1.3.4 Equivalence entre configurations

On caractérise les classes de k-isomorphisme grâce à des conditions satisfaites par les k-configurations de leurs représentants.

Définition 78 (Relation d'équivalence sur les k-configurations)
Soient $C = (n_1, \ldots, n_{2^{k-1}+3})$ et $D = (m_1 \ldots, m_{2^{k-1}+3})$ deux k-configurations. On dit que ces deux k-configurations C et D sont **équivalentes***, et on note $C \simeq D$ si les quatres conditions suivantes sont vérifiées :*

1. *On a :*
$$\sum_{j=2^{k-1}+1}^{2^{k-1}+3} n_j = 0 \Leftrightarrow \sum_{j=2^{k-1}+1}^{2^{k-1}+3} m_j = 0.$$
2. *Pour tout $l \leq 2^{k-1}$, on a :*
$$n_l \neq m_l \Rightarrow (n_l \geq k - \lceil \log_2 l \rceil \text{ et } m_l \geq k - \lceil \log_2 l \rceil).$$
3. *Pour $l = 2^{i-1} + 1$, pour $i \in \{1, \ldots, k-1\}$, on a :*
si
$$(n_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} n_j \text{ ou } m_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j)$$
alors
$$n_l \neq m_l \Rightarrow (n_l > k - \lceil \log_2 l \rceil + 1 \text{ et } m_l > k - \lceil \log_2 l \rceil + 1).$$
4. *Pour tout $i \in \{1, \ldots, k-1\}$, on a :*
$$\sum_{j=2^{i-1}+1}^{2^i} n_j \neq \sum_{j=2^{i-1}+1}^{2^i} m_j \Rightarrow \sum_{j=2^{i-1}+1}^{2^{k-1}+3} n_j > k - i \text{ et } \sum_{j=2^{i-1}+1}^{2^{k-1}+3} m_j > k - i.$$
5. *Pour $l = 2^{k-1} + \alpha$, avec $\alpha \in \{1, 2\}$, on a :*
$$(n_l = \sum_{j=2^{k-\alpha-1}+1}^{2^{k-1}+3} n_j = 1 \Leftrightarrow m_l = \sum_{j=2^{k-\alpha-1}+1}^{2^{k-1}+3} m_j = 1).$$

Remarque 24 *Deux structures qui ont une même k-configuration sont k-isomorphes d'après la Proposition 21, mais deux structures k-isomorphes n'ont pas forcément la même k-configuration. C'est pourquoi on introduit une notion d'équivalence entre les k-configurations afin de caractériser les classes de k-isomorphisme via les k-configurations.*

Proposition 25 (Caractérisation des classes de k-isomorphisme)
Deux structures $\mathcal{A}$ et $\mathcal{B}$ modèles de T sont k-isomorphes si et seulement si les k-configurations $C_{\mathcal{A}}$ et $C_{\mathcal{B}}$ qui leurs sont associées sont équivalentes.

Preuve :

$\boxed{\Rightarrow}$: on suppose que les deux configurations C_1 et C_2 ne sont pas équivalentes, et on montre que les deux structures ne sont pas k-isomorphes. On procède de la façon suivante : pour chaque cas dans lequel les deux configurations ne sont pas équivalentes (on nie une des conditions), on exhibe un énoncé de profondeur de quantification inférieure ou égale à k qui est vrai dans l'une des structures mais faux dans l'autre.

1. On suppose que la première condition n'est pas vérifiée. On peut supposer sans perte de généralité que
$$\sum_{j=2^{k-1}+1}^{2^{k-1}+3} n_j = 0,$$
mais
$$\sum_{j=2^{k-1}+1}^{2^{k-1}+3} m_j \neq 0.$$
Alors, l'énoncé suivant :
$$\forall x[C_1(x) \vee \ldots \vee C_{2^{k-1}}(x)]$$
dit que tous les éléments de la structure sont dans des cycles de longueur $\leq 2^{k-1}$. Il est de profondeur de quantification k, et il est vrai dans $\mathcal{M}_1$ mais pas dans $\mathcal{M}_2$.
2. On suppose que la deuxième condition n'est pas vérifiée. On suppose qu'il existe un $l \leq 2^{k-1}$ tel que :
   - $n_l < k - \lceil \log_2 l \rceil$ ou $m_l < k - \lceil \log_2 l \rceil$,
   - mais $n_l \neq m_l$.

   On peut supposer sans perte de généralité que $n_l < k - \lceil \log_2 l \rceil$. On dispose d'un énoncé θ_{n_l} qui dit qu'il y a exactement n_l cycles de longueur l :
$$\begin{aligned}\theta_{n_l} :=& \exists x_1[C_l(x_1) \wedge \exists x_2(C_l(x_2) \wedge \neg V_{\lceil \log_2 l \rceil -1}(x_1, x_2) \wedge \ldots \wedge \exists x_{n_l}(C_l(x_{n_l}) \wedge \\ & \neg V_{\lceil \log_2 l \rceil -1}(x_1, x_{n_l}) \wedge \ldots \wedge \neg V_{\lceil \log_2 l \rceil -1}(x_{n_l-1}, x_{n_l}) \wedge \forall z[C_l(z) \rightarrow V_{\lceil \log_2 l \rceil -1}(x_1, z) \\ & \vee \ldots \vee V_{\lceil \log_2 l \rceil -1}(x_{n_l}, z)]) \ldots)].\end{aligned}$$

   Cet énoncé est de profondeur de quantification $n_l + 1 + \lceil \log_2 l \rceil \leq k$. Il est vrai dans $\mathcal{M}_1$ mais pas dans $\mathcal{M}_2$.
3. On suppose que la troisième condition n'est pas vérifiée. On suppose que $l = 2^{i-1}+1$, pour un $i \in \{1, \ldots, k-1\}$. On suppose que :
$$(n_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} n_j \text{ ou } m_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j)$$
mais
$$n_l \neq m_l \text{ et } (n_l \leq k - \lceil \log_2 l \rceil + 1 \text{ ou } m_l \leq k - \lceil \log_2 l \rceil + 1).$$
Sans perte de généralité, on peut supposer que
$$n_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} n_j.$$

– Si on a $n_l \neq m_l$ et $n_l \leq k - \lceil \log_2 l \rceil + 1$: alors l'énoncé

$$\exists x_1[C_l(x_1) \wedge \ldots \wedge \exists x_p((\neg C_1(x_p) \wedge \ldots \wedge \neg C_{2^{i-2}}(x_p)) \wedge \neg V_{i-1}(x_1, x_p) \wedge \ldots \wedge \neg V_{i-1}(x_{p-1}, x_p) \wedge \exists z[\neg C_1(z) \wedge \ldots \wedge \neg C_{2^{i-2}}(z) \wedge \neg V_{i-1}(x_1, z) \wedge \ldots \wedge \neg V_{i-1}(x_p, z)])) \ldots]$$

exprime qu'il y a strictement plus de p cycles de longueur l. Elle est de profondeur de quantification $p + \lceil \log_2(l-1) \rceil = p + \lceil \log_2 l \rceil - 1 = k$, pour $p = k - i + 1$, cet énoncé est vrai dans $\mathcal{M}_2$, pas dans $\mathcal{M}_1$.

– Si on a $n_l \neq m_l$ et $m_l \leq k - \lceil \log_2 l \rceil + 1$:
  – si un des m_j pour $j > 2^{k-1}$ est non nul, alors la condition (1.) n'est pas respectée, et on conclut comme en 1.,
  – sinon, si un des m_j pour $j \leq 2^{k-1}$ et $j \neq l$ est non nul, alors la condition (2.) n'est pas respectée, et on conclut comme en 2.,
  – sinon, c'est qu'on a

$$m_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j,$$

et on peut conclure comme ci-dessus, en échangeant les rôles des deux structures.

4. On suppose que la quatrième condition n'est pas vérifiée. On suppose qu'il existe un $i \in \{1, \ldots k-1\}$ tel que :

$$\sum_{j=2^{i-1}+1}^{2^i} n_j \neq \sum_{j=2^{i-1}+1}^{2^i} m_j,$$

mais

$$\sum_{j=2^{i-1}+1}^{2^{k-1}+3} n_j \leq k - i \text{ ou } \sum_{j=2^{i-1}+1}^{2^{k-1}+3} m_j \leq k - i.$$

On peut supposer sans perte de généralité que

$$\sum_{j=2^{i-1}+1}^{2^{k-1}+3} n_j \leq k - i.$$

– Si on a :

$$\sum_{j=2^{i-1}+1}^{2^i} n_j < k - i,$$

alors tous les n_j pour $j \in \{2^{i-1}+1, \ldots, 2^i\}$ sont $< k - \lceil \log_2 j \rceil$. Si l'un des m_j correspondants est distinct de n_j, la deuxième condition est mise en défaut et on conclut comme en 2., sinon, cela implique que

$$\sum_{j=2^{i-1}+1}^{2^i} n_j = \sum_{j=2^{i-1}+1}^{2^i} m_j,$$

ce qui est impossible d'après l'hypothèse de départ.

– Si on a :
$$\sum_{j=2^{i-1}+1}^{2^i} m_j < k - i,$$
alors tous les n_j pour $j \in \{2^{i-1}+1, \ldots, 2^i\}$ sont $< k - \lceil \log_2 j \rceil$. Si l'un des n_j correspondants est distinct de m_j, la deuxième condition est mise en défaut et on conclut comme en 2., sinon, cela implique que
$$\sum_{j=2^{i-1}+1}^{2^i} n_j = \sum_{j=2^{i-1}+1}^{2^i} m_j,$$
ce qui est impossible d'après l'hypothèse de départ.
– Sinon, on a :
$$\sum_{j=2^{i-1}+1}^{2^i} n_j = k - i \text{ et } \sum_{j=2^{i-1}+1}^{2^i} n_j > k - i,$$
et on considère l'énoncé
$$\begin{aligned}\Theta_p :=& \exists x_1[(C_{2^{i-1}+1}(x_1) \vee \ldots \vee C_{2^i}(x_1)) \wedge \exists x_2[(C_{2^{i-1}+1}(x_2) \vee \ldots \vee C_{2^i}(x_2)) \wedge \\ & \neg V_i(x_1, x_2) \wedge \ldots \wedge \exists x_p[(C_{2^{i-1}+1}(x_p) \vee \ldots \vee C_{2^i}(x_p)) \wedge \neg V_i(x_1, x_p) \wedge \ldots \wedge \\ & \neg V_i(x_{p-1}, x_p) \wedge \exists z[\neg C_1(z) \wedge \ldots \wedge \neg C_{2^{i-1}}(z) \wedge \neg V_i(x_1, z) \wedge \ldots \wedge \neg V_i(x_p, z)]]]].\end{aligned}$$

Cet énoncé exprime qu'il y a au moins p cycles de longueur dans $\{2^{i-1}+1, \ldots, 2^i\}$, et au moins un autre cycle de longueur $\geq 2^{i-1}+1$. Pour $p = k - i$, cet énoncé est de profondeur de quantification k, et il est vrai dans $\mathcal{M}_2$ mais pas dans $\mathcal{M}_1$.

5. On suppose que la cinquieme condition n'est pas vérifiée. On suppose que $l = 2^{k-1} + \alpha$, avec $\alpha \in \{1, 2\}$, et que :
$$n_l = \sum_{j=2^{k-\alpha-1}+1}^{2^{k-1}+3} n_j = 1,$$
mais
$$m_l = \sum_{j=2^{k-\alpha-1}+1}^{2^{k-1}+3} m_j \neq 1 \text{ ou } m_l \neq \sum_{j=2^{k-\alpha-1}+1}^{2^{k-1}+3} m_j.$$
Dans les deux cas, l'énoncé $\exists x G_{2^{k-1}+\alpha}$ est vrai dans $\mathcal{M}_1$ mais pas dans $\mathcal{M}_2$.

$\boxed{\Leftarrow}$: On suppose maintenant que les configurations C_1 et C_2 sont équivalentes, et on montre que les deux structures sont k-isomorphes en exhibant une stratégie gagnante pour Joueur II dans le jeu d'Ehrenfeucht en k coups entre les deux structures. La stratégie est la même que dans les preuves précédentes, à savoir que si Joueur I joue l'élément a_{p+1} dans une intersection de voisinages au rang $k - p$ des a_i déjà choisis, Joueur II choisit l'élément b_{p+1} dans l'intersection correspondant dans l'autre structure, et si Joueur I joue l'élément a_{p+1} en dehors de tous les voisinages des a_i déjà joués, alors Joueur II essaie d'en faire autant, et de placer b_{p+1} dans un cycle de même longueur que a_{p+1} si ce cycle est "petit au rang $k-p$". Montrons que nous pouvons toujours effectuer ce jeu. On suppose qu'on a déjà joué p coups, et que Joueur I choisit un élément a_{p+1} sur un cycle de longueur l de $\mathcal{M}_1$, en dehors de tous les voisinages au rang $k-p$ des a_i pour $i = 1 \dots p$.

- Si $l \geq 2^{k-1}+3$, alors la Condition 1 impose qu'il y ait un cycle de longueur $\geq 2^{k-1}+1$ dans $\mathcal{M}_2$. D'après la Condition 5, on ne peut avoir un seul cycle de longueur $2^{k-1}+1$ ou un seul cycle de longueur $2^{k-1}+2$ dans $\mathcal{M}_2$. Comme on a recouvert au plus $p(2^{k-p}+1)$ éléments au cours des précédents p coups, on a au moins un élément dans l'un de ces cycles qui n'est pas recouvert. Joueur II choisit un tel élément.
- Si $l = 2^{k-1}+2$, on a deux possibilités.
  - Si c'est le seul cycle de $\mathcal{M}_1$ de longueur $\geq 2^{k-3}+1$, alors c'est aussi le cas dans $\mathcal{M}_2$, et on ne l'a pas recouvert avant : on ne peut pas l'avoir mis en correspondance avec un cycle plus petit, qui est encore totalement définissable jusqu'à $p = 2$, et après $p = 2$, les voisinages ne peuvent plus recouvrir le cycle de longueur $2^{k-1}+2$. Il reste donc au moins un élément non recouvert sur le cycle de $\mathcal{M}_2$. Joueur II choisit cet élément.
  - Sinon, il y a dans $\mathcal{M}_2$ au moins un cycle de longueur $\geq 2^{k-3}+1$, dont au moins un cycle de longueur $\geq 2^{k-1}+1$ d'après la Condition 1. Si dans $\mathcal{M}_2$, il n'y a qu'un seul cycle de longueur $2^{k-1}+1$, alors la Condition 5 impose qu'il y ait au moins un autre cycle de longueur $\geq 2^{k-2}+1$. Donc si $p = 1$ et qu'on a recouvert le cycle de longueur $2^{k-1}+1$ au coup 1, Joueur II peut choisir un élément d'un cycle de longueur $\geq 2^{k-2}+1$. Si $p = 2$ et qu'on a recouvert un cycle de longueur $2^{k-1}+2$ dans $\mathcal{M}_2$, Joueur II peut choisir un élément d'un cycle de longueur $\geq 2^{k-3}+1$. Si $p > 2$, on ne peut plus recouvrir les cycles moyens, donc on prend un élément non recouvert d'un cycle moyen ou grand de $\mathcal{M}_2$.
- Si $l = 2^{k-1}+1$, on a deux possibilités.
  - Si c'est le seul cycle de $\mathcal{M}_1$ de longueur $\geq 2^{k-2}+1$, alors on est dans la même situation dans $\mathcal{M}_2$. Joueur II peut alors jouer, on n'a pas recouvert le cycle de $\mathcal{M}_2$ au coup p.
  - Sinon, il y a dans $\mathcal{M}_2$ au moins un cycle de longueur $\geq 2^{k-2}+1$, dont au moins un cycle de longueur $\geq 2^{k-1}+1$ d'après la Condition 1. Si dans $\mathcal{M}_2$, il n'y a qu'un seul cycle de longueur $2^{k-1}+1$, alors la Condition 3 impose qu'il y ait au moins un autre cycle de longueur $\geq 2^{k-2}+1$. Donc si $p = 1$ et qu'on a recouvert le cycle de longueur $2^{k-1}+1$ au coup 1, Joueur II peut choisir un élément d'un cycle de longueur $\geq 2^{k-2}+1$. Si $p = 2$ et qu'on a recouvert un cycle de longueur $2^{k-2}+2$ dans $\mathcal{M}_2$, Joueur II peut choisir un élément d'un cycle de longueur $\geq 2^{k-2}+1$. Si $p > 2$, on ne peut plus recouvrir les cycles moyens, donc on prend un élément non recouvert d'un cycle moyen ou grand de $\mathcal{M}_2$.
- Sinon, $l \in \{1, \dots 2^{k-1}\}$. D'après la Condition 2, il y a au moins un cycle de longueur l dans $\mathcal{M}_2$. On a joué p coups. La question est donc de savoir si on a recouvert tous les cycles de longueur l de $\mathcal{M}_2$, et si c'est le cas, peut-on trouver un cycle de longueur $\geq 2^{\lceil \log_2 l \rceil - 1}+1$ qui a encore un élément non recouvert dans $\mathcal{M}_2$.
  - Si $n_l < k - \lceil \log_2 l \rceil$ ou $m_l < k - \lceil \log_2 l \rceil$, alors on a $m_l = n_l$, donc si Joueur I a pu choisir un cycle de longueur l dans $\mathcal{M}_1$, alors Joueur II peut aussi en choisir

un dans $\mathcal{M}_2$.

- Sinon, on a $n_l \geq k - \lceil \log_2 l \rceil$ et $m_l \geq k - \lceil \log_2 l \rceil$.
  - Si $m_l > p$, alors cela signifie qu'il reste au moins un cycle de longueur l non touché dans $\mathcal{M}_2$. Joueur II choisit un élément sur ce cycle.
  - Sinon, on a $p \geq m_l \geq k - \lceil \log_2 l \rceil$. Si Joueur I n'a pas recouvert tous les cycles de longueur l au rang p, Joueur II choisit un élément non recouvert d'un cycle de longueur l. Sinon, cela signifie que Joueur I a recouvert tous les cycles de longueur l au coup p.
    - Si $l = 2^{i-1} + 1$ et $m_l \leq k - \lceil \log_2 l \rceil + 1$, et

$$m_l = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j,$$

      alors on a $m_l = n_l$ d'après la Condition 3, donc si Joueur I a pu prendre un élément dans un cycle de longueur l, Joueur II peut également.
    - Sinon, on a ($m_l \geq k - \lceil \log_2 l \rceil$ et ($l \neq 2^{i-1} + 1$ ou $m_l \neq \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j$)), ou $m_l > k - \lceil \log_2 l \rceil$, en posant $i = \lceil \log_2 l \rceil$.

      Cas où $p = k - i$. Alors, on a $m_l = p = k - i$ et ($l \neq 2^{i-1} + 1$ ou $m_l \neq \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j$). Si Joueur I a recouvert p cycles de longueur l au coup p, alors Joueur II a recouvert p cycles de longueur l au coup p, et nécessairement, pour que Joueur I puisse choisir au coup $p + 1$ un élément non recouvert d'un cycle de longueur l, on a $n_l > k - i$ (En effet, les voisinages au coup p sont de taille $2^i + 1$, et les voisinages des coups d'avant encore plus grands, donc nécessairement, à un cycle de longueur l correspond un cycle de longueur l dans les coups précédents). Alors, on a :

$$\sum_{j=2^{i-1}+1}^{2^i} n_j > k - i = p.$$

      1. Si on a :

$$\sum_{j=2^{i-1}+1}^{2^i} m_j = \sum_{j=2^{i-1}+1}^{2^i} n_j > p,$$

         alors on a encore un élément non recouvert sur un cycle de longueur $\geq 2^{i-1} + 1$. Joueur II peut choisir un tel élément.
      2. Sinon, on a, d'après la Condition 4 :

$$\sum_{j=2^{i-1}+1}^{2^{k-1}+3} m_j > k - i = p,$$

         donc Joueur II peut choisir un élément sur un cycle de longueur $\geq 2^{i-1}+1$ non touché.

      Cas où $p > k - i$. Alors les voisinages sont de taille $2^{k-p} + 1 \leq 2^{i-1} + 1$.

1. Si $l = 2^{i-1} + 1$ et
$$m_l = \sum_{j=2^{i-2}+1}^{2^i} m_j,$$
alors $m_l > k - i + 1$ (et $n_l > k - i + 1$). On ne peut pas recouvrir en p coups $k - i + 2$ cycles de longueur $2^{i-1} + 1$.
2. Si $l = 2^{i-1} + 1$ et
$$m_l < \sum_{j=2^{i-2}+1}^{2^{k-1}+3} m_j,$$
alors il y a au moins $k - i$ cycles de longueur $2^{i-1} + 1$ et au moins un autre cycle de longueur $> 2^{i-2} + 1$, et on ne peut pas les recouvrir tous en p coups.
3. Sinon, on a $l \in \{2^{i-1} + 2, \ldots 2^i\}$, et $m_l \geq k - i$, et on ne peut pas les recouvrir tous en p coups.

$\square$

Ces critères de caractérisation des classes de k-isomorphisme vont nous aider à décrire très précisément les cardinaux des représentants de ces classes.

13.1.4 Etude des cardinaux des représentants des classes de k-isomorphisme

On fixe $k \geq 3$.

Définition 79 (Grand cycle)
Soit $\mathcal{M}$ un modèle de T, et $C = \{n_1, \ldots, n_{2^{k-3}+1}\}$ sa k-configuration. On dit que $l \geq 2^{k-1} + 1$ est une longueur de **grand cycle** *si $\mathcal{M}$ est k-isomorphe à une structure $\mathcal{M}'$ construite à partir de $\mathcal{M}$ en ajoutant un élément à un cycle de longueur l.*

Remarque 25 *Si on a $(n_{2^{k-1}+1}, n_{2^{k-1}+2}, n_{2^{k-1}+3}) \neq (1, 0, 0)$ ou $(n_{2^{k-1}+1}, n_{2^{k-1}+2}, n_{2^{k-1}+3}) \neq (0, 1, 0)$, alors toutes les longueurs de cycles pour $l \geq 2^{k-1}+1$ sont des longueurs de grands cycles.*

Ces grand cycles jouent un rôle particulier sur le cardinal des représentants des classes de k-isomorphisme, car s'il y a un grand cycle, on a un modèle de cardinal n pour tout n assez grand. Dans le cas où il n'y a pas de grand cycle, on peut caractériser précisément les cardinaux atteints par les représentants d'une classe de k-isomorphisme.

13.1.4.1 Classes avec au moins un grand cycle

Proposition 26 (Classe avec un grand cycle)
Soit $\mathcal{M}$ un représentant de $\mathcal{C}$, et $C = \{n_1, \ldots, n_{2^{k-3}+1}\}$. On suppose que $\mathcal{M}$ a au moins un grand cycle. Alors pour tout représentant $\mathcal{M}'$ de $\mathcal{C}$, $\mathcal{M}'$ a également un grand cycle.

Preuve : Soit $\mathcal{M}'$ un représentant de $\mathcal{C}$. Si $\mathcal{M}'$ n'a pas de grand cycle, alors :
- soit $\mathcal{M}'$ n'a aucun cycle de longueur $l \geq 2^{k-1} + 1$, ce qui est impossible d'après la Condition 1 d'équivalence des k-configurations,
- soit $\mathcal{M}'$ a un seul cycle de longueur $l \geq 2^{k-2} + 1$ et ce cycle est de longueur $2^{k-1} + 1$, ce qui est impossible d'après la Condition 5 d'équivalence des k-configurations,
- soit $\mathcal{M}'$ a un seul cycle de longueur $l \geq 2^{k-3} + 1$ et ce cycle est de longueur $2^{k-1} + 2$, ce qui est impossible d'après la Condition 5 d'équivalence des k-configurations.

□

Soit $\mathcal{M}_0$ un représentant de $\mathcal{C}$ de cardinal n_0 minimal. Soit $n \geq n_0$. On peut construire un représentant de $\mathcal{C}$ de cardinal n en ajoutant $n - n_0$ élément à un grand cycle de $\mathcal{M}_0$. Donc, pour tout n assez grand, cette classe a un représentant de cardinal n.

13.1.4.2 Classes sans grand cycle

On ne considère maintenant que des classes sans grand cycle. Le seul moyen d'augmenter le cardinal d'un représentant d'une classe est maintenant de rajouter des cycles de longueur donnée. Ces cycles seront appelés cycles nombreux.

Définition 80 (Cycles nombreux)
Soient $\mathcal{M}$ un modèle de T sans grand cycle, et $C_{\mathcal{M}} = \{n_1, \ldots, n_{2^{k-1}+3}\}$ sa k-configuration. Soit $l \in \{1, \ldots, 2^{k-1}\}$. On dit que des cycles de longueur l sont **nombreux** *si $\mathcal{M}$ est k-isomorphe à la structure $\mathcal{M}'$ obtenue à partir de $\mathcal{M}$ en ajoutant un cycle de longueur l.*

Remarque 26 *Cette notion n'est pas définie pour les cycles moyens, car ceux-ci deviennent grands à partir du moment où les conditions de la définition sont satisfaites.*

Remarque 27 *Si on a $n_l \geq k$, la longueur l est obligatoirement une longueur de cycle nombreux.*

Si le cardinal d'une structure n'ayant pas de grand cycle est assez grand, cela signifie que, nécessairement, il y a des cycles nombreux pour certaines longueurs. On montre alors que les configurations équivalentes associées à des structures de cardinal suffisant ont les mêmes cycles nombreux.

Définition 81 (Ensemble des longueurs des cycles nombreux)
Soit $\mathcal{C}$ une classe de k-isomorphisme sans grand cycle, $\mathcal{M}$ un représentant de $\mathcal{C}$, et $C_{\mathcal{M}} = (n_1, \ldots, n_{2^{k-1}+2}, 0)$ sa configuration associée. Soient $l_1, \ldots, l_s$ les longueurs de cycles nombreux. On pose :

$$LN_{\mathcal{M}} = \{l_1, \ldots, l_s\}$$

*l'***ensemble des longueurs des cycles nombreux** *de la structure $\mathcal{M}$.*

Proposition 27
Soit $\mathcal{M}$ un modèle de T sans grand cycle, soit $C = (n_1, \ldots, n_{2^{k-1}}, 0)$ sa k-configuration, et $LN_{\mathcal{M}} = \{l_1 < \ldots < l_m\}$ son ensemble de longueurs de cycles nombreux. Alors, si $l < l_m$ et $l \notin LN_{\mathcal{M}}$, alors on a $n_l < k - \lceil \log_2 l \rceil$.

Preuve : Soit l tel que $l \notin LN_{\mathcal{M}}$ et $l < l_m$. On suppose que $n_l \geq k - \lceil \log_2 l \rceil$. Soit $\mathcal{M}'$ la structure construite à partir de $\mathcal{M}$ en ajoutant un $(n_l + 1)$ième cycle de longueur l. Alors :

- la condition 1 d'équivalence des configurations est vérifiée (on n'a pas touché aux cycles de longueur $\geq 2^{k-1} + 1$),
- la condition 2 d'équivalence des configurations est vérifiée,
- la condition 3 d'équivalence des configurations est vérifiée (les prémisses sont fausses),
- la condition 4 d'équivalence des configurations est vérifiée (les cycles de longueur l_m sont nombreux et comptent dans la somme de nombres de cycles de longueur $\geq 2^{i-1} + 1$),
- la condition 5 d'équivalence des configurations est vérifiée (on n'a pas touché aux cycles de longueur $\geq 2^{k-1} + 1$).

Donc les deux structures sont k-isomorphes, ce qui contredit $l \notin LN_{\mathcal{M}}$, d'ou $n_l < k - \lceil \log_2 \rceil$.

□

Remarque 28 *Si on a l une longueur de cycles nombreux, on a nécessairement $n_l \geq k - \lceil \log_2 \rceil$ d'après la Condition 2.*

Remarque 29 *Si l est une longueur de cycles nombreux dans une structure $\mathcal{M}$, alors l reste une longueur de cycles nombreux dans toute structure $\mathcal{M}'$ construite à partir de $\mathcal{M}$ en ajoutant un nombre arbitraire de cycles de longueur l.*

Proposition 28
Soit $\mathcal{C}$ une classe sans grand cycle, et $\mathcal{M}_1$, $\mathcal{M}_2$ deux représentants de $\mathcal{C}$ ayant des ensembles non vides de longueurs de cycles nombreux. Alors, on a :

$$LN_{\mathcal{M}_1} = LN_{\mathcal{M}_2}.$$

Preuve : On note $LN_{\mathcal{M}_1} = \{l_1^1, \ldots, l_r^1\}$ et $LN_{\mathcal{M}_2} = \{l_1^2, \ldots, l_s^2\}$. On suppose que $l_1^1 \leq \ldots \leq l_r^1$ et $l_1^2 \leq \ldots \leq l_s^2$. On suppose de plus que $r \leq s$. On note $C_{\mathcal{M}_1} = (n_1^1, \ldots, n_{2^{k-1}}^1, 0)$ la configuration associée à $\mathcal{M}_1$ et $C_{\mathcal{M}_2} = (n_1^2, \ldots, n_{2^{k-1}}^2, 0)$ la configuration associée à $\mathcal{M}_2$. On montre par récurrence sur $p \in \{1, \ldots, r\}$, qu'on a $l_p^1 = l_p^2$.

Pour $p = 1$, on a, d'une part :

$$\forall i \in \{1, \ldots, l_1^1 - 1\},\ n_i^1 < k - \lceil \log_2 i \rceil,$$

et, d'autre part :

$$\forall i \in \{1, \ldots, l_1^2 - 1\},\ n_i^2 < k - \lceil \log_2 i \rceil,$$

par définition des ensembles LN et d'après la Proposition 27. Les configurations sont équivalentes, donc nécessairement, on a $l_1^1 = l_1^2$.

Montrons que l'hypothèse au rang p entraîne l'hypothèse au rang $p+1$. On a donc $l_p^1 = l_p^2 = l_p$. On a, d'une part :

$$\forall i \in \{l_p + 1, \ldots, l_{p+1}^1 - 1\},\ n_i^1 < k - \lceil \log_2 i \rceil,$$

et, d'autre part :

$$\forall i \in \{l_p + 1, \ldots, l_{p+1}^2 - 1\},\ n_i^2 < k - \lceil \log_2 i \rceil,$$

par définition des ensembles LN et d'après la Proposition 27. Les configurations sont équivalentes, donc nécessairement, on a $l_{p+1}^1 = l_{p+1}^2$.

Il reste à montrer que $r = s$. On raisonne par l'absurde : on suppose que $r < s$, et on montre que $l_s \in LN_{\mathcal{M}_1}$. Pour cela, on considère la structure $\mathcal{M}_1'$ construite à partie de $\mathcal{M}_1$ en ajoutant un cycle de longueur l_s. On montre qu'il y a équivalence entre les k-configurations de $\mathcal{M}_1$ et $\mathcal{M}_1'$ en passant par l'équivalence des configurations entre $\mathcal{M}_1$ et $\mathcal{M}_2$. Par définition d'une longueur de cycles nombreux, on peut supposer $n_{l_s}^2$ arbitrairement grand, par exemple égal à k.

- La Condition 1 est vérifiée, car on ne touche pas aux cycles de longueur $\geq 2^{k-1} + 1$.
- La Condition 2 est vérifiée, car du fait de l'équivalence entre $\mathcal{M}_1$ et $\mathcal{M}_2$, on a $n_{l_s}^1 \geq k - \lceil \log_2 l_s \rceil$, et d'autre part on a $n_{l_s} + 1 > n_{l_s} \geq k - \lceil \log_2 l_s \rceil$.

- La Condition 3 est vérifiée, car si $l_s = 2^{i-1}+1$ et $n^1_{l_s} = \sum_{j=2^{i-2}+1}^{2^{k-1}+3} n^1_j$, alors du fait de l'équivalence entre $\mathcal{M}_1$ et $\mathcal{M}_2$, on a $n^1_{l_s} > k - \lceil \log_2 l_s \rceil + 1$, et d'autre part on a $n_{l_s} + 1 > n_{l_s} > k - \lceil \log_2 l_s \rceil + 1$.
- La Condition 4 est vérifiée, car d'après l'équivalence entre $\mathcal{M}_1$ et $\mathcal{M}_2$, on a $\sum_{j=2^{i-1}+1}^{2^{k-1}+3} n^1_j > k - i$ et d'autre part, $\sum_{j=2^{i-1}+1}^{2^{k-1}+3} n^1_j + n_{l_s} > \sum_{j=2^{i-1}+1}^{2^{k-1}+3} n^1_j > k - i$.
- La Condition 5 est vérifiée car on ne touche pas aux cycles de longueur $\geq 2^{k-1}+1$.

On ne peut donc avoir $r < s$.

Par conséquent, les deux configurations ont les mêmes ensembles de longueurs de cycles nombreux.

□

On peut donc associer à une classe de k-isomorphisme un ensemble de longueurs de cycles nombreux, qui est l'ensemble LN commun à tous ses représentants de cardinal suffisamment grand. On considère par exemple, que si la structure est de cardinal $n > k \times (2^{k-1}+2)$, alors elle est de cardinal suffisamment grand (c'est assez grand pour garantir qu'on a des cycles nombreux).

Définition 82 (Cycle nombreux dans une classe)
Soit $\mathcal{C}$ une classe de k-isomorphisme sans grand cycle et possédant au moins un représentant $\mathcal{M}$ de cardinal $n > k \times (2^{k-1}+2)$. On note

$$LN_{\mathcal{C}} = LN_{\mathcal{M}}$$

*l'***ensemble des longueurs des cycles nombreux** *de la classe $\mathcal{C}$.*

Remarque 30 *Comme les longueurs des cycles nombreux sont toujours les mêmes, les longueurs des cycles qui ne sont pas nombreux sont également toujours les mêmes. Comme les cycles peu nombreux sont toujours en même nombre, pour tout représentant d'une classe donnée, on peut parler du nombre de cycles commun à tous les représentants pour toute longueur où les cycles ne sont pas nombreux.*

Définition 83 (Configuration d'une classe)
Soit $\mathcal{C}$ une classe de k-isomorphisme sans grand cycle et possédant au moins un représentant $\mathcal{M}$ de cardinal $n > k \times (2^{k-1}+2)$. On définit la **configuration** *de la classe $\mathcal{C}$ comme étant un ensemble de la forme :*

$$C_{\mathcal{C}} = \{n_1, \ldots, n_{2^{k-1}+3}\},$$

où n_i est le signe ∞ si $i \in LN_{\mathcal{C}}$, et n_i est le nombre de cycles de longueur i commun à tous les représentants de $\mathcal{C}$ sinon.

Cette notion est bien définie d'après la remarque et la proposition précédentes. On peut désormais caractériser les cardinaux des représentants d'une classe de k-isomorphisme sans grand cycle.

Proposition 29 *On considère une classe de k-isomorphisme $\mathcal{C}$ sans grand cycle, ayant au moins un représentant de cardinal $n > k \times (2^{k-1}+2)$. On note $LN_{\mathcal{C}} = \{l_1, \ldots, l_m\}$ l'ensemble des longueurs des cycles nombreux de $\mathcal{C}$, et les longueurs des cycles peu nombreux sont : $\{l'_1, \ldots, l'_r\}$. On note $C_{\mathcal{C}} = \{n_1, \ldots, n_{2^{k-1}+3}\}$ la configuration de la classe $\mathcal{C}$. Alors :*

1. *tout représentant de C de cardinal $n > k \times (2^{k-1}+2)$ satisfait :*

$$n \equiv \sum_{i=1}^{r} n_{l'_i} l'_i \ [pgcd(l_1, \ldots, l_m)].$$

2. *Pour tout nombre n assez grand de la forme*

$$n \equiv \sum_{i=1}^{r} n_{l'_i} l'_i \ [pgcd(l_1, \ldots, l_m)],$$

il existe un représentant de $\mathcal{C}$ de cardinal n.

Preuve : On pose $d = \mathrm{pgcd}(l_1, \ldots, l_m)$.

Soit $\mathcal{X}$ un représentant de $\mathcal{C}$. Il compte $n_{l'_i}$ cycles de longueur l'_i pour $1 \leq i \leq r$ et de nombreux cycles de longueur l_j pour $1 \leq j \leq m$. Son nombre d'éléments est donc de la forme :

$$n = \sum_{i=1}^{r} n_{l'_i} l'_i + \sum_{j=1}^{m} \alpha_j l_j.$$

avec les $\alpha_j > 0$. Donc :

$$n \equiv \sum_{i=1}^{r} n_{l'_i} l'_i \ [\mathrm{pgcd}(l_1, \ldots, l_m)].$$

On suppose maintenant qu'on a un nombre n assez grand de la forme :

$$n = \sum_{i=1}^{r} n_{l'_i} l'_i + \alpha d.$$

On veut montrer qu'il existe un représentant de C de ce cardinal, tout en précisant la notion de “assez grand”.
Les cycles de longueur $l_1, \ldots, l_m$ sont nombreux, il y en a donc au moins $N_i = k - \lceil \log_2 l_i \rceil$ pour la longueur l_i. On pose $n_1 = n - (\sum_{i=1}^{r} n_{l'_i} l'_i + \sum_{j=1}^{m} N_j l_j)$. Le nombre n_1 est de la forme βd. On veut montrer que ce nombre n_1 peut correspondre à une somme de la forme $\sum_{j=1}^{m} \mu_j l_j$ pour représenter une décomposition du représentant de $\mathcal{C}$ que l'on

veut construire. On pose $a_1 = l_1/d, \ldots, a_m = l_m/d$. On a donc ramené le problème à la décomposition de β (pour un β assez grand) sous la forme :

$$\beta = \sum_{j=1}^{m} \alpha_j a_j. \text{ (avec } \alpha_j \geq 1).$$

Cette décomposition nous est donnée par le lemme général suivant :

Lemme 9 *Soient $a_1, \ldots a_m$ des nombres entiers tels que $pgcd(a_1, \ldots a_m) = 1$. Alors pour tout $\beta \geq m \times ppcm(a_1, \ldots a_m)$, il existe $\alpha_1, \ldots \alpha_m$ tels que pour tout $i \in \{1 \ldots m\}$, on ait $\alpha_i \geq 1$, et*

$$\beta = \sum_{i=1}^{m} \alpha_i a_i.$$

Preuve : (du lemme)
Comme $\text{pgcd}(a_1, \ldots a_m) = 1$, il existe $\beta_1, \ldots, \beta_m$ non nuls tels que

$$1 = \sum_{i=1}^{m} \beta_i a_i.$$

On suppose, sans perte de généralité, que $\beta_1, \ldots \beta_r > 0$ et $\beta_{r+1}, \ldots \beta_m < 0$. On pose $u = \text{ppcm}(a_1, \ldots a_m)$. On a :

$$\beta = \underbrace{\beta\beta_1 a_1 + \ldots + \beta\beta_r a_r}_{>0} + \underbrace{\beta\beta_{r+1} a_{r+1} + \ldots + \beta\beta_m a_m}_{<0}.$$

Pour i entre 1 et r : on pose $\beta\beta_i = Q_i \frac{u}{a_i} + R_i$, avec $\frac{u}{a_i} \geq R_i > 0$ (Variante de la division euclidienne rendue possible par le fait que chaque $\beta\beta_i$ est > 0). On a alors :

$$\beta = R_1 a_1 + \ldots + R_r a_r + (\sum_{i=1}^{r} Q_i)u + \beta\beta_{r+1} a_{r+1} + \ldots + \beta\beta_m a_m.$$

Pour i entre $r+1$ et $m-1$: on pose $\beta\beta_i = Q'_i \frac{u}{a_i} + R'_i$, avec $\frac{u}{a_i} \geq R'_i > 0$ (et $Q'_i \leq 0$). On a alors :

$$\beta = R_1 a_1 + \ldots + R'_{m-1} a_{m-1} + \left[\frac{u}{a_m}(\sum_{i=1}^{r} Q_i + \sum_{i=r+1}^{m-1} Q'_i) + \beta\beta_m\right] a_m.$$

Comme on a $\beta \geq mu$ et :

$$R_1 a_1 + \ldots + R_r a_r + R_{r+1} a_{r+1} + \ldots + R'_{m-1} a_{m-1} \leq (m-1)u,$$

nécessairement le coefficient devant a_m est strictement positif. Tous les coefficients devant les a_i ont été rendus strictement positifs, c'est ce qu'on voulait.

□

En conclusion, pour $n \geq \max(k \times (2^{k-1}+2)+1, \sum_{i=1}^{r} n_{l'_i} l'_i + \sum_{j=1}^{m} N_j l_j + m \times \mathrm{ppcm}(l_1/d, \ldots l_m/d))$ qui satisfait la congruence voulue, on a un représentant de la classe $\mathcal{C}$ de cardinal n.

□

13.1.4.3 Cas d'un grand nombre premier p

Afin de prouver que la conjecture de Ash constante n'est pas vérifiée pour un graphe de bijection, on s'intéresse au cas particulier où le pgcd des longueurs des cycles nombreux est un grand premier p, compris entre $2^{k-2}+1$ et 2^{k-1}. On montre notamment la proposition suivante.

Proposition 30 *Pour tout nombre premier $p \in \{2^{k-2}+1, \ldots, 2^{k-1}\}$, le nombre de classes de k-isomorphisme sans grand cycle ayant p comme pgcd des longueurs de cycles nombreux n'est pas divisible par p.*

Cette proposition découle du Lemme 10 et du Lemme 11.

Lemme 10 *Soit $p \in \{2^{k-2}+1, \ldots, 2^{k-1}\}$. Soit $\mathcal{C}$ une classe sans grand cycle, et ayant pour longueurs de cycles nombreux les éléments de $LN_{\mathcal{C}} = \{l_1, \ldots, l_m\}$. On suppose que pgcd$(l_1, \ldots, l_m) = p$. Alors $LN_{\mathcal{C}} = \{p\}$ et il n'y a aucun cycle de longueur $l > 2^{k-2}$, sauf des cycles de longueur p.*

Preuve : Pour que $\mathrm{pgcd}(l_1, \ldots, l_m) = p$, il faut que p divise tous les l_i. Or chaque l_i est inférieur à $2^{k-1} < 2p$, donc $L = \{p\}$. Donc pour toute longueur de cycle $l \neq p$ comprise entre $2^{k-2}+1$ et p, on a :

$$n_l < k - \lceil \log_2 l \rceil \leq k - \lceil \log_2(2^{k-2}+1) \rceil = 1.$$

Il n'y a donc aucun cycle de longueur $l \neq p$ comprise entre $2^{k-2}+1$ et p. D'autre part, s'il y a un $l \geq p+1$ tel que $n_l \geq k - \lceil \log_2(2^{k-2}+1) \rceil = 1$, on montre alors par l'examen des Conditions 1 à 5 que la configuration est équivalente à celle où l'on rajoute 1 à n_l, autrement dit $l \in LN_{\mathcal{C}}$, ce qui est impossible. Il n'y a donc aucun cycle de longueur $l > p$ non plus.

□

Lemme 11 *Le nombre de classes de k-isomorphisme sans grand cycle dont le pgcd des longueurs des cycles nombreux est p est :*

$$(k-1)\prod_{h=0}^{k-3}(k-h-2)^{2^h}.$$

Preuve : On note dans le tableau suivant le nombre de cycles possibles pour les différentes longueurs dans une classe sans grand cycle (comme on a des cycles de longueur p, les nombres possibles de cycles des autres longueurs sont comprises entre 0 et $k - \lceil \log_2 l \rceil$) :

Longueur	1	2	3 4	...	$2^{k-2}+1$ $\dots p \dots$ 2^{k-1}	$\geq 2^{k-1}+1$
Nombre	0 $\vdots$ $\vdots$ $\vdots$ $k-1$	0 $\vdots$ $\vdots$ $k-2$	0 $\vdots$ $k-3$	...	$0 \dots 0$ ∞ $0 \dots 0$	0

Le nombre de classes est alors :

$$(k-1)\times(k-2)\times(k-3)^2\times\ldots\times(k-(k-3)-2)^{2^{k-3}} = (k-1)\prod_{h=0}^{k-3}(k-h-2)^{2^h}.$$

□

Preuve : (de la Proposition 30) Comme $p > 2^{k-2}$ et pour tout $h \in \{0,\dots k-3\}$, on a $k-h-2 < 2^{k-2}$, on a clairement :

$$p \not| \,(k-1)\prod_{h=0}^{k-3}(k-h-2)^{2^h}.$$

□

Corollaire 6 *Le nombre $N_{p,k}(n)$ de classes de k-isomorphisme sans grand cycle qui ont un pgcd de longueurs de cycles nombreux égal à p et qui possèdent un représentant de taille n est ultimement périodique de période p, et non ultimement constant.*

Preuve : La périodicité de période p découle de la Proposition 29. On scinde les classes de k-isomorphisme dont le pgcd des longueurs des cycles nombreux est p en p catégories :
- celles dont le cardinal des représentants est congru à 0 modulo p,
- celles dont le cardinal des représentants est congru à 1 modulo p,
- ...
- celles dont le cardinal des représentants est congru à $p-1$ modulo p.

Si pour toutes les catégories, on a le même nombre de classes, le nombre p divise nécessairement le nombre total de classes dont le pgcd des longueurs des cycles nombreux est p. Ce qui n'est pas le cas d'après la Proposition 30.

□

13.1.5 La conjecture de Ash constante n'est pas vérifiée

Analysons le nombre de classes de k-isomorphisme. On peut séparer les classes en trois catégories :

1. les classes ayant au moins un grand cycle. Celles-ci ont un représentant de taille n, pour tout n assez grand. On note N_∞ le nombre de ces classes ;
2. les classes dont les représentants sont de cardinal borné. Celles-ci n'interviennent pas dans le comportement asymptotique de la fonction de comptage des classes ;
3. les classes sans grand cycle, et ayant au moins un représentant de cardinal $n > k \times (2^{k-1}+2)$. La cardinalité des représentants d'une classe $\mathcal{C}$ de cette catégorie est périodique de période le pgcd des longueurs des cycles nombreux.

On note Λ_k l'ensemble des nombres entiers que l'on peut atteindre en calculant tous les pgcd de m nombres pris dans $\{1,\ldots,2^{k-1}\}$ $(m \geq 1)$. On note également $N_{d,k}(n)$ le nombre de classe de k-isomorphisme appartenant à la troisième catégorie dont le pgcd des longueurs des cycles nombreux est d et qui ont au moins un représentant de cardinal n. On peut décomposer le nombre de classes de k-isomorphisme ayant un représentant de cardinal n de la façon suivante :

$$N_{T,k}(n) = N_\infty + \sum_{d\in\Lambda_k} N_{d,k}(n).$$

Il suffit alors de remarquer les faits suivants :
- D'après la Proposition 29, chaque $n \mapsto N_{d,k}(n)$ est ultimement périodique de période d (mais peut être ultimement constant). Donc $n \mapsto N_{T,k}(n)$ est ultimement périodique (de période un diviseur de $\prod_{d\in\Lambda_k} d$). On retrouve le résultat du Théorème 18.
- On peut toujours trouver un nombre premier p entre $2^{k-2}+1$ et 2^{k-1}, pour $k \geq 3$.
- Pour un tel p, $n \mapsto N_{p,k}(n)$ est ultimement périodique de période p et non ultimement constant d'après le Corollaire 6 de la Proposition 30. Comme p est premier avec tous les autres pgcd, sa contribution non constante à $N_{T,k}(n)$ ne peut pas être compensée par les autres $N_{d,k}(n)$. Donc $n \mapsto N_{T,k}(n)$ est non ultimement constante.

Nous venons donc de prouver le Théorème 17.

13.1.6 Conclusion

Nous venons d'exhiber un exemple de théorie pour laquelle le comportement de la fonction de comptage des classes de k-isomorphisme est périodique mais non constant. Nous avions déjà l'exemple de la théorie des algèbres de Boole (Chapitre 11, Paragraphe 11.2.4), mais comme nous l'avons déjà fait remarquer, la théorie des algèbres de Boole est très contrainte, alors que celle d'un graphe de bijection l'est un peu moins (il suffit de tourner en rond). Cela nous conforte dans l'intuition selon laquelle plus la théorie est contrainte et moins la fonction de comptage est régulière, et inversement. Il y a donc une possibilité pour que la conjection de Ash (constante) soit vraie quand la théorie est la théorie vide.

D'autre part, nous venons de montrer que le complémentaire d'un spectre de bijection est un spectre, comme conséquence de la conjecture de Ash périodique.

13.2 Généralisations d'un graphe de bijection

Nous allons désormais nous intéresser à des théories qui ressemblent à celle d'un graphe de bijection, et pour lesquelles on a un résultat positif à la conjecture de Ash périodique. Nous supposons que le langage contient à chaque fois l'égalité. Il s'agit de :
- un graphe de fonction (Paragraphe 13.2.1),
- un graphe de fonction et des relations unaires (Paragraphe 13.2.2).
- un graphe non orienté de degré 2 (Paragraphe 13.2.3),
- un graphe orienté de degré total 2 (Paragraphe 13.2.4),

13.2.1 Un graphe de fonction

Le langage σ se compose de l'égalité et d'un prédicat F. La théorie d'un graphe de fonction est la suivante :

$$T_f = \{\forall x(\exists y[F(x,y) \wedge \forall z(F(x,z) \rightarrow (z=y))])\}$$

(on a perdu l'existence et l'unicité de l'antécédent).

Les modèles finis de cette théorie ressemblent aux modèles finis de la théorie d'un graphe de bijection, à ceci près qu'on a la possibilité d'ajouter des arbres finis dont la racine est un élément d'un cycle. On peut voir Figure 13.3 un exemple d'un modèle de T_f.

On montre le Théorème 19 suivant :

Theorème 19 (Conjecture de Ash pour un graphe de fonction)
Soit $k \geq 3$. La fonction de Ash $n \mapsto N_{T_f,k}(n)$ est ultimement périodique.

Preuve : Pour montrer ce résultat, nous réalisons des découpages de branches et de cycles, méthode initialement utilisée par Durand, Fagin et Loesher dans [DFL98]. Ces découpages

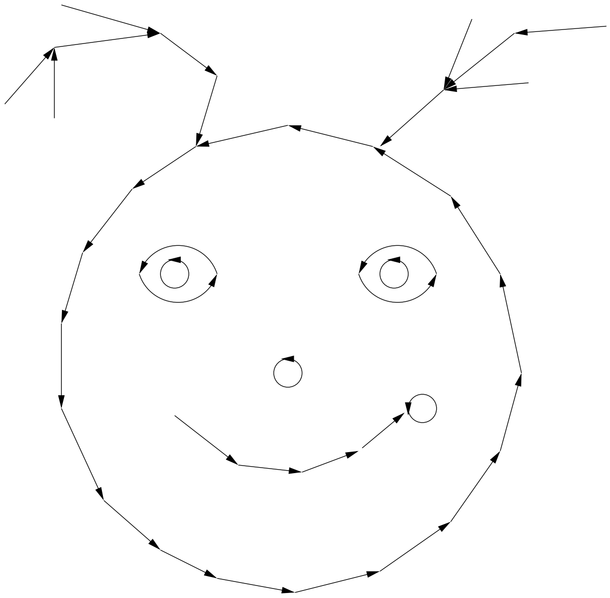


FIG. 13.3 – Un modèle fini de la théorie d'un graphe de fonction.

sont semblables au découpage des grands cycles présenté au Paragraphe 13.1.2.2. On obtient le résultat présenté ci-dessous.

Proposition 31
Soit $k \geq 2$. Il existe deux constantes A_k et L_k telles que tout modèle fini de T_f est k-isomorphe à un modèle fini de T_f de même cardinal dont les cycles sont de longueur inférieure ou égale à L_k et dont tous les arbres présents sur ces cycles sont de taille inférieure ou égale à A_f.

Ce résultat est démontré dans [DFL98].

Définition 84 (Cycle chevelu)
On appelle **cycle chevelu** *une composante connexe d'un modèle fini de T_f.*

Il n'y a qu'un nombre fini de cycles chevelus satisfaisant les deux conditions :
- tous les cycles sont de longueur inférieure ou égale à L_k,
- tous les arbres connectés à ces cycles sont de taille inférieure ou égale à A_k.

Notons D_k ce nombre de cycles chevelus, et Γ la taille maximale d'un tel cycle chevelu. Soit maintenant une classe $\mathcal{C}$ de k-isomorphisme de modèles finis de T_f qui possède un nombre infini de représentants (ce qui implique qu'elle contient des représentants de cardinal arbitrairement grand). On considère un représentant $\mathcal{M}$ de $\mathcal{C}$ de cardinal

$n > k \times D_k \times \Gamma$.
Alors $\mathcal{M}$ est k-isomorphe (par découpages) à un représentant $\mathcal{M}'$ de $\mathcal{C}$ de même cardinal que $\mathcal{M}$ et tel que ses cycles sont de longueur inférieure ou égale à L_k et les arbres sur les cycles sont de taille inférieure ou égale à A_k.
Alors, nécessairement, il existe un des cycles chevelus en plus de k exemplaires. Soit m la taille de ce cycle chevelu. On peut ajouter au modèle $\mathcal{M}'$ autant d'exemplaires de ce cycle chevelu que l'on veut, sans sortir de la classe de k-équivalence $\mathcal{C}$. Cela signifie que tous les cardinaux de la forme $n + \alpha m$ pour $\alpha \in \mathbb{N}$ sont atteints dans la classe $\mathcal{C}$.
Si l'on réalise la même opération pour tous les représentants de $\mathcal{C}$ dont le cardinal dépasse $k \times D_k \times \Gamma$, on peut en conclure que la classe $\mathcal{C}$ apparaît périodiquement dans la fonction de comptage des classes de k-isomorphisme de modèles finis de T_f, avec une période divisant $\text{ppcm}(m)_{m \leq \Gamma}$.
C'est le cas de toutes les classes de k-isomorphisme de modèles finis de T_f ayant au moins un représentant de cardinal $n > k \times D_k \times \Gamma$. Donc la fonction de Ash $n \mapsto N_{T_f,k}(n)$ est ultimement périodique.

$\square$

Conclusion : on vient de montrer que le complémentaire d'un spectre de fonction est un spectre. En réalité, on peut montrer plus ([DFL98]). Nous faisons état de ces résultats supplémentaires au Paragraphe 13.3.

13.2.2 Un graphe de fonction et des relations unaires

On se donne maintenant un langage $\sigma = \{F, =, C_1, \ldots, C_r\}$, où F est un prédicat binaire, et les C_i des prédicats unaires (coloriage), et la théorie est T_f (on ne précise pas de règle pour le coloriage). On assiste alors au phénomène suivant : les σ-structures qui sont modèles de T_f sont encore des cycles avec des cheveux, mais on colorie les éléments de la structure avec r couleurs. La discussion sur le découpage reste exactement la même, seules changent les bornes A_k et L_k, qui augmentent fortement. On peut donc également prouver un résultat de périodicité de la fonction de comptage des classes de k-isomorphisme lorsqu'on rajoute un coloriage à un graphe de fonction.

Conclusion : le complémentaire d'un spectre de fonction avec des relations unaires est un spectre.

13.2.3 Un graphe non orienté de degré 2

Le langage σ se compose de l'égalité et d'un prédicat G. La théorie d'un graphe non orienté de degré 2 est la suivante :

$$\begin{aligned} T_{no2} = & \{\forall x \forall y [G(x,y) \to G(y,x)] \wedge \forall x \forall y \forall z ((y \neq z) \wedge (G(y,x) \wedge G(z,x))) \\ & \to \forall t [((t \neq y) \wedge (t \neq z)) \to (\neg G(t,x))]\}. \end{aligned}$$

Les modèles finis de cette théorie se présentent sous la forme de cycles et de chaînes comme sur la Figure 13.4.

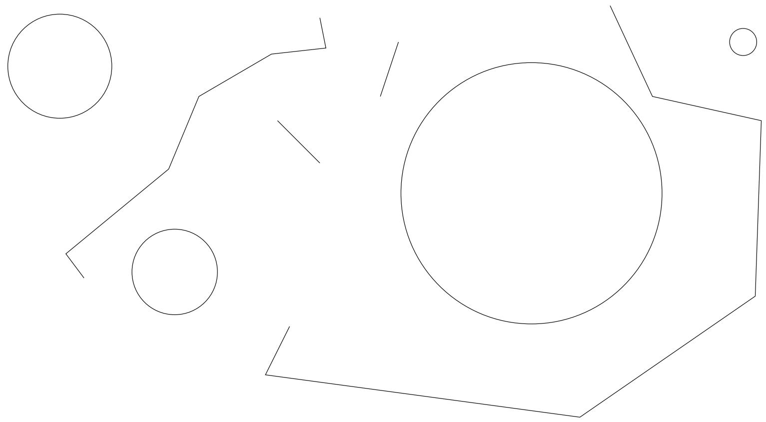


FIG. 13.4 – Un modèle fini de la théorie d'un graphe non orienté de degré 2.

On montre le Théorème 20 suivant :

Theorème 20 (Conjecture de Ash (graphe non orienté de degré 2))
Soit $k \geq 3$. La fonction $n \mapsto N_{T_{no2},k}(n)$ est ultimement périodique.

Preuve : Nous utilisons encore une fois une technique de découpage. On montre aisément, par des méthodes analogues à celles utilisées dans [DFL98], et au Paragraphe 13.1.2.2, qu'il existe deux longueurs L_k et C_k telle que tout modèle fini de T_{o2} est k-isomorphe à un modèle fini de T_{o2} tel que toutes ses chaînes sont de longueur inférieure ou égale à L_k et tous ses cycles de longueurs inférieures ou égale à C_k. On conclut alors de la même manière qu'au Paragraphe 13.2.1.

□

Nous pouvons raisonnablement conjecturer que la conjecture de Ash constante n'est pas vraie dans ce cas. En effet, les objets que l'on rencontre dans les modèles de T_{no2} ressemblent fortement à ceux que l'on rencontre dans les graphes de bijection.

Conclusion : le complémentaire d'un spectre de graphe non orienté de degré 2 est un spectre.

Remarque 31 *Dans ce cas, on peut également ajouter des relations unaires sans changer le résultat.*

13.2.4 Un graphe orienté de degré total 2

Le langage σ se compose de l'égalité et d'un prédicat G. La théorie d'un graphe orienté de degré total 2 est la suivante :

$$\begin{aligned}T_{o2} = \{&\forall x \forall y \forall z([(y \neq z) \wedge [(G(y,x) \wedge G(z,x)) \vee (G(y,x) \wedge G(x,z)) \\ &\vee (G(x,y) \wedge G(x,z))]]) \rightarrow \forall t[((t \neq y) \wedge (t \neq z)) \rightarrow (\neg G(t,x) \wedge \neg G(x,t))]\}.\end{aligned}$$

Les modèles finis de cette théorie se présentent sous la forme de cycles et de chaînes comme sur la Figure 13.5.

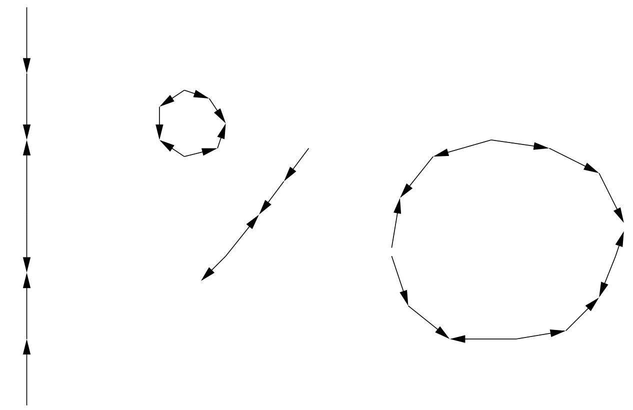


FIG. 13.5 – Un modèle fini de la théorie d'un graphe orienté de degré total 2.

On montre le Théorème 21 suivant :

Theorème 21 (Conjecture de Ash (graphe orienté de degré total 2)) *Soit $k \geq 3$. La fonction $n \mapsto N_{T_{o2},k}(n)$ est ultimement périodique.*

Preuve : Le fait de rajouter une orientation au graphe ne change pas ici la possibilité de découpage par rapport au cas précédent. Cependant, les longueurs à partir desquelles on peut découper sont beaucoup plus importantes dans le cas orienté que dans le cas non orienté. L'idée est qu'on peut associer à un élément d'un modèle de T_{o2} un mot de longueur au plus 2^k décrivant les orientations successives des arêtes dans son voisinage. Pour pouvoir découper, il faut pouvoir être sûr que le cycle ou la chaîne est de longueur suffisante pour que l'on puisse trouver deux éléments $(k-1)$-isomorphes suffisamment éloignés. Cela reste possible, mais les bornes A_k et L_k sont beaucoup plus grandes que dans le cas non orienté.

$\square$

Conclusion : le complémentaire d'un spectre de graphe orienté de degré total 2 est un spectre.

Remarque 32 *Dans ce cas, on peut également ajouter des relations unaires sans changer le résultat.*

13.3 Vers le cas général

13.3.1 Précision sur les fonctions

On peut en dire plus sur le complémentaire d'un spectre de fonction que la simple information qu'il s'agit d'un spectre. En effet, Durand, Fagin et Loescher montrent dans [DFL98] le théorème suivant :

Theorème 22 (Caractérisations d'un spectre de fonction)
Soit σ un langage avec un symbole de fonction unaire. Soit σ_k un langage contenant un symbole de fonction et des symboles de prédicats unaires. Soit S une partie de $\mathbb{N}$. Les assertions suivantes sont équivalentes :

1. *S est le spectre d'un σ-énoncé du premier ordre.*
2. *S est le spectre d'un σ-énoncé du premier ordre, où l'interprétation de F est restreinte à une bijection.*
3. *S est le spectre d'un σ_k-énoncé du premier ordre.*
4. *S est ultimement périodique.*
5. *S est une union finie de suites arithmétiques.*
6. *S est définissable dans l'arithmétique de Presburger.*
7. *Quand S est écrit en unaire, S est reconnaissable par un automate fini.*

Comme le complémentaire d'un ensemble ultimement périodique est également ultimement périodique, ce résultat implique que le complémentaire d'un spectre de σ-énoncé impliquant T_f ou T est également un σ-énoncé impliquant T_f ou T.

Pour poursuivre les travaux d'approche du cas général, nous avons maintenant plusieurs possibilités, dont deux ont retenu notre attention :

- la première possibilité consiste à rechercher des cas intermédiaires plus compliqués que les cas de graphes de degré 2 ou d'un graphe de fonction, en étendant les degrés autorisés pour le graphe ;
- la deuxième piste consiste à se rapprocher d'un cas que l'on sait équivalent au cas général, à savoir le cas de deux fonctions et l'égalité. Pour cela, on peut imaginer des cas intermédiaires menant à des cas où l'on peut coder deux fonctions.

13.3.2 Autres cas intermédiaires vers une relation binaire et l'égalité

Les cas intermédiaires que nous pouvons considérer, sont par exemple :

- un graphe orienté de degré sortant 2, qui généralise le cas d'un graphe de fonction qui est un graphe orienté de degré sortant 1. Il est plus difficile dans ce cas de décrire des structures régulières comme des cycles chevelus ;
- un graphe orienté de degré total 3, dont les modèles sont plus compliqués que des réunions de cycles et de chaînes ;

- un graphe orienté de degré sortant 3, généralisant le cas d'un graphe de degré sortant 2 ;
- un graphe non orienté de degré 3, cas dans lequel encore une fois les structures sont plus compliquées à décrire que dans le cas d'un graphe non orienté de degré 2 (cycles et chaînes) ;
- ...
- et ainsi de suite en augmentant progressivement le degré jusqu'à tomber sur un cas se ramenant au cas général.

13.3.3 Plusieurs relations binaires : vers deux graphes de fonction et l'égalité

Les cas intermédiaires que nous pouvons considérer, sont par exemple :
- un graphe orienté de degré sortant 2, muni de relations unaires (coloriage), cas dont nous pressentons la difficulté égale à celle du cas d'un graphe de degré sortant 2 ;
- un graphe orienté de degré sortant 2 codant deux fonctions ;
- un graphe quelconque codant deux fonctions ;
- deux graphes de fonction avec une contrainte entre les fonctions (par exemple, une relation fonctionnelle implique l'autre) ;
- ...
- et ainsi de suite jusqu'à tomber sur un cas se ramenant au cas général.

Chapitre 14

Résultats sur les équivalences

Dans ce chapitre, on s'intéresse aux relations d'équivalence. Rappelons qu'une relation binaire R est une relation d'équivalence si l'énoncé φ_R est satisfait, où φ_R est l'énoncé de profondeur de quantification 3 suivant :

$$\forall x[R(x,x) \wedge \forall y([R(x,y) \rightarrow R(y,x)] \wedge \forall z[(R(x,y) \wedge R(y,z)) \rightarrow R(x,z)])].$$

Cette piste se décompose en quatre étapes.

- On commence par s'intéresser, au Paragraphe 14.1, au cas des relations d'équivalence sans égalité, pour lequel on montre que la conjecture de Ash constante est vérifiée. Cette étude nous permet de nous familiariser avec les modèles finis de la théorie d'une ou plusieurs équivalences.
- On s'intéresse ensuite, au Paragraphe 14.2, au cas d'une relation d'équivalence et l'égalité. Dans ce cas, on peut montrer que la conjecture de Ash périodique est satisfaite, par des méthodes analogues à celles présentées au Chapitre 13 pour le cas d'une bijection et l'égalité.
- Ensuite, nous essayons d'élargir le cas précédent à deux relations d'équivalence et l'égalité, en commençant par un sous-cas “facile” : on suppose que les deux relations d'équivalence sont imbriquées. On montre ainsi, au Paragraphe 14.3, que la conjecture de Ash périodique est vérifiée dans ce cas.
- Nous présentons ensuite une catégorie de structures de deux équivalences et l'égalité dans lesquelles on peut coder n'importe quel graphe, ce qui nous ramène au cas général via le résultat de Fagin que nous avons présenté dans le Chapitre 11. Nous décrivons ce codage dans le Paragraphe 14.4.

14.1 Plusieurs relations d'équivalence sans égalité

On considère $p \geq 1$ relations d'équivalences, notées $\equiv_1, \ldots, \equiv_p$. Une structure finie où sont interprétées ces relations se présente sous la forme d'un ensemble partitionné de p façons différentes, en classes de $\equiv_1$-équivalence, de $\equiv_2$-équivalence, etc. Nous en présentons un exemple pour 3 équivalences sur la Figure 14.1.

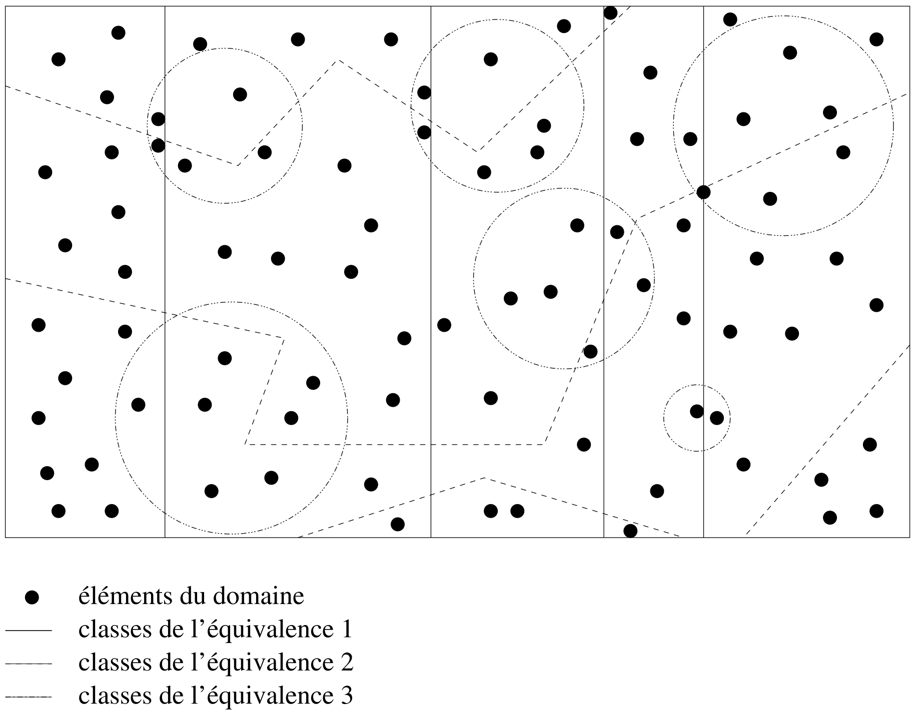


FIG. 14.1 – Une structure finie avec 3 relations d'équivalence.

Comme le langage ne contient pas l'égalité, deux éléments contenus dans une même intersection de relations sont indiscernables : on ne peut pas dire, rien qu'avec les relations d'équivalence, que ce sont deux éléments distincts. Par conséquent, si on rajoute un élément dans cette intersection, on ne peut pas dire que ce modèle est différent du précédent. Autrement dit, les deux structures sont équivalentes, et a fortiori, k-équivalentes. Ceci implique que la fonction de Ash est ultimement constante dans ce cas.

Ce qui distingue les classes de k-isomorphismes de modèles de $\varphi_{\equiv_1} \wedge \ldots \wedge \varphi_{\equiv_p}$, ce n'est donc en aucun cas le cardinal du modèle. En revanche, on peut décrire :
- les types d'intersections que l'on peut rencontrer dans le modèle ;
- le nombre d'intersections d'un type donné, jusqu'à k ($0, 1, \ldots, k-1$ ou "beaucoup").

S'il l'on rajoute l'égalité au langage, on se donne la possibilité de compter les éléments distincts qui apparaissent dans chaque intersection. On ne peut donc pas rajouter un élément au domaine et rester dans la même classe de k-isomorphisme aussi facilement que dans le cas sans égalité.

14.2 Une relation d'équivalence et l'égalité

Nous considérons maintenant le langage $\sigma = \{\equiv, =\}$, et la théorie :

$$T_{\equiv} = \{\varphi_{\equiv}\}.$$

Nous décrivons les classes de k-isomorphisme de modèles de cette théorie au Paragraphe 14.2.1 afin de montrer, au Paragraphe 14.2.3 que la conjecture de Ash périodique est vérifiée pour une relation d'équivalence et l'égalité. Les modèles de $T_\equiv$ se présentent sous la forme d'une partition du domaine en classes de $\equiv$-équivalence. Chacune de ces classes compte un certain nombre d'éléments. Un modèle de $T_\equiv$ peut donc se décrire par la donnée du nombre de classes ayant un élément, le nombre de classes ayant deux éléments, etc. On commence par montrer qu'on ne peut pas compter plus de k éléments par classe au rang de quantification k, autrement dit :

Proposition 32 (Taille des classes)
Soit $\mathcal{A}$ un modèle de $T_\equiv$. Soient C_1 et C_2 deux classes de $\mathcal{A}$ ayant au moins k éléments. Alors les deux classes C_1 et C_2 sont k-isomorphes.

Preuve : On établit une stratégie gagnante pour Joueur II dans un jeu en k coups entre la classe C_1 et la classe C_2. Quand Joueur I choisit un élément de la classe C_i qui a déjà été choisi, Joueur II choisit dans l'autre classe l'élément qui correspond. Quand Joueur I choisit dans la classe C_i un élément qui n'a pas été choisi, Joueur II choisit dans l'autre classe un élément qui n'a pas été choisi. C'est toujours possible, puisque les classes C_1 et C_2 ont au moins k éléments. Il est clair que les k-uples de C_1 et C_2 respectivement choisis dans ce jeu satisfont les mêmes $\{\equiv, =\}$-formules sans quantificateurs.

□

On a donc un analogue des "grands cycles" rencontrés dans l'étude des classes de k-isomorphisme d'un graphe de bijection. Il s'agit des "grandes classes", à savoir les classes ayant plus de k éléments. Ainsi, la donnée d'un k-uple $(n_1, \ldots, n_k)$, où n_i, pour $i < k$, représente le nombre de classes ayant i éléments, et n_k le nombre de classes ayant au moins k éléments, suffit à décrire une structure à k-isomorphisme près. Cependant, il faut, comme pour le cas d'un graphe de bijection, décrire une notion d'équivalence sur ces k-uples pour atteindre une caractérisation des classes de k-isomorphisme de modèles finis de $T_\equiv$.

14.2.1 Caractérisation des classes de k-isomorphismes

Définition 85 (k-configuration)
Une k-configuration est la donnée d'un k-uple d'entiers $(n_1, \ldots, n_k)$.

A chaque modèle de $T_\equiv$ est associé une k-configuration. On caractérise les classes de k-isomorphisme grâce à des conditions satisfaites par les k-configurations de leurs représentants.

Définition 86 (Relation d'équivalence sur les k-configurations)
Soient $C = (n_1, \dots, n_k)$ et $D = (m_1 \dots, m_k)$ deux k-configurations. On dit que ces deux k-configurations C et D sont **équivalentes**, *et on note $C \simeq D$ si :*

1. *pour tout $i \in \{1, \dots k\}$, $n_i \neq m_i \Rightarrow (n_i \geq k - i$ et $m_i \geq k - i)$,*
2. *pour tout $i \in \{1, \dots k\}$, $\sum_{j=i}^{k} n_j \neq \sum_{j=i}^{k} m_j \Rightarrow (\sum_{j=i}^{k} n_j > k - i$ et $\sum_{j=i}^{k} m_j > k - i)$,*

Proposition 33 (Caractérisation des classes de k-isomorphisme)
Deux structures $\mathcal{A}$ et $\mathcal{B}$ modèles de T sont k-isomorphes si et seulement si les k-configurations $C_{\mathcal{A}}$ et $C_{\mathcal{B}}$ qui leurs sont associées sont équivalentes.

Preuve : on a besoin du lemme suivant.

Lemme 12
Soit $k \geq 2$. Pour tout $j \in \{1, \dots, k-1\}$, il existe un énoncé φ_j sur le langage $\{\equiv, =\}$, tel que φ_j exprime l'existence d'une classe à j éléments, et φ_j est de profondeur de quantification inférieure ou égale à $j + 1$.

Preuve : (du lemme)
L'énoncé est simple à exhiber :

$$\begin{aligned}\varphi_j =& \exists x_1 \exists x_2 ((x_1 \neq x_2) \wedge (x_2 \equiv x_1) \wedge \dots \wedge \exists x_j [(x_j \neq x_1) \wedge \dots (x_j \neq x_{j-1}) \wedge (x_j \equiv x_1) \\ & \wedge \forall z ([(z \neq x_1) \wedge \dots \wedge (z \neq x_j)] \to \neg (z \equiv x_1))] \dots) \\ =& \exists x_1 \varphi'_j(x_1).\end{aligned}$$

□

Montrons maintenant la proposition.

$\boxed{\Leftarrow}$: on montre que si les configurations C et D ne sont pas équivalentes, alors $\mathcal{A}$ et $\mathcal{B}$ ne sont pas k-isomorphes. On suppose que les configurations C et D ne sont pas équivalentes. Cela signifie qu'une des conditions d'équivalence est fausse.

1. Si la première condition est fausse : on suppose qu'il existe un $i \in \{1, \dots, k\}$ tel que $n_i < k - i$ et $n_i \neq m_i$. Alors, d'après le Lemme 12, on a un énoncé θ_{n_i} de profondeur de quantification $n_i + i < k$ qui dit qu'il y a exactement n_i classes à i éléments, il s'agit de l'énoncé :

$$\begin{aligned}\theta_{n_i} =& \exists x_1 (\varphi'_i(x_1) \wedge \dots \wedge \exists x_{n_i} [(x_{n_i} \not\equiv x_1) \wedge \dots \wedge (x_{n_i} \not\equiv x_{n_i - 1}) \wedge \varphi'_i(x_{n_i}) \\ & \wedge \forall z [((z \not\equiv x_1) \wedge \dots (z \not\equiv x_{n_i})) \to \neg \varphi'_i(z)]] \dots).\end{aligned}$$

L'énoncé θ_{n_i} est vrai dans $\mathcal{A}$ mais pas dans $\mathcal{B}$. Donc ces deux structures ne peuvent pas être k-isomorphes.

2. Si la deuxième condition est fausse : on suppose qu'il existe un $i \in \{1, \dots, k\}$ tel que $\sum_{j=i}^{k} n_j \neq \sum_{j=i}^{k} m_j$ et $\sum_{j=i}^{k} n_j \leq k - i$.
   - Si
   $$\sum_{j=i}^{k} m_j < k - i,$$
   alors c'est vrai de chaque m_j, et la Condition 1 est mise en défaut. On conclut comme en 1.
   - Sinon, on a :
   $$\sum_{j=i}^{k} m_j \geq k - i.$$
   Comme les deux sommes sont inégales, elles ne peuvent être égales à $k - i$ en même temps. Donc
   $$\sum_{j=i}^{k} m_j > \sum_{j=i}^{k} n_j = p.$$
   Alors il y a un énoncé Ψ_p de profondeur de quantification $\leq k$, qui dit qu'il y a strictement plus de p classes à au moins i éléments. Il s'agit de l'énoncé :
   $$\Psi_p = \exists x_1(\neg\varphi'_1(x_1) \wedge \ldots \wedge \neg\varphi'_{i-1}(x_1) \wedge \ldots \wedge \exists x_{p+1}[(x_{p+1} \not\equiv x_1) \wedge \ldots \\ \wedge (x_{p+1} \not\equiv x_p)] \wedge [\neg\varphi'_1(x_{p+1}) \wedge \ldots \wedge \neg\varphi'_{i-1}(x_{p+1})]) \ldots).$$
   Cet énoncé est vrai dans $\mathcal{B}$ et pas dans $\mathcal{A}$, donc ces deux structures ne peuvent pas être k-isomorphes.

$\boxed{\Rightarrow}$: on suppose maintenant que les configurations C et D sont équivalentes, et on doit montrer que les structures $\mathcal{A}$ et $\mathcal{B}$ sont k-isomorphes. On définit une stratégie gagnante pour Joueur II dans un jeu d'Ehrenfeucht en k coups entre ces deux structures. Si x est un élément d'une des structures $\mathcal{A}$ ou $\mathcal{B}$, on note $[x]$ la classe de $\equiv$-équivalence de l'élément x.

- Premier coup : si Joueur I choisit un élément de $\mathcal{A}$ dans une classe à j éléments, Joueur II choisit un élement de $\mathcal{B}$ dans une classe à j éléments et inversement. C'est toujours possible car $n_j \neq 0$ si et seulement si $m_j \neq 0$.
- ...
- On suppose qu'on a joué $p < k$ coups, et qu'ont été choisis des couples d'éléments dans $\mathcal{A}$ et $\mathcal{B}$, notés $(a_i, b_i)_{i \in \{1,\dots,p\}}$, tels que l'on ait les trois propriétes suivantes :
  1. on a un isomorphisme partiel entre le p-uple $(a_1, \dots, a_p)$ et le p-uple $(b_1, \dots, b_p)$,
  2. $|[a_i] \backslash \{a_1, \dots, a_i\}| < k - i \Leftrightarrow |[b_i] \backslash \{b_1, \dots, b_i\}| < k - i$,
  3. $|[a_i] \backslash \{a_1, \dots, a_i\}| < k - i \Rightarrow |[a_i] \backslash \{a_1, \dots, a_i\}| = |[b_i] \backslash \{b_1, \dots, b_i\}|$.

  On suppose que Joueur I choisit un élément a de $\mathcal{A}$.
  - Si a est l'un des a_i, Joueur II choisit le b_i correspondant. Les trois conditions restent valables.

- Si $a \in [a_i]$, pour un certain $i \leq p$, mais sans être un des a_i : on suppose que l'indice i est le plus grand tel que $a \in [a_i]$.

  Premier cas : $|[a_i]\backslash\{a_1, \ldots, a_i\}| < k-i$. Alors $|[a_i]\backslash\{a_1, \ldots, a_i\}| = |[b_i]\backslash\{b_1, \ldots, b_i\}|$. Donc si on a pu choisir a dans $[a_i]\backslash\{a_1, \ldots, a_i\}$, Joueur II peut choisir un b distinct des b_i dans $[b_i]\backslash\{b_1, \ldots, b_i\}$. Les trois conditions restent valables.

  Deuxième cas : $|[a_i]\backslash\{a_1, \ldots, a_i\}| \geq k - i$. Alors $|[b_i]\backslash\{b_1, \ldots, b_i\}| \geq k - i$, et Joueur II peut toujours choisir un b qui convient. Les trois conditions restent valables.

- Si a n'est dans aucun des classes des a_i.

  Premier cas : $j = |[a]| < k - p \Rightarrow p < k - j$.
  - Si $n_j < k - j$, alors $m_j = n_j$, donc il y a au moins une classe à j éléments dans $\mathcal{B}$ qui n'a pas été touchée. Joueur II choisit un b dans l'une de ces classes. Les trois conditions restent valables.
  - Si $n_j \geq k - j$, alors $m_j \geq k - j > p$ et il y a au moins une classe à j éléments dans $\mathcal{B}$ qui n'a pas été touchée. Joueur II choisit un b dans l'une de ces classes. Les trois conditions restent valables.

  Deuxième cas : $j = |[a]| \geq k - p$. On a $\sum_{i=k-p}^{k} n_i \leq p$ si et seulement si $\sum_{i=k-p}^{k} m_i \leq p$.
  - Si $\sum_{i=k-p}^{k} n_i \leq p$, alors $\sum_{i=k-p}^{k} n_i = \sum_{i=k-p}^{k} m_i$, donc si Joueur I choisit a dans une classe non touchée parmi celles participant au terme de gauche, alors Joueur II peut choisir un b dans une classe non touchée participant au membre de droite. Les trois conditions restent valables.
  - Si $\sum_{i=k-p}^{k} n_i > p$, alors $\sum_{i=k-p}^{k} m_i > p$, et il y a au moins une classe non touchée parmi celles participant à la somme. Joueur II choisit un b dans l'une de ces classes. Les trois conditions restent valables.

- Au bout de k coups, on obtient deux k-uples qui sont en isomorphsime partiel (donc satisfont les mêmes formules sans quantificateurs).

On vient donc de décrire une stratégie gagnante pour Joueur II dans le jeu d'Ehrenfeucht en k coup entre les structures $\mathcal{A}$ et $\mathcal{B}$. Ces deux structures sont donc k-isomorphes.

□

Ces critères de caractérisation des classes de k-isomorphisme vont nous aider à décrire très précisément les cardinaux des représentants de ces classes.

14.2.2 Etude des cardinaux des représentants des classes de k-isomorphisme

On fixe $k \geq 2$. On appelle *grande classe* une classe ayant au moins k éléments.

14.2.2.1 Classes de k-isomorphisme avec au moins une grande classe

Soit $\mathcal{C}$ une classe de k-isomorphisme telle qu'il existe un représentant de $\mathcal{C}$ avec $n_k \neq 0$. Alors tous les représentants de $\mathcal{C}$ ont également au moins une grande classe. Soit $\mathcal{M}_0$ un représentant de $\mathcal{C}$ de cardinal n_0 minimal. Soit $n \geq n_0$. On peut construire un représentant de $\mathcal{C}$ de cardinal n au ajoutant $n - n_0$ élément à une grande classe de $\mathcal{M}_0$. Donc, pour tout n assez grand, cette classe de k-isomorphisme a un représentant de cardinal n.

14.2.2.2 Classes de k-isomorphisme sans grande classe

On ne considère maintenant que des classes de k-isomorphisme dont tous les représentants satisfont $n_k = 0$.

Définition 87 (Classes nombreuses)
Soit C une k-configuration. Soit $l \in \{1, \dots, k\}$. On dit que des classes à l éléments sont **nombreuses** *si :*
- *$n_l \geq k - l$ et*
- *pour tout $l' \in \{1, \dots, l\}$, on a $\sum_{i=l'}^{k} n_i > k - l'$.*

Si le cardinal d'une structure n'ayant pas de grande classe est assez grand, cela signifie que, nécessairement, il y a des classes nombreuses pour certaines tailles de classes. On montre alors que les configurations équivalentes associées à des structures de cardinal suffisant ont les mêmes classes nombreuses.

Définition 88 (Ensemble des tailles des classes nombreuses)
Soit $\mathcal{C}$ une classe de k-isomorphisme, et $\mathcal{M}$ un représentant de $\mathcal{C}$, et $C_{\mathcal{M}} = (n_1, \dots, n_k)$ sa configuration associée. Soient $l_1, \dots, l_s$ les tailles des classes nombreuses. On pose :

$$LN_{\mathcal{M}} = \{l_1, \dots, l_s\}$$

*l'***ensemble des tailles des classes nombreuses** *de la structure $\mathcal{M}$.*

Proposition 34
Soit $\mathcal{M}$ un modèle de $T_{\equiv}$ sans grande classe, soit $C = (n_1, \dots, n_{k-1}, 0)$ sa k-configuration, et $LN_{\mathcal{M}} = \{l_1, \dots, l_m\}$ son ensemble de tailles de classes nombreuses. Pour tout $l \notin LN_{\mathcal{M}}$ et $l < l_m$, alors on a $n_l < k - l$.

Preuve : Soit l tel que $l \notin LN_{\mathcal{M}}$ et $l < l_i$ pour un $i \in \{1, \dots, m\}$. On suppose que $n_l \geq k - l$. Comme l_i est une taille de classe nombreuse, pour tout $l' \leq l_i$, on a $\sum_{j=l'}^{k} n_j > k - l'$. A fortiori, pour tout $l' \leq l$, on a $\sum_{j=l'}^{k} n_j > k - l'$. Ce qui contredit $l \notin LN_{\mathcal{M}}$.

□

Proposition 35
Soit $\mathcal{C}$ une classe de k-isomorphisme sans grande classe, et $\mathcal{M}_1$, $\mathcal{M}_2$ deux représentants de $\mathcal{C}$ ayant des classes nombreuses. Alors, on a :

$$LN_{\mathcal{M}_1} = LN_{\mathcal{M}_2}.$$

Preuve : On note $LN_{\mathcal{M}_1} = \{l_1^1, \ldots, l_r^1\}$ et $LN_{\mathcal{M}_2} = \{l_1^2, \ldots, l_s^2\}$. On suppose que $l_1^1 \leq \ldots \leq l_r^1$ et $l_1^2 \leq \ldots \leq l_s^2$. On suppose de plus que $r \leq s$. On note $C_{\mathcal{M}_1} = (n_1^1, \ldots, n_k^1)$ la configuration associée à $\mathcal{M}_1$ et $C_{\mathcal{M}_2} = (n_1^2, \ldots, n_k^2)$ la configuration associée à $\mathcal{M}_2$. On montre par récurrence sur $p \in \{1, \ldots, r\}$, on a $l_p^1 = l_p^2$.

Pour $p = 1$, on a, d'une part :

$$\forall i \in \{1, \ldots, l_1^1 - 1\},\ n_i^1 < k - i,$$

et, d'autre part :

$$\forall i \in \{1, \ldots, l_1^2 - 1\},\ n_i^2 < k - i,$$

par définition des ensembles LN. Les configurations sont équivalentes, donc nécessairement, on a $l_1^1 = l_1^2$.

Montrons que l'hypothèse au rang p entraîne l'hypothèse au rang $p+1$. On a donc $l_p^1 = l_p^2 = l_p$. On a, d'une part :

$$\forall i \in \{l_p + 1, \ldots, l_{p+1}^1 - 1\},\ n_i^1 < k - i,$$

et, d'autre part :

$$\forall i \in \{l_p + 1, \ldots, l_{p+1}^2 - 1\},\ n_i^2 < k - i,$$

par définition des ensembles LN. Les configurations sont équivalentes, donc nécessairement, on a $l_{p+1}^1 = l_{p+1}^2$.

Il reste à montrer que $r = s$. On a, pour tout $i \in \{1, \ldots, l_s\}$, $\sum_{j=i}^k n_j^2 > k - i$, et $n_{l_s}^2 \geq k - l_s$ donc par équivalence des configurations, on a également pour tout $i \in \{l_r + 1, \ldots, l_s\}$, $\sum_{j=i}^k n_j^1 > k - i$, et $n_{l_s}^1 \geq k - l_s$, ce qui implique que $l_s \in LN_{\mathcal{M}_1}$, donc $s \leq r$, et finalement $s = r$.

Finalement, les deux configurations ont les mêmes ensembles de tailles de classes nombreuses.

□

On peut donc associer à une classe de k-isomorphisme un ensemble de taille de classes nombreuses, qui est l'ensemble LN commun à tous ses représentants de cardinal suffisamment grand. On considère par exemple, que si la structure est de cardinal $n > k^2$, alors elle est de cardinal suffisamment grand (c'est assez grand pour garantir qu'on a des classes nombreuses).

Définition 89 (Classes nombreuses d'une classe de k-isomorphisme)
Soit $\mathcal{C}$ une classe de k-isomorphisme sans grande classe et possédant au moins un représentant $\mathcal{M}$ de cardinal $n > k^2$. On note

$$LN_{\mathcal{C}} = LN_{\mathcal{M}}$$

*l'***ensemble des tailles des classes nombreuses** *de la classe de k-isomorphisme $\mathcal{C}$.*

Définition 90 (Configuration d'une classe de k-isomorphisme)
Soit $\mathcal{C}$ une classe de k-isomorphisme sans grande classe et possédant au moins un représentant $\mathcal{M}$ de cardinal $n > k^2$. On définit la **configuration** *de la classe $\mathcal{C}$ comme étant un ensemble de la forme :*

$$C_{\mathcal{C}} = \{n_1, \ldots, n_k\},$$

où n_i est le signe ∞ si $i \in LN_{\mathcal{C}}$, et n_i est le nombre de classes à i éléments, commun à tous les représentant de $\mathcal{C}$, sinon.

Proposition 36
On considère une classe de k-isomorphisme $\mathcal{C}$ sans grande classe, et ayant au moins un représentant de cardinal $n > k^2$. On note $LN_{\mathcal{C}} = \{l_1, \ldots, l_m\}$ l'ensemble des tailles des classes nombreuses de $\mathcal{C}$, et les tailles des classes peu nombreuses sont : $\{l'_1, \ldots, l'_r\}$. On note $C_{\mathcal{C}} = \{n_1, \ldots, n_k\}$ la configuration de la classe $\mathcal{C}$. Alors :

1. *tout représentant de C de cardinal $n > k^2$ satisfait :*

$$n \equiv \sum_{i=1}^{r} n_{l'_i} l'_i \ [pgcd(l_1, \ldots, l_m)] \ ;$$

2. *Pour tout nombre n assez grand de la forme*

$$n \equiv \sum_{i=1}^{r} n_{l'_i} l'_i \ [pgcd(l_1, \ldots, l_m)],$$

il existe un représentant de $\mathcal{C}$ de cardinal n.

Preuve : La démonstration est en tout point analogue à celle de la Proposition 29 du Chapitre 13.

□

14.2.3 La conjecture de Ash périodique est vérifiée

Le théorème suivant découle immédiatement de la Proposition 36.

Theorème 23 (Ash périodique)
Pour tout $k \geq 2$, la fonction

$$n \mapsto N_{T_{\equiv},k}(n)$$

est ultimement périodique, de période un diviseur de ppcm$(i)_{i=1\dots k}$.

On ne sait pas à l'heure actuelle si cette périodicité est stricte ou non (c'est-à-dire qu'on ne sait pas si la fonction de comptage des classes de k-isomorphisme est ultimement constante ou bien strictement périodique). On ne peut pas utiliser un contre-exemple du même type que pour la bijection, à savoir un “grand premier”, car ici le nombre de classes ayant un grand premier comme pgcd des tailles des classes nombreuses est divisible par ce premier. Il faut donc trouver un contre-exemple par une autre méthode, ou bien prouver par une étude exhaustive des cas possibles que cette fonction de comptage est constante.

14.3 Deux relations d'équivalence imbriquées et l'égalité

On considère désormais le langage $\sigma = \{\equiv_1, \equiv_2, =\}$, et on suppose que la relation $\equiv_2$ est incluse dans la relation $\equiv_1$, c'est-à-dire que chaque classe d'équivalence pour $\equiv_1$ est une réunion de classes d'équivalence pour $\equiv_2$. Autrement dit, la théorie considérée est :

$$T_{\sqsubseteq} = \{\varphi_{\equiv_1}, \varphi_{\equiv_2}, \forall x \forall y ((x \equiv_2 y) \to (x \equiv_1 y))\}.$$

Nous ne faisons pas une étude détaillée des classes de k-isomorphisme de modèles finis de cette théorie. On remarque simplement que deux classes $\equiv_2$ de taille supérieure ou égale à k qui sont contenues dans une même classe $\equiv_1$ sont k-isomorphes. Donc une classe $\equiv_1$ est déterminée, à k-isomorphisme près, par le nombre de classes $\equiv_2$ de taille i qu'elle contient, pour $i = 1, \dots, k-1, \infty$, où la taille ∞ signifie que la classe en question a au moins k éléments. A une classe $\equiv_1$, on peut associer un k-uple $(n_1, \dots, n_k)$, où n_i est le nombre de classes $\equiv_2$ de taille i qui sont incluses dans la classe $\equiv_1$, si $i < k$, et n_k le nombre de classes $\equiv_2$ de taille au moins k. De plus, si l'un des n_i est supérieur ou égal à k, la classe $\equiv_1$ considérée est k-isomorphe à toute classe ayant les mêmes n_j et un n_i égal à k. On peut donc se restreindre au k-uples $(n_1, \dots, n_k)$ avec tous les n_i entre $\{0, \dots, k\}$.

Dès lors, il n'y a pas plus de $k(k+1)$ classes $\equiv_1$ non k-isomorphes. Un modèle fini de $T_{\sqsubseteq}$ est alors caractérisé par un $k(k+1)$-uple $(m_1, \dots, m_{k(k+1)})$ décrivant le nombre de classes $\equiv_1$ de chaque catégorie. De la même façon que précédemment, nous pouvons restreindre ces m_i à des valeurs inférieures ou égales à k.

On considère alors une classe $\mathcal{C}$ de k-isomorphisme de modèles de $T_{\sqsubseteq}$ ayant un représentant $\mathcal{M}$ de cardinal $n > k^3(k+1)^2$. Alors :

- soit il y a une classe $\equiv_2$ qui au moins k éléments. Alors, pour tout $r \in \mathbb{N}$, si on ajoute r éléments dans cette classe, on obtient une structure k-isomorphe à $\mathcal{M}$ et de cardinal $n + r$. Autrement dit, la classe $\mathcal{C}$ possède un représentant de taille m pour tout $m \geq n$;
- soit aucune classe $\equiv_2$ n'a au moins k éléments, mais il existe une classe $\equiv_1$, notée C_1, pour laquelle on a une classe $\equiv_2$ en plus de k exemplaires. On suppose qu'il s'agit d'une classe $\equiv_2$ de taille i, pour un certain $i \in \{1, \ldots, k-1\}$. Alors, pour tout $r \in \mathbb{N}$, si on ajoute r classes $\equiv_2$ à i éléments dans la classe C_1, on obtient une structure k-isomorphe à $\mathcal{M}$ et de cardinal $n + ir$. On a donc une périodicité des cardinaux des représentants de $\mathcal{C}$ pour une période divisant $\mathrm{ppcm}(i)_{i=1\ldots k-1}$;
- soit aucune classe $\equiv_2$ n'a plus de k éléments, et aucune des classes $\equiv_2$ n'est en plus de k exemplaires dans une même classe $\equiv_1$, mais il existe une classe $\equiv_1$ présente en plus de k exemplaires. Soit j la taille de cette classe. On a $j \in \{1, \ldots, \frac{k(k-1)^2}{2}\}$. Alors, pour tout $r \in \mathbb{N}$, si on ajoute r classes $\equiv_1$ à j éléments dans $\mathcal{M}$, on obtient une structure k-isomorphe à $\mathcal{M}$ et de cardinal $n+jr$. On a donc une périodicité des cardinaux des représentants de $\mathcal{C}$ pour une période divisant $\mathrm{ppcm}(i)_{i=1\ldots\frac{k(k-1)^2}{2}}$.

Quoiqu'il arrive, on a donc une périodicité des cardinaux des représentants de $\mathcal{C}$ pour une période divisant $\mathrm{ppcm}(i)_{i=1\ldots\frac{k(k-1)^2}{2}}$. D'où le résultat suivant.

Theorème 24 (Ash périodique, deux équivalences imbriquées)
Pour tout $k \geq 2$, la fonction

$$n \mapsto N_{T_{\sqsubseteq},k}(n)$$

est ultimement périodique, de période un diviseur de ppcm$(i)_{i=1\ldots\frac{k(k-1)^2}{2}}$.

14.4 Codage d'un graphe dans deux relations d'équivalence et l'égalité

On montre maintenant que si l'on arrive à étendre encore un peu les résultats précédents, on va pourvoir obtenir la conjecture du Spectre dans le cas général. Pour cela, on va décrire une théorie T_G sur le langage $\sigma = \{\equiv_1, \equiv_2, =\}$ telle que les modèles finis de T_G sont exactement les graphes non orientés, via un codage qui augmente polynomialement le cardinal du modèle. C'est-à-dire qu'à tout graphe à n sommets est associé un unique modèle de T_G de cardinal $2n^2$, et tout modèle de T_G est d'un cardinal de la forme $2n^2$, et représente l'image par le codage d'un unique graphe à n sommets. Si on arrive à montrer que le complémentaire d'un spectre pour T_G est un spectre, il est simple de montrer la conjecture du Spectre pour une relation de graphe, et donc pour tout spectre.

Le codage consiste à transformer les arêtes d'un graphes en une chaîne d'équivalences de taille suffisante pour pouvoir distinguer les points de la chaîne des sommets originels. De plus, pour pouvoir maîtriser de manière parfaite le cardinal de la structure transformée, il faut également coder les anti-arêtes, et ce de manière à pouvoir les distinguer des arêtes.

Enfin, pour pouvoir distinguer les sommets du graphe des nouveaux points ajoutés, on duplique chaque sommet du graphe.

14.4.1 Codage d'une arête et d'une anti-arête

Pour coder une arête entre deux points du graphe u et v, on ajoute quatre éléments c_u, c_v, m_u, m_v, et on duplique u et v en deux éléments chacun u_1, u_2, v_1, v_2 respectivement, qui satisfont les relations d'équivalences suivantes :

- u_1 et u_1 sont dans une même classe $\equiv_1$ et dans une même classe $\equiv_2$. On peut donc parler de u (resp. v) pour désigner indifféremment u_1 ou u_2 (resp. v_1 ou v_2),
- u et v sont dans deux classes $\equiv_1$ et $\equiv_2$ distinctes,
- u et c_u sont dans la même classe $\equiv_1$, mais pas dans la même classe $\equiv_2$,
- v et c_v sont dans la même classe $\equiv_1$, mais pas dans la même classe $\equiv_2$,
- c_u et m_u sont dans la même classe $\equiv_2$, mais pas dans la même classe $\equiv_1$,
- c_v et m_v sont dans la même classe $\equiv_2$, mais pas dans la même classe $\equiv_1$,
- m_u et m_v sont dans la même classe $\equiv_1$, mais pas dans la même classe $\equiv_2$,
- les classes $\equiv_1$ et $\equiv_2$ des points milieu m_u et m_v ne contiennent que deux éléments respectivement,
- les classes $\equiv_2$ des points de contact c_u et c_v ne contiennent que deux éléments respectivement.

Ce codage est résumé Figure 14.2.

$$\boxed{\mathbf{u_1, u_2}} \equiv_1 c_u \equiv_2 m_u \equiv_1 m_v \equiv_2 c_v \equiv_1 \boxed{\mathbf{v_1, v_2}}$$

FIG. 14.2 – Codage d'une arête entre deux points du graphe u et v.

Pour coder une absence d'arête entre deux points du graphe u et v, on ajoute quatre éléments c_u, c_v, m_u, m_v qui satisfont les relations d'équivalences suivantes avec les points u et v (on échange les rôles des deux relations d'équivalence) :

- u et v sont dans deux classes $\equiv_1$ et $\equiv_2$ distinctes,
- u et c_u sont dans la même classe $\equiv_2$, mais pas dans la même classe $\equiv_1$,
- v et c_v sont dans la même classe $\equiv_2$, mais pas dans la même classe $\equiv_1$,
- c_u et m_u sont dans la même classe $\equiv_1$, mais pas dans la même classe $\equiv_2$,
- c_v et m_v sont dans la même classe $\equiv_1$, mais pas dans la même classe $\equiv_2$,
- m_u et m_v sont dans la même classe $\equiv_2$, mais pas dans la même classe $\equiv_1$,
- les classes $\equiv_1$ et $\equiv_2$ des points milieu m_u et m_v ne contiennent que deux éléments respectivement,
- les classes $\equiv_1$ des points de contact c_u et c_v ne contiennent que deux éléments respectivement.

Ce codage est résumé Figure 14.3.

$$\boxed{\mathbf{u_1}, \mathbf{u_2}} \equiv_2 c_u \equiv_1 m_u \equiv_2 m_v \equiv_1 c_v \equiv_2 \boxed{\mathbf{v_1}, \mathbf{v_2}}$$

FIG. 14.3 – Codage d'une arête entre deux points du graphe u et v.

On peut donc facilement transformer un graphe non orienté quelconque à n sommets en modèle de $\varphi_{\equiv_1} \wedge \varphi_{\equiv_2}$ à $2n^2$ éléments. Sur l'exemple de la Figure 14.4, on a transformé un graphe non orienté à 3 éléments en une structure à 18 éléments.

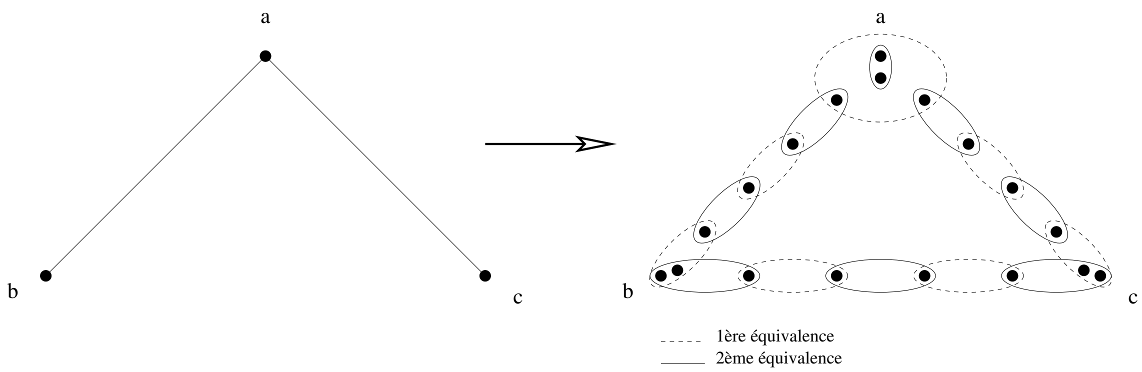


FIG. 14.4 – Exemple de codage d'un graphe à 3 éléments.

Les points originels du graphe sont les seuls à avoir un autre élément qui lui est congru $\equiv_1$ et $\equiv_2$ en même temps. C'est de cette manière qu'on peut les distinguer des autres points de la structure, qui sont tous obligatoirement dans une arête ou une anti-arête.

14.4.2 La théorie T_G

Nous formalisons maintenant à l'aide de formules du langage $\{\equiv_1, \equiv_2, =\}$ la propriété, pour une structure finies sur ce langage, d'être le codage d'un graphe non orienté. Nous pouvons décrire grâce à cela la théorie T_G qui nous intéresse.

On commence par définir la propriété d'être un point originel du graphe :

$$\begin{aligned} Point(x) :=& \exists y[(y \neq x) \wedge (y \equiv_1 x) \wedge (y \equiv_2 x) \wedge \forall z[((z \neq y) \wedge (z \neq x)) \\ & \rightarrow [(z \equiv_1 x) \leftrightarrow (z \not\equiv_2 x)]]]. \end{aligned}$$

Cette formule exprime qu'il existe un élément distinct de x qui est à la fois dans la même classe $\equiv_1$ et dans la même classe $\equiv_2$, et c'est le seul à avoir cette propriété vis-à-vis de x.

On décrit ensuite la propriété d'être un point de contact dans une arête :

$$\begin{aligned} Connect(x) :=& \neg Point(x) \wedge \exists y[Point(y) \wedge (y \equiv_1 x) \wedge (y \not\equiv_2 x)] \wedge \exists y[\neg Point(y) \wedge (y \neq x) \\ & \wedge (y \equiv_2 x) \wedge (y \not\equiv_1 x) \wedge \forall z([(z \neq x) \wedge (z \equiv_2 x)] \rightarrow (z = y))]. \end{aligned}$$

Cette formule exprime qu'il y a un point originel du graphe dans la classe $\equiv_1$ de x, et qu'il y a dans la classe $\equiv_2$ de x un unique point, et ce point n'est pas dans sa classe $\equiv_1$.

De façon analogue, on définit la propriété d'être un point de contact dans une anti-arête :

$$\begin{aligned}NonConnect(x) :=&\neg Point(x) \wedge \exists y[Point(y) \wedge (y \equiv_2 x) \wedge (y \not\equiv_1 x)] \wedge \exists y[\neg Point(y)\\ &\wedge (y \neq x) \wedge (y \equiv_1 x) \wedge (y \not\equiv_2 x) \wedge \forall z([(z \neq x) \wedge (z \equiv_1 x)] \rightarrow (z = y))].\end{aligned}$$

Cette formule exprime qu'il y a un point originel du graphe dans la classe $\equiv_2$ de x, et qu'il y a dans la classe $\equiv_1$ de x un unique point, et ce point n'est pas dans sa classe $\equiv_2$.

Enfin, on décrit la propriété d'être un point milieu (c'est la même propriété pour les arêtes et les anti-arêtes) :

$$\begin{aligned}Milieu(x) :=&\neg Point(x) \wedge \neg Connect(x) \wedge \neg NonConnect(x) \wedge \exists y[(y \neq x) \wedge (y \equiv_2 x)\\ &\wedge (y \not\equiv_1 x) \wedge \forall z([(z \equiv_2 x) \wedge (z \neq x)] \rightarrow (z = y))] \wedge \exists y[(y \neq x) \wedge (y \equiv_1 x)\\ &\wedge (y \not\equiv_2 x) \wedge \forall z([(z \equiv_1 x) \wedge (z \neq x)] \rightarrow (z = y))].\end{aligned}$$

Cette formule exprime que x n'est ni un point originel du graphe, ni un point de contact dans une arête ou une anti-arête, et n'a qu'un seul élément autre que lui-même dans sa classe $\equiv_1$ d'une part, et dans sa classe $\equiv_2$ d'autre part.

On décrit la propriété pour un sexuple d'éléments d'être une arête :

$$\begin{aligned}Arete(u, c_u, m_u, m_v, c_v, v) :=&\ Point(u) \wedge Connect(c_u) \wedge Milieu(m_u) \wedge Milieu(m_v)\\ &\wedge Connect(c_v) \wedge Point(v) \wedge (u \equiv_1 c_u) \wedge (c_u \equiv_2 m_u) \wedge (m_u \equiv_1 m_v) \wedge (m_v \equiv_2 c_v)\\ &\wedge (c_v \equiv_1 v).\end{aligned}$$

On décrit la propriété pour un sexuple d'éléments d'être une anti-arête :

$$\begin{aligned}AntiArete(u, c_u, m_u, m_v, c_v, v) :=&\ Point(u) \wedge NonConnect(c_u) \wedge Milieu(m_u)\\ &\wedge Milieu(m_v) \wedge NonConnect(c_v) \wedge Point(v) \wedge (u \equiv_2 c_u) \wedge (c_u \equiv_1 m_u)\\ &\wedge (m_u \equiv_2 m_v) \wedge (m_v \equiv_1 c_v) \wedge (c_v \equiv_2 v).\end{aligned}$$

On donne maintenant une série d'énoncé qui vont décrire ce qu'est la propriété “être le codage d'un graphe”. Une structure finie $\mathcal{M}$ qui est le codage d'un graphe se caractérise par les propriétés suivantes :

1. Tous les points de $\mathcal{M}$ sont soit des points originels, soit des points de contact dans une arête, soit des points de contact dans une anti-arête, soit des points milieu.
2. Un point originel du graphe n'a pas d'arête ou d'anti-arête qui le relie à lui-même.

3. Tout élément u de $\mathcal{M}$ qui n'est pas un point originel est soit dans une arête soit dans une anti-arête.
4. Deux points originels sont toujours connectés par une arête ou une anti-arête unique.

Ces propriétés sont exprimées par les énoncés suivants :

$$\begin{aligned}
&\Psi_1 := \forall x[Point(x) \vee Connect(x) \vee NonConnect(x) \vee Milieu(x)].\\
&\Psi_2 := \forall x[Point(x) \to \forall y \forall z \forall t \forall u(\neg Arete(x,y,z,t,u,x) \wedge \neg AntiArete(x,y,z,t,u,x)].\\
&\Psi_3 := \forall x(Connect(x) \to \exists y \exists z \exists t \exists u \exists v[Arete(y,x,z,t,u,v)]).\\
&\Psi_4 := \forall x(NonConnect(x) \to \exists y \exists z \exists t \exists u \exists v[AntiArete(y,x,z,t,u,v)]).\\
&\Psi_5 := \forall x(Milieu(x) \to \exists y \exists z \exists t \exists u \exists v[Arete(y,z,x,t,u,v) \vee AntiArete(y,z,x,t,u,v)]).\\
&\Psi_6 := \forall x \forall y[(Point(x) \wedge Point(y)) \to ((x \equiv_1 y) \vee \exists z \exists t \exists u \exists v[(Arete(x,z,t,u,v,y)\\
&\qquad \wedge \forall z' \forall t' \forall u' \forall v'[\neg AntiArete(x,z',t',u',v',y) \wedge (Arete(x,z',t',u',v',y)\\
&\qquad \to ((z'=z) \wedge (t'=t) \wedge (u'=u) \wedge (v'=v)))]) \vee (AntiArete(x,z,t,u,v,y)\\
&\qquad \wedge \forall z' \forall t' \forall u' \forall v'[\neg Arete(x,z',t',u',v',y) \wedge (AntiArete(x,z',t',u',v',y)\\
&\qquad \to ((z'=z) \wedge (t'=t) \wedge (u'=u) \wedge (v'=v)))]])].
\end{aligned}$$

On pose maintenant

$$T_G = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Psi_6\}.$$

14.4.3 Correspondance entre les graphes non orientés et les modèles de T_G

On montre maintenant par récurrence sur n que :
- les modèles finis de T_G sont nécessairement de cardinal $2n^2$;
- à chaque modèle fini $\mathcal{M}$ de T_G de cardinal $2n^2$, on peut faire correspondre un graphe non orienté $G_{\mathcal{M}}$ à n sommets tel que $\mathcal{M}$ soit le codage de $G_{\mathcal{M}}$.

Pour $n = 1$: on essaie de construire un modèle de T_G de cardinal le plus petit possible. Soit $\mathcal{M}$ un modèle de T_G de cardinal minimal. Alors il y a au moins un élément x dans $\mathcal{M}$.
- soit x vérifie $Point(x)$, et alors il a au moins un élément y différent de x qui est $\equiv_1$ et $\equiv_2$ à x, et satisfait $Point(y)$;
- soit x vérifie $Connect(x)$ et alors il y a au moins une arête ou une anti-arête contenant x, donc au moins 7 autres éléments : ce n'est pas minimal ;
- soit x vérifie $NonConnect(x)$ et alors il y a au moins une arête ou une anti-arête contenant x, donc au moins 7 autres éléments : ce n'est pas minimal ;
- soit x vérifie $Milieu(x)$ et alors il y a au moins une arête ou une anti-arête contenant x, donc au moins 7 autres éléments : ce n'est pas minimal.

Un modèle de taille minimale de T_G est alors obligatoirement composé de deux éléments distincts qui sont dans une même classe $\equiv_1$ et une même classe $\equiv_2$. C'est de manière évidente le codage d'un graphe constitué d'un seul sommet.

Montrons que l'hypothèse au rang n implique l'hypothèse au rang $n+1$: on suppose que les cardinaux des modèles de T_G inférieurs à $2n^2$ sont exactement les $2i^2$, pour $i \geq n$, et que chaque modèle de T_G de cardinal $\leq 2n^2$ est le codage d'un graphe non orienté à n sommets.
On considère maintenant un modèle $\mathcal{M}$ de T_G de cardinal $n < m \leq 2(n+1)^2$. Montrons que $m = 2(n+1)^2$ et que $\mathcal{M}$ est le codage d'un graphe à $n+1$ sommets. Soit $2k$ le nombre d'éléments de $\mathcal{M}$ qui satisfont $Point(.)$. Alors il y a au moins soit une chaîne $Arete$, soit une chaîne $AntiArete$ entre tout couple de bipoints qui satisfont $Point(.)$. C'est-à-dire qu'il y a au moins $2k(k-1)$ autres éléments dans $\mathcal{M}$, ne satisfaisant pas $Point(.)$. Donc il y a au moins $2k + 2k(k-1) = 2k^2$ éléments dans $\mathcal{M}$. Si $k \leq n$, alors on a $2k^2 \leq 2n^2$, donc il y a d'autres éléments dans le domaine de $\mathcal{M}$. Soit x un élément qui n'a pas encore été compté. L'élément x ne peut pas satisfaire $Point(x)$ par définition de k. D'autre part, si x est un point satisfaisant $Connect$, $PasConnect$ ou $Milieu$, alors il est nécessairement dans une arête ou une antiarête. Or, une arête ou une anti-arête est nécéssairement bordée par deux éléments satisfaisant $Point$. Si ces deux points sont déja comptés dans les $2k$ éléments de $\mathcal{M}$ satisfaisant $Point$, alors le point x est déjà compté dans les arêtes et les anti-arêtes reliant les point satisfaisant $Point$: c'est donc impossible. Donc $k \leq n+1$, et par conséquent $m \leq 2(n+1)^2$, ce qui implique $m = 2(n+1)^2$.

D'autre part, les axiomes Ψ_1 à Ψ_6 impliquent nécessairement que le modèle $\mathcal{M}$ est le codage d'un graphe à $n+1$ sommets. En effet, il suffit de transformer les doublets d'éléments satisfaisant $Point$ en sommets, et chaque quadruple d'éléments de connection ou milieu formant une arête entre deux points en arête. On élimine également les quadruples d'éléments formant le cœur d'une anti-arête.

14.5 Conclusion

Le Théorème 23 implique que la conjecture du spectre est vraie pour les énoncés faisant intervenir une seule relation d'équivalence. Le complémentaire du spectre d'un tel énoncé est une union de la forme :

$$\bigcup_{j=1}^{N_{\sigma,k}} (Sp(\theta_j) \cap (a_j + b_j\mathbb{N})),$$

où $Sp(\theta_j)$ est un spectre de plusieurs relations d'équivalence. La Proposition 36 nous indique de plus que les $a_j + b_j\mathbb{N}$ peuvent également être vus comme des spectres d'une relation d'équivalence. On peut donc préciser la conjecture du Spectre dans ce cas : le complémentaire du spctre d'un énoncé faisant intervenir une seule relation d'équivalence est un spectre d'énoncé faisant intervenir plusieurs relations d'équivalence.

Nous laissons ouverte la question concernant le comportement de la fonction de Ash pour la théorie T_G, et les théories intermédiaires entre $T_{\equiv}$ et T_G.

Conclusion et perspectives de la Partie III

Conclusion

Cette troisième partie présente quelques solutions partielles au problème NE $\stackrel{?}{=}$ CoNE. Aucune conclusion définitive n'a été dégagée, mais nous avons, en quelque sorte, réglé le sort des cas simples, à savoir en l'occurrence le cas d'une profondeur de quantification égale à 2, le cas d'un graphe de fonction, le cas d'une relation d'équivalence. Pour chacune des pistes étudiées, il nous a semblé très difficile de continuer à progresser avec les mêmes techniques que celles que nous avons employées aux Chapitres 12, 13 et 14. Si nous analysons les raisons, parfois assez subjectives, qui nous ont empêché d'aller plus loin, nous serons peut-être en mesure de déterminer une frontière entre d'une part, des cas "faciles" pour lesquels il est relativement aisé de montrer qu'ils satisfont une des conjectures de Ash, mais qui ne nous permettent pas de conclure dans le cas général, et d'autre part, des cas "difficiles" dont la résolution apporterait une réponse à la conjecture du Spectre dans le cas général.

- Dans le Chapitre 12, nous évoquons la possibilité que la difficulté de la généralisation aux profondeurs de quantification supérieures à 2 du cas $k = 2$ provienne du caractère indécidable de certaines théories à profondeur de quantification 3, tandis que FO_2 est décidable.
- Dans le Chapitre 13, les méthodes employées ne peuvent pas s'appliquer directement pour régler le cas des graphes de degré sortant 2, car les composantes connexes deviennent très compliquées.
- Dans le Chapitre 14, les méthodes employées ne peuvent pas s'appliquer directement au cas des relations d'équivalence non imbriquées (et plus particulièrement celles qui codent un graphe non orienté), car les classes d'équivalence deviennent très difficile à décrire.

Remarquons enfin que, dans tous les cas que nous avons présenté, la forme des spectres est particulière : il s'agit d'unions d'ensembles arithmétiques. Il nous est donc possible de conclure directement sur la forme des complémentaires de ces spectres (ce sont des unions d'ensembles arithmétiques également, et donc des spectres).

Perspectives

Dès lors, nous pouvons dégager les perspectives suivantes :

- on peut essayer de trouver d'autres théories pour lesquelles la conjecture de Ash (constante, périodique, ou ultrafaible) implique la conjecture du Spectre dans le cas général.
- on peut essayer de formaliser le lien implicite entre décidabilité, expressivité des théories et la satisfaction des différentes conjectures de Ash ;
- au lieu de s'intéresser à une théorie, on peut généraliser à toute une classe de structures satifaisant une propriété qui n'est pas forcément définissable au premier ordre.

D'autres théories pour lesquelles la conjecture du Spectre est aussi difficile que le cas général

On peut essayer de faire une liste de théories pour lesquelles la conjecture du Spectre est aussi difficile que dans le cas général. Par exemple, des théories pour lesquelles on a un moyen de rembourrer polynomialement un graphe fini, ou un modèle de la théorie de deux fonctions, dans un modèle de cette théorie. Ou encore, des théories pour lesquelles on a prouvé que tout spectre peut se mettre sous la forme d'un spectre d'énoncé impliquant cette théorie, éventuellement par le biais d'une transformation polynomiale. Remarquons qu'il n'est pas nécessaire que la transformation soit polynomiale, il suffit qu'elle soit dans la classe NE, mais les exemples connus semblent indiquer qu'il est assez naturel de réaliser une transformation polynomiale. Une fois établie une liste de théories vérifiant ces propriétés, on peut essayer de progresser vers ces théories en étudiant des théories plus contraintes, et en essayant d'analyser les raisons pour lesquelles les méthodes employées ne fonctionnent plus dans le cas des théories de la liste. Ou bien, qui sait, trouver une théorie $\mathcal{T}$ pour laquelle la conjecture de Ash (constante, périodique, ou ultrafaible) implique la conjecture du Spectre dans le cas général et pour laquelle on sait résoudre la conjecture de Ash (constante, périodique, ou ultrafaible)...

Dans la thèse, nous avons donné plusieurs exemple de telles théories, comme la théorie d'un graphe non orienté, la théorie de deux fonctions, la théorie de deux équivalences. Il en existe d'autres, comme par exemple la théorie de plusieurs ordres totaux ([DLS98]). La théorie d'un seul ordre total est catégorique, c'est-à-dire que pour un cardinal n donné, il existe au plus un modèle de cet ordre, à isomorphisme près. Comme de plus pour tout cardinal n, il existe un modèle de la théorie d'un seul ordre total, cela signifie qu'il y a exactement un modèle de cette théorie de cardinal n, à isomorphisme près, pour tout n. Il est donc facile de remarquer que la fonction de Ash est constante de valeur 1, et donc la conjecture du Spectre est vraie pour un seul ordre total. En revanche, pour deux ordres totaux, il n'y a plus catégoricité, et on ne peut conclure aussi facilement.

Décidabilité et expressivité

Nous pouvons également nous interroger sur le lien éventuel entre la décidabilité d'une théorie et le comportement de la fonction de comptage des classes de k-isomorphisme de ses modèles finis. Par exemple, pouvons-nous exhiber :

- une théorie décidable qui satisfait la conjecture de Ash constante pour tout k (la réponse est oui, la théorie d'un ordre total, voir le paragraphe ci-dessus) ;

- une théorie indécidable qui satisfait la conjecture de Ash constante pour tout k ;
- une théorie décidable qui satisfait la conjecture de Ash périodique mais non constante pour tout k (la réponse est oui, la théorie d'un graphe de bijection) ;
- une théorie indécidable qui satisfait la conjecture de Ash périodique mais non constante pour tout k ;
- une théorie décidable qui satisfait la conjecture de Ash ultrafaible mais non périodique ni constante pour tout k ;
- une théorie indécidable qui satisfait la conjecture de Ash ultrafaible mais non périodique ni constante pour tout k ;

ou bien donner des relations d'implication entre la décidabilité de la théorie et la non-satisfaction d'une de ces conjectures, voire même une caractérisation des théories décidables par ce biais ?

On peut notamment se poser la question de l'existence d'une théorie naturelle, ayant de bonnes propriétés (par exemple un modèle pour tout cardinal), pour laquelle la fonction de Ash n'est pas périodique.

Une autre notion qui intervient énormément dans l'étude de la conjecture de Ash ultrafaible via les restrictions sémantiques est la notion d'expressivité d'une théorie. En effet, plus la théorie est expressive, et plus la résolution de la conjecture de Ash pour cette théorie nous donne des indications sur le cas général. Par exemple, n'importe quel spectre peut être interprété, via une transformation polynomiale, comme un spectre d'un énoncé de deux fonctions, ou bien comme un spectre d'une relation de graphe. La question est de savoir s'il y a un lien entre l'expressivité d'une théorie et la satisfaction des différentes formes de la conjecture de Ash. Par exemple, les théories très expressives satisfont-elles les formes les plus faibles de la conjecture de Ash, tandis que les théories peu expressives satisfont les formes les plus fortes, ou bien le contraire ? Peut-on hiérarchiser les théories en fonction de ce critère ? Nous n'avons pas d'intuition concernant ce point.

Passer des théories à des classes de structures non définissables au premier ordre

Au lieu de considérer des théories et leurs modèles, on peut également s'intéresser à des classes de structures plus générales, à savoir par exemple la classe des σ-structures finies, sur un langage σ constitué d'une relation binaire et l'égalité, qui sont des arbres. La propriété "être un arbre" n'est pas définissable par un énoncé du premier ordre. On peut considérer les classes de k-isomorphisme de structures qui sont des arbres. On compte alors le nombre d'arbres de cardinal n qui ne sont pas k-isomorphes, et on peut essayer de déterminer le comportement de cette fonction de Ash généralisée. Dans le cas des arbres, on peut par exemple deviner que cette fonction sera périodique, par des arguments de découpage semblable à ceux utilisés au Chapitre 13. On peut se demander dans quelle mesure le comportement de cette fonction, relativisée à une classe de structure et non plus à une théorie, peut nous renseigner sur le complémentaire du spectre d'un énoncé sur le langage σ. En fait, il s'agit de nous renseigner sur des ensembles qui ne sont pas vraiment des spectres, mais qui leur ressemblent beaucoup. Dans notre exemple, le "spectre" d'un énoncé φ est l'ensemble des cardinaux des arbres qui satisfont φ. Il est vraisemblable que le complémentaire d'un tel ensemble est un ensemble de même type, étant donnée la périodicité de la fonction de comptage des classes de k-isomorphisme d'arbres. Mais tout cela reste à formaliser.

Conclusion générale

Cette thèse est née de l'amusement commun suscité chez Denis Richard et moi-même par les destinées. Il n'existait qu'un seul ouvrage de référence sur les destinées, à savoir la thèse de leur créateur, Francis Nézondet. Comme une seule thèse n'était pas suffisante pour assouvir la curiosité de Denis vis-à-vis de ces objets, j'ai commencé cette thèse avec l'ambition de mettre au clair ce que l'on pouvait faire et dire avec les destinées.

Dans un premier temps, j'ai repris les travaux de Francis Nézondet sur l'algorithme de décision des énoncés de profondeur de quantification donnée, et parallèlement j'ai commencé à réaliser un catalogue de 2-destinées et de 3-destinées, de diverses structures arithmétiques, décidables ou non. Parmi celles-ci, figurait la resistante 3-destinée de $\langle \mathbb{N}, S, \perp \rangle$. Tout ceci était fort distrayant, et instructif sur le pouvoir d'expression de ces structures.

Malheureusement, il était difficile d'en tirer des résultats généraux sur les destinées. Nous avons donc commencé à nous intéresser aux cas où l'on pouvait dire quelque chose sur les p-destinées avec p plus grand (c'est-à-dire, p suffisamment grand pour qu'on ne puisse plus dessiner une p-destinée sur un tableau...). Grâce à Maxim Vsemirnov, nous avons pu produire l'exemple des p-destinées réduites de $\langle \mathbb{N}, \leq \rangle$.

Vint ensuite le temps de la généralisation : pouvait-on obtenir des résultats analogues sur d'autres structures ? Suivant la piste de la structure $\langle \mathbb{N}, | \rangle$, et grâce au coup de pouce de Nicole Schweikardt, nous avons regardé ce qu'il en était des structures H-bornées. C'est alors que j'ai pu, grâce à une idée de Yuri Mattiassevitch, établir l'inclusion stricte de la classe des structures H-bornées dans la classe des structures à destinées récursives. Nous avons poursuivi ces travaux jusqu'aux problèmes ouverts présentés dans la Conclusion de la Partie II : existe-t-il des structures décidables qui ne sont pas à destinées récursives ?

Malika More a eu ensuite la bonne idée de présenter les destinées à Etienne Grandjean. Il nous a abondamment documentées sur la conjecture de Ash. Nous avons dès lors trouvé un lien très fort entre les destinées et la conjecture de Ash, par le biais des classes de k-isomorphisme. L'application des destinées à ce problème de théorie des modèles finis a redonné un second souffle à cette thèse, comme peut en témoigner le volume de la Partie III. Nous n'avons pas, là encore, et loin s'en faut, fait le tour de la question.

En résumé, les destinées sont passionnantes, à la fois en tant qu'objet d'étude et en tant qu'outil de réflexion sur les problèmes logiques faisant intervenir la profondeur de quantification des énoncés.

Bibliographie

[Ajt83] Ajtai (M.). – Σ_1^1 Formulae on Finite Structures. *Annals of Pure and Applied Logic*, 1983, vol. 24, pp. 1–48.

[Ash94] Ash (C. J.). – A Conjecture Concerning the Spectrum of a Sentence. *Mathematical Logic Quartely*, 1994, vol. 40, pp. 393–397.

[Ass55] Asser (G.). – Das Repräsentenproblem in Prädikatenkalkül der ersten Stufe mit Identität. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 1955, vol. 1, pp. 252–263.

[BDG90] Balcázar (J. L.), Díaz (J.) et Gabarró (J.). – *Structural Complexity II.* – Springer-Verlag, 1990.

[Ben62] Bennett (J). – *On spectra.* – Thèse de Doctorat, Princeton University, 1962.

[Ber85] Berge (C.). – *Graphes.* – Gauthier-Villars, 1985.

[BGS75] Baker (T.), Gill (J.) et Solovay (R.). – Relativisations of the NP $\stackrel{?}{=}$ P Question. *SIAM Journal on Computing*, 1975, vol. 4, pp. 431–442.

[Ceg01] Cegielski (P.). – Destinies and Decidability. *In : Journées des Arithmétiques Faibles 20.* – `http ://www.univ-paris12.fr/lacl/jaf/paper/cegielski.ps`, 2001.

[Cha00] Chateau (A.). – Les théories du successeur et de la coprimarité. – Mémoire de DEA, Université Paris 7, 2000.

[Cha01] Chateau (A.). – Décision des théories à nombre borné de variables sur les langages relationnels finis : un algorithme utilisant les destinées. *In : CNRIUT 2001.* – Publications de l'université de Saint-Etienne, 2001. pp. 497–506.

[Cha02] Chateau (A.). – Automatisation de la décision de formules logiques via la construction des destinées. *In : CNRIUT 2002.* – Publications de l'IUT du Creusot (Université de Bourgogne), 2002. pp. 85–92.

[CL94] Cori (R.) et Lascar (D.). – *Logique mathématique, cours et exercices.* – Masson, 1994.

[CMP00] Chailloux (E.), Manoury (P.) et Pagano (B.). – *Développement d'application avec Objective Caml.* – O'Reilly, 2000.

[CP] Cunningham Project ( ). – `http ://www.ceria.purdue.edu/homes/ssw/cun/`.

[DFL98] Durand (A.), Fagin (R.) et Loescher (B.). – Spectra with Only Unary Function Symbols. *Computer Science Logic, LNCS*, 1998, vol. 1414, pp. 189–202.

[DLS98] Durand (A.), Lautemann (C.) et Schwentick (T.). – Subclasses of Binary NP. *Journal of Logic and Computation*, 1998, vol. 8, pp. 189–207.

[Dub95] Dubhashi (D. P.). – *Complexity of Logical Theories.* – Departement of Computer Science, University of Aarhus, Denmark, BRICS (Basic Research In Computer Science), 1995. Rapport technique.

[EF99] Ebbinghaus (H.-D.) et Flum (J.). – *Finite Model Theory.* – Springer, 1999.

[EFT94] Ebbinghaus (H.-D.), Flum (J.) et Thomas (W.). – *Mathematical Logic.* – Springer, 1994.

[End72] Enderton (H. B.). – *A Mathematical Introduction to Logic.* – Academic Press, 1972.

[Fag74] Fagin (R.). – Generalized First-Order Spectra and Polynomial-Time Recognizable Sets. *Complexity of Computation, SIAM-AMS Proceedings*, 1974, vol. 7, pp. 43–73.

[Fag75] Fagin (R.). – A Spectrum Hierarchy. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 1975, vol. 21, pp. 123–134.

[Fag90] Fagin (R.). – Finite-Model Theory - a Personal Perspective. *In : ICDT'90, Third International Conference on Database Theory, Paris, France, December 12-14, 1990, Proceedings*, éd. par Abiteboul (S.) et Kanellakis (P. C.). – Springer, 1990. pp. 3–24.

[FR79] Ferrante (J.) et Rackoff (C. W.). – The Computational Complexity of Logical Theories. *In : Lecture Notes in Mathematics.* – Springer-Verlag, 1979.

[Fra72] Fraïssé (R.). – *Cours de logique mathématique.* – Gauthier-Villars, 1972.

[GM79] Gondran (M.) et Minoux (M.). – *Graphes et algorithmes.* – Eyrolles, 1979.

[Gol84] Goldfarb (W. D.). – The unsolvability of the Gödel class with identity. *Journal of Symbolic Logic*, 1984, vol. 49, pp. 1237–1252.

[Grz53] Grzegorczyk (A.). – Some Classes of Recursive Functions. *Rosprawy Mat.*, 1953, vol. 4, pp. 1–46.

[Gui96] Guillaume (M.). – Nouvelles récentes sur les ensembles Y et X. *Actes du LLAIC (Laboratoire de Logique, Algorithmique et Informatique de Clermont)*, 1996, vol. VI.

[Gui01] Guillaume (M.). – A Short Report on the Present State of the Research in the LLAIC on the 3-transversal of Destinies in $(S, \perp)$. *Actes du LLAIC (Laboratoire de Logique, Algorithmique et Informatique de Clermont)*, 2001, vol. VII.

[HIS85] Hartmanis (J.), Immermann (N.) et Sewelson (J.). – Sparse Sets in NP-P-EXPTIME vs. NEXPTIME. *Information and Control*, 1985, vol. 65, pp. 159–181.

[HO02] Hemaspaandra (L. A.) et Ogihara (M.). – *The Complexity Theory Companion.* – Springer, 2002.

[Hod93] Hodges (W.). – Model Theory. *In : Encyclopedia of Mathematics and its Applications.* – Cambridge University Press, 1993.

[Imm88] Immerman (N.). – Nondeterministic Space is Closed under Complementation. *SIAM Journal of Computation*, 1988, vol. 17, pp. 935–938.

[JS74] Jones (N. D.) et Selman (A. L.). – Turing Machines and the Spectra of First-Order Formulas. *Journal of Symbolic Logic*, 1974, vol. 39, pp. 139–150.

[LB98] Le Bars (J.-M.). – *Probabilités asymptotiques et pouvoir d'expression des fragments de la logique du second ordre.* – Thèse de Doctorat, Université de Caen, 1998.

[LW99] Leroy (X.) et Weiss (P.). – *Le langage Caml.* – Dunod, 1999.

[Mat95] Matiiassevitch (Y.). – *Le dixième problème de Hilbert. Son indécidabilité.* – Masson, 1995.

[Mau94] Maurin (F.). – *Complexité de l'addition ordinale.* – Thèse de Doctorat, Université de Caen, 1994.

[Mic81] Michel (P.). – Borne supérieure de la complexité de la théorie de $\mathbb{N}$ muni de la relation de divisibilité. *In : Lecture Notes in Mathematics.* – Springer-Verlag, 1981.

[Mor75] Mortimer (M.). – On Languages with Two Variables. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 1975, vol. 21, pp. 135–140.

[Mos56] Mostowski (A.). – Concerning a Problem of H.Scholz. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 1956, vol. 12, pp. 210–214.

[Néz97] Nézondet (F.). – *p-destinées et applications à la théorie du successeur et de la coprimarité sur les entiers.* – Thèse de Doctorat, Université d'Auvergne, 1997.

[Pap94] Papadimitriou (C. H.). – *Computationnal Complexity.* – Addison-Wesley, 1994.

[Poi85] Poizat (B.). – *Cours de théorie des modèles.* – Nur al-Mantiq wal-Ma'Rifah, 1985.

[Sch52] Scholz (H.). – Ein ungelöstes Problem in der symbolischen Logik. *Journal of Symbolic Logic*, 1952, vol. 17, p. 160.

[Sch96] Schwentick (T.). – On Winning Ehrenfeucht Games and Monadic NP. *Annals of Pure and Applied Logic*, 1996, vol. 79, pp. 61–92.

[Sze87] Szelepcsényi (R.). – The Method of Forcing for Non-Deterministic Automata. *Bulletin of EATCS*, 1987, vol. 33, pp. 96–100.

[VL90] Van Leuwen (J.) (édité par). – *Handbook of Theoretical Computer Science, Algorithms and Complexity.* – Elsevier, 1990.

Index