N° d'ordre : 207
N° bibliothèque : 01ENSL0207

ÉCOLE NORMALE SUPÉRIEURE DE LYON
Laboratoire de l'informatique du parallélisme

THÈSE
pour obtenir le grade de
Docteur de l'École normale supérieure de Lyon
Spécialité : Informatique
au titre de l'École doctorale de mathématiques et informatique fondamentale

présentée et soutenue publiquement le 7 décembre 2001
par M. **Hervé FOURNIER**

Complexité et expressibilité sur les réels

Devant la commission d'examen formée de :
M. Michel Cosnard (président du jury)
M. Etienne Grandjean (rapporteur)
M. Dimitri Grigoriev
M. Stéphane Grumbach
M. Pascal Koiran (directeur de thèse)
M. Christian Michaux (rapporteur)

Remerciements

Ce fut un réel plaisir de travailler sous la direction de Pascal Koiran. Pour le précieux enseignement qu'il m'a prodigué, son enthousiasme, sa disponibilité et ses nombreux conseils, je le remercie.

Je remercie Etienne Grandjean et Christian Michaux d'avoir accepté de rapporter cette thèse, ainsi que Michel Cosnard, Dima Grigoriev et Stéphane Grumbach pour leur présence au sein du jury.

Pour les fructueuses discussions je remercie les membres de l'IGD participant au groupe de travail Complexité algébrique, en particulier Bruno Poizat, Olivier Chapuis, Ménard Bourgade et Guillaume Malod.

Je tiens à remercier les membres de l'équipe MC2 où règne une joyeuse ambiance. Jacques Mazoyer y est pour beaucoup et je le remercie. Je remercie également Natacha Portier et Marianne Delorme pour leurs conseils, concernant la préparation des exposés par exemple. Je remercie Jean-Christophe Dubacq, Bruno Martin et Nicolas Ollinger avec qui j'ai partagé pauses café et coins de tableaux au cours de ces années.

Table des matières

Introduction

Dans divers domaines de l'informatique (par exemple les bases de données contraintes) et des mathématiques (par exemple l'analyse numérique), on est amené à étudier des questions de nature algorithmique sur une structure qui ne l'est pas : le corps des nombres réels. Nous abordons dans ce document deux types de questions de cette nature. Les premières concernent la complexité sur les nombres réels, et les secondes la notion de modèle fini plongé dans les réels et les questions d'expressibilité qui s'y rapportent. Il ne faudrait cependant pas croire que les spécificités des problèmes rencontrés proviennent du caractère non-algorithmique des réels. Pour preuve, presque toutes les questions soulevées, ainsi que les techniques mises en œuvre pour les résoudre, restent pertinentes si on considère les rationnels (ou les nombres algébriques, ou encore la clôture réelle des rationnels selon le contexte) à la place des réels. Ce qui fait la différence avec le cas classique (la complexité booléenne pour la première partie, la théorie des modèles finis pour la seconde) c'est la présence des opérations d'addition et éventuellement de produit sur cette structure ce qui revient bien, dans le cadre de la complexité, à considérer un coût unitaire pour les opérations algébriques.

Complexité. L'introduction d'un modèle de calcul sur les réels par Blum Shub et Smale [BSS89], le modèle BSS, a fourni un cadre formel pour se poser des questions naturelles. Des questions de calculabilité : l'ensemble de Mandelbrot est-il décidable ? Et des questions de complexité : existe-t-il un algorithme en temps polynomial décidant si un polynôme réel en plusieurs variables, donné par la suite de ses coefficients, a un zéro ? C'est le coût algébrique qui est considéré ici : on effectue en un seul pas de calcul l'addition ou le produit de deux nombres réels avec une précision infinie. Le but est de capturer l'essence même de la complexité de certains problèmes. Cette approche n'est pas nouvelle : la question de la complexité de l'évaluation d'un polynôme en un point, par exemple, est très ancienne – on fait remonter la complexité algébrique au début des années cinquante [Str90, BCS97]. L'approche BSS se généralise à n'importe quelle structure : c'est le modèle de calcul de Poizat [Poi95]. En limitant les ressources disponibles pour un calcul, comme le temps ou l'utilisation de fonctions conseil, on définit l'analogue des classes de complexité usuelles. Cela débouche sur une théorie de la com-

plexité algébrique structurelle. Notons qu'il existe une autre approche de la complexité algébrique structurelle, due à Valiant ; consulter [Bür00] pour les développements récents dans ce domaine, en particulier les liens qu'entretient cette théorie avec la complexité booléenne et le modèle BSS.

Comme dans le cas booléen, l'attention se porte en particulier sur la puissance du non-déterminisme. Et comme dans le cas booléen encore, la question $\mathrm{P} = \mathrm{NP}$ semble difficile pour de nombreuses structures naturelles telles que les réels avec addition et ordre, le corps des réels ou encore le corps des complexes. À défaut d'être en mesure de résoudre ces problèmes, on essaie de les relier entre eux : ce sont les théorèmes de transfert. Un exemple percutant de tel résultat est le théorème de Blum, Cucker, Shub et Smale : la réponse à la question $\mathrm{P} = \mathrm{NP}$ est la même sur tous les corps algébriquement clos de caractéristique nulle. D'autres résultats sont obtenus par la technique des parties booléennes. Citons par exemple deux résultats dus à Pascal Koiran : si $\mathrm{P} = \mathrm{NP}$ sur les réels avec addition et ordre, ou sur le corps des complexes, alors $\mathrm{P/poly} = \mathrm{NP/poly}$. Notons que cette technique ne permet pas d'obtenir des transferts dans l'autre sens. Obtenir de tels résultats est l'objet principal de nos travaux, qui sont menés sur les réels avec addition (et éventuellement ordre).

On expose au chapitre 1 la notion de calcul sur une structure arbitraire. Les classes de complexité y sont définies, ainsi que les notions de réduction et de complétude. On rappelle aussi deux méthodes pour obtenir des théorèmes de transfert : les parties booléennes, permettant d'établir des transferts entre une structure arbitraire et le cas classique, et la méthode des structures élémentairement équivalentes. On fait le point au chapitre 2 sur les spécificités du calcul sur les réels avec addition. On explique pourquoi les arrangements d'hyperplans interviennent naturellement dans ce cadre. Au chapitre 3, on rappelle en détails deux constructions géométriques, l'une de Meyer auf der Heide et l'autre de Meiser, pour localiser un point dans un arrangement d'hyperplans par un arbre décision linéaire de petite profondeur. La taille des coefficients des hyperplans tests de ces arbres est évaluée. Ces résultats nous permettent d'établir au chapitre 4 des théorèmes de transfert sur les réels avec addition et ordre : par exemple, il est montré que le problème $\mathrm{P} = \mathrm{NP}$ sur cette structure est équivalent au problème booléen $\mathrm{P/poly} = \mathrm{NP/poly}$. Ce résultat repose sur un algorithme polynomial de localisation de point dans un arrangement d'hyperplans réels à l'aide d'un oracle booléen NP. La structure des réels avec addition uniquement est étudiée au chapitre 5. La situation sur cette structure est bien différente puisqu'on a sur cette structure la séparation inconditionnelle $\mathrm{P} \neq \mathrm{NP}$. La séparation $\mathrm{NP} \neq \Sigma^2$ est aussi établie. Cependant, on montre que séparer les niveaux supérieurs de la hiérarchie est aussi dur que certains problèmes de même nature dans la hiérarchie polynomiale booléenne. Enfin, on étudie au chapitre 6 l'existence possible de problèmes creux NP-complets sur les réels avec addition (et éventuellement ordre), dans le but d'établir des versions réelles des théorèmes de Mahaney et de Karp-Lipton.

Expressibilité. Dans la seconde partie le point de vue est différent. On s'intéresse à un modèle fini plongé dans une structure infinie comme les réels avec addition et ordre, le corps ordonné des réels ou encore le corps des complexes. Les questions concernent

ce qu'on peut exprimer à propos du modèle fini avec la logique du premier ordre. On rencontre cette situation dans les bases de données contraintes [KLP00] – notons que c'est aussi la situation décrite dans l'article de Grädel et Gurevich [GG98]. Quelques-uns des résultats fondamentaux relatifs au gain d'expressivité obtenu au premier ordre à l'aide d'une telle structure sont rappelés dans le chapitre 7.

Le chapitre 8 est consacré à une variante de ce problème : on étudie l'influence de la structure sous-jacente sur le rang de quantification nécessaire pour exprimer certaines propriétés du modèle fini. Par exemple, on montre que l'addition – la structure d'espace vectoriel – permet de diminuer de manière exponentielle des requêtes de cardinalité (un ensemble, interprété par un prédicat unaire, a-t-il au moins m éléments ?) ; par contre, ajouter en plus le produit sur un corps algébriquement clos ne mène à aucun gain supplémentaire.

Enfin, on étudie dans le chapitre 9 la notion d'effondrement actif-naturel : si les variables quantifiées ne parcourent que le modèle fini plutôt que le domaine tout entier de la structure sous-jacente, a-t-on une perte d'expressivité ? Sur les structures fortement minimales et les corps différentiellement clos, on montre que ce n'est pas le cas par une méthode de localisation qui est à rapprocher de celle utilisée dans la première partie pour établir des théorèmes de transfert. La propriété d'effondrement actif-naturel est ensuite comparée à celle d'élimination uniforme des quantificateurs introduite par Basu [Bas99].

1. Calcul et complexité sur une structure arbitraire

Une structure, c'est un ensemble muni d'opérations. Nous présentons dans ce chapitre des modèles de calcul sur une structure arbitraire à la manière de Poizat [Poi95, Goo94]. En limitant la quantité de ressources (telles que le temps, le non-déterminisme ou l'utilisation de fonctions conseils) disponible pour faire les calculs, on définit l'analogue des classes de complexité usuelles sur n'importe quelle structure. Comme dans le cas classique, notre attention se porte en particulier sur les classes P et NP. Les notions de réduction et de complétude se prolongent naturellement dans ce contexte, et permettent d'identifier les problèmes les plus durs au sein d'une classe de complexité. Enfin, nous exposons deux méthodes permettant d'établir des théorèmes de transfert, c'est-à-dire de relier des questions de complexité structurelle sur une première structure à ces mêmes questions sur une autre structure.

1.1 Structures

Une structure $\mathcal{S}$, c'est un ensemble M, le domaine de la structure, muni d'un nombre fini de fonctions $f_1, \ldots, f_s$. Chaque fonction f_i est une application de M^{r_i} dans M, où $r_i \in \mathbb{N}$ est l'arité de f_i. Les fonctions d'arité nulle sont appelées des constantes. On supposera que toute structure est munie de deux constantes distinctes 0 et 1, ainsi que de l'égalité. On ne distingue pas les fonctions des relations même si on s'autorise à parler de relations : les relations sont vues comme des fonctions à valeurs dans $\{0, 1\}$.

Exemples de base. Nous adoptons les notations suivantes pour les structures qui nous intéressent le plus : la structure standard $\mathbb{Z}_2 = (\mathbb{Z}/2\mathbb{Z}, 0, 1, +, =)$, les réels avec addition et égalité $\mathbb{R}_{vs} = (\mathbb{R}, 0, 1, +, -, =)$, les réels avec addition et ordre $\mathbb{R}_{ovs} = (\mathbb{R}, 0, 1, +, -, <)$, le corps ordonné des réels $\mathbb{R} = (\mathbb{R}, 0, 1, +, -, \times, <)$ et le corps des complexes $\mathbb{C} = (\mathbb{C}, 0, 1, +, -, \times, =)$. L'absence de la division dans les deux dernières

structures ne constitue pas une perte de généralité pour ce qui est des problèmes de décision, et simplifie la présentation.

La présence de deux constantes distinctes 0 et 1, si elle est naturelle sur la structure standard ou sur un corps, l'est moins en ce qui concerne les espaces vectoriels $\mathbb{R}_{ovs}$ et $\mathbb{R}_{vs}$ par exemple. On peut facilement s'affranchir de cette convention. Une manière de procéder est décrite dans [Hem01].

Autre exemple : les dictionnaires arborescents de Poizat [Poi95]. Le domaine de cette structure est l'ensemble des arbres binaires infinis dont les nœuds sont étiquetés par un élément de $\{0,1\}$, ainsi que les constantes 0 et 1. Les fonctions sont les destructeurs "fils gauche", "fils droit" et "racine", ainsi que les constantes 0 et 1 et l'égalité.

1.2 Calcul sur une structure

Nous présentons deux modèles de calcul sur une structure quelconque $\mathcal{S}$ de domaine M : la machine de Turing et les circuits.

1.2.1 Mots et langages

Une suite finie d'éléments de M est appelé un mot sur M. On note $M^\infty = \bigcup_{n\in\mathbb{N}} M^n$ l'ensemble des mots sur M. Le mot vide est noté λ. On note $|x|$ la longueur d'un mot x. Ainsi $|x| = n$ pour $x \in M^n$, et $|\lambda| = 0$. On définit la concaténation xy de deux mots $x = (x_1, \ldots, x_n)$ et $y = (y_1, \ldots, y_m)$ de M^∞ par $xy = (x_1, \ldots, x_n, y_1, \ldots, y_m)$. On définit également un codage raisonnable $\langle\cdot,\cdot\rangle : M^\infty \times M^\infty \to M^\infty$, par exemple $\langle x, y\rangle = (1, x_1, 1, x_2, \ldots, 1, x_n, 0, y_1, y_2, \ldots, y_m)$ On définit également par récurrence sur n le codage $\langle x_1, \ldots, x_n\rangle = \langle x_1, \langle x_2, \ldots, x_n\rangle\rangle$.

Une partie de M^∞ est appelée langage ou problème. On définit l'union disjointe de deux langages L_1 et L_2 de M^∞ par $L_1 \oplus L_2 = \{0x,\ x \in L_1\} \cup \{1x,\ x \in L_2\}$. Pour deux ensembles de langages $\mathcal{C}_1$ et $\mathcal{C}_2$, on définit $\mathcal{C}_1 \oplus \mathcal{C}_2 = \{L_1 \oplus L_2,\ L_1 \in \mathcal{C}_1 \text{ et } L_2 \in \mathcal{C}_2\}$.

1.2.2 Machine de Turing

Une machine de Turing à k rubans sur une structure $\mathcal{S}$ est composée de k rubans bi-infinis (la mémoire), chacun muni d'une tête de lecture écriture, et d'un graphe fini (le programme). Chaque case de la mémoire contient un élément de M ou bien le symbole blanc $\diamond \notin M$. On note $C_i \in (M \cup \{\diamond\})^{\mathbb{Z}}$ le contenu du ruban i, et $p_i \in \mathbb{Z}$ la position de la tête sur ce ruban. Le programme est un graphe fini orienté étiqueté (Q, δ, q_0), $q_0 \in Q$ étant le sommet initial et δ la relation d'incidence (ou fonction de branchement). On note $q \in Q$ l'instruction courante. Les sommets Q du graphe sont étiquetés par les instructions suivantes :

- bouger la tête i vers la gauche ou la droite : $p_i \leftarrow p_i - 1$ ou $p_i \leftarrow p_i + 1$,
- écrire un symbole blanc : $C_i(p_i) \leftarrow \diamond$,

- copier un élément : $C_i(p_i) \leftarrow C_j(p_j)$,
- pour un fonction f de $\mathcal{S}$ d'arité r, appliquer la fonction f au r éléments pointés par p_j et écrire le résultat dans la case pointée par p_i : $C_i(p_i) \leftarrow f(C_j(p_j), C_j(p_j + 1), \ldots, C_j(p_j + r - 1))$,
- instruction de branchement $branche(p_j)$: cela de permet de faire un saut si $C_j(p_j) = 0$,
- instruction d'arrêt *stop*.

Le degré sortant d'un sommet étiqueté *stop* est nul. Un sommet étiqueté par une instruction de branchement est de degré sortant 2 : l'une des arêtes est étiquetée 0 et l'autre 1. Les sommets étiquetés par d'autres instructions sont de degré sortant 1.

On appelle configuration de la machine à un instant donné le triplet (C, p, q) composé des contenus de chaque ruban, de la position des têtes et de l'instruction courante. Sur l'entrée $(x_1, \ldots, x_n) \in M^\infty$ la configuration initiale est la suivante : toutes les cases des rubans contiennent des symboles blancs à l'exceptions des n premières cases pointées par p_1 : pour $0 \leqslant i \leqslant n-1$, $C_1(p_1 + i) = x_i$. Toutes les têtes sont en position nulle : $p_i = 0$ pour $1 \leqslant i \leqslant k$. Enfin, l'instruction courante est l'état initial : $q = q_0$. On note $(C, p, q) \vdash (C', p', q')$ si on passe de la première configuration à la seconde en un pas de calcul. Un pas de calcul se déroule comme ceci. Si l'instruction est *stop*, la machine s'arrête. Si c'est l'instruction de branchement $branche(p_j)$, la machine va dans l'état correspondant à l'arête sortante de q étiquetée 0 si $C_j(p_j) = 0$ et dans l'autre sinon. Pour les autres instructions, le contenu des rubans et la position des têtes de la machine sont modifiés comme expliqué ci-dessus, et la machine va dans l'état correspondant à l'unique arrête sortante de cet état.

Quand on lance le calcul d'une machine A sur l'entrée $(x_1, \ldots, x_n)$, on a deux possibilités. Si la machine ne s'arrête pas, on dit qu'elle diverge et on note $f_A(x_1, \ldots, x_n) = \uparrow$. Sinon, elle s'arrête au bout d'un temps fini. Soit $(y_1, \ldots, y_m)$ le uple pointé par la tête du ruban k – autrement dit, $C_k(p_k + i) = y_{i+1} \in M$ pour $0 \leqslant i \leqslant m-1$ et $C_k(p_k + m)$ est le symbole blanc. Dans ce cas, on dit que la machine A converge et qu'elle rend la réponse $f_A(x_1, \ldots, x_n) = (y_1, \ldots, y_m) \in M^\infty$. Ainsi chaque machine A calcule une fonction $f_A : M^\infty \to M^\infty \cup \{\uparrow\}$. On étudie souvent des problèmes de décision. Dans ce cas, chaque élément de M^∞ est accepté ou rejeté par la machine A. On dira qu'une machine A décide le langage $L \subset M^\infty$ si elle calcule sa fonction caractéristique. Autrement dit, $f_A(\bar{x}) = 1$ si $\bar{x} \in L$ et $f_A(\bar{x}) = 0$ sinon. Le temps de calcul mis par la machine A sur l'entrée $(x_1, \ldots, x_n)$ est le nombre de pas de calcul que la machine a effectués avant d'arriver dans l'état *stop* si le calcul converge, et ∞ si le calcul diverge. On note $\text{time}_A(n)$ le sup du temps de calcul de A sur les entrées de M^n. Ainsi time_A est une application de $\mathbb{N} \to \mathbb{N} \cup \{\infty\}$.

La règle générale est de permettre l'utilisation d'un nombre fini arbitrairement grand de constantes de la structure, en plus de celles présentes dans la signature, pour une machine donnée. On les appelle des paramètres. Cela signifie que lorsqu'on parle d'une machine sur $\mathcal{S} = (M, f_1, \ldots, f_s)$, c'est en fait la donnée d'une suite $(\alpha_1, \ldots, \alpha_p)$ d'éléments de M et d'une machine (sans paramètres supplémentaires) sur la structure $(M, f_1, \ldots, f_s, \alpha_1, \ldots, \alpha_p)$.

1.2.3 Circuits

On présente maintenant un autre modèle de calcul, celui des circuits. C'est à partir de ce modèle que nous définirons les classes parallèles. Une alternative serait de définir des machines de Turing parallèles. La différence fondamentale entre la vision "circuits" et "machine de Turing" vient du fait qu'un circuit donné traite les entrées d'une taille fixée. Si on veut calculer une fonction f définie sur M^∞ avec des circuits, il faut se donner une famille de circuits $(C_n)_{n\in\mathbb{N}}$, le circuit C_n calculant la restriction de f à M^n. On parle de modèle de calcul *non-uniforme*. Cette approche des questions de complexité structurelle est défendue dans [Poi95]. Pour l'étude de la complexité classique du point de vue des circuits, consulter [Vol99] et [Weg87].

Soit $\mathcal{S}$ une structure. Un circuit C sur $\mathcal{S}$ est un graphe orienté acyclique dont chaque nœud est étiqueté par une entrée x_i, une sortie y_j ou une fonction f_i de la structure $\mathcal{S}$. Les nœuds d'entrée $x_1, \ldots, x_n$ sont de degré entrant nul, et les nœuds étiquetés par une fonction f_i sont de degré entrant égal à leur arité r_i. Les nœuds $y_1, \ldots, y_m$ sont de degré entrant 1 et de degré sortant nul, ils calculent la fonction identité et sont utiles pour désigner la sortie du circuit. On a également besoin d'un nœud appelé sélecteur $S(x, y, z)$ défini par $S(1, y, z) = y$, et $S(x, y, z) = z$ pour $x \neq 1$. Ainsi, un circuit à n entrées $(x_1, \ldots, x_n)$ et m sorties $(y_1, \ldots, y_m)$ calcule une fonction de M^n dans M^m. Une suite de circuits $(C_n)_{n\in\mathbb{N}}$, où C_n à n entrées, calcule donc une fonction de M^∞ dans M^∞. Comme précédemment, on dit qu'un circuit décide une partie de M^n si elle calcule la fonction caractéristique de cette partie. De même, on parle d'une famille de circuits décidant un langage sur M. On appelle taille d'un circuit C et on note $\mathrm{size}(C)$ le nombre de nœuds de ce circuit, et profondeur la longueur du plus long chemin dans le circuit.

Une famille $(C_n)_{n\in\mathbb{N}}$ de circuits est dite P-uniforme si :
- la fonction qui sur l'entrée 1^n donne le nombre de sorties du circuit C_n est calculable en temps polynomial,
- il existe pour chaque $n \in \mathbb{N}$ une numérotation des nœuds de C_n à valeurs dans $\{1, \ldots, \mathrm{size}(C_n)^{O(1)}\}$, le numéro du nœud de sortie y_j étant j,
- la fonction qui à $\langle 1^n, i\rangle$ associe la description du nœud numéro i du circuit C_n est calculable en temps polynomial (sur une machine de Turing classique) ; cette description comporte un codage de l'étiquette du nœud i ainsi que la suite, dans l'ordre, des numéros des nœuds entrants.

Comme pour la machine de Turing, une famille de circuits peut utiliser un nombre fini arbitrairement grand de paramètres. Autrement dit, se donner une famille de circuits (avec paramètres) sur une structure $\mathcal{S}$, c'est se donner un vecteur de paramètres $(\alpha_1, \ldots, \alpha_p) \in M^\infty$ et une famille de circuits sans paramètres sur la structure M étendue par les constantes $(\alpha_1, \ldots, \alpha_p)$.

1.3 Classes de complexité

On définit dans cette section un certain nombre de classes de complexité sur une structure arbitraire. On retrouve les classes de complexité classiques en prenant la structure $\mathcal{S} = \mathbb{Z}_2$. Si on omet de préciser une structure en indice, cela signifie qu'on fait référence au cas classique.

Le temps polynomial. Soit $\mathcal{S}$ une structure de domaine M. On définit $\mathrm{P}_\mathcal{S}$ la classe des langages de M^∞ décidés par une machine de Turing sur $\mathcal{S}$ en temps polynomial. Dire que $L \in \mathrm{P}_\mathcal{S}$ signifie donc qu'il existe une machine de Turing A (avec paramètres) sur la structure $\mathcal{S}$ calculant la fonction caractéristique de L sur M^∞, et telle que $\mathrm{time}_A(n) = n^{O(1)}$. On définit aussi $\mathrm{PF}_\mathcal{S}$ l'ensemble des fonctions de M^∞ dans M^∞ calculées par une machine de Turing $\mathcal{S}$ en temps polynomial.

On peut également définir $\mathrm{P}_\mathcal{S}$ en terme de circuits. La classe $\mathrm{P}_\mathcal{S}$ est l'ensemble des langages de M^∞ décidés par une suite P-uniforme de circuits de tailles polynomiales (avec un nombre fini de paramètres, les mêmes pour tous les circuits, conformément à la définition donnée ci-dessus). En effet, le calcul d'une machine fonctionnant en t pas peut être simulé par un circuit de taille $O(t^2)$, et cette famille est uniforme quand t parcourt $\mathbb{N}$. L'autre sens vient du fait que l'évaluation de circuits se fait en temps polynomial.

Le temps exponentiel. La classe $\mathrm{EXP}_\mathcal{S}$ est l'ensemble des langages de M^∞ décidés par une machine de Turing sur $\mathcal{S}$ en temps exponentiel. Autrement dit, $L \in \mathrm{EXP}_\mathcal{S}$ s'il existe une machine à paramètres A sur $\mathcal{S}$ décidant L et vérifiant $\mathrm{time}_A(n) = 2^{n^{O(1)}}$.

Le temps polynomial non-déterministe. On définit maintenant la classe $\mathrm{NP}_\mathcal{S}$. Un langage $L \subset M^\infty$ est dans $\mathrm{NP}_\mathcal{S}$ si et seulement s'il existe un langage $A \in \mathrm{P}_\mathcal{S}$ et un polynôme p tels que pour tout élément $x \in M^\infty$

$$x \in L \quad \text{ssi} \quad \exists y \in M^{p(|x|)}\ \langle x, y\rangle \in A.$$

Statut de "P = NP ?" : divers exemples.

Exemple 1. P $\neq$ NP sur la structure $\mathbb{Z} = (\mathbb{Z}, 0, 1, +, -, =)$. En effet, $\{\langle x_1, \ldots, x_n\rangle \in \mathbb{Z}^\infty,\ x_1 \text{ est pair}\}$ est dans $\mathrm{NP}_\mathbb{Z}$ mais pas dans $\mathrm{P}_\mathbb{Z}$: plus précisément, cet ensemble n'est pas décidable sur $\mathbb{Z}$. Plus généralement, le fait que $\mathrm{P}_\mathbb{Z} \neq \mathrm{NP}_\mathbb{Z}$ est une conséquence de la proposition suivante.

Proposition 1.1 *Si* P = NP *sur une structure S, alors la structure $S \cup \mathrm{dom}(S)$ – la structure S étendue par des constantes nommant chacun des éléments de son domaine – a l'élimination des quantificateurs.*

Le fait que les langages de NP sur les structures $\mathbb{C}$ et $\mathbb{R}$ (et donc $\mathbb{R}_{ovs}$ et $\mathbb{R}_{vs}$) soient décidables est une conséquence de l'élimination des quantificateurs sur ces structures. Ceci dit, l'élimination des quantificateurs sur une structure n'entraîne pas $\mathrm{P}_\mathcal{S} = \mathrm{NP}_\mathcal{S}$

comme le montre l'exemple suivant.

Exemple 2. Sur $\mathbb{R}_{vs}$, on a $\mathrm{P} \neq \mathrm{NP}$: voir chapitre 5.

Exemple 3. Les dictionnaires arborescents $\mathcal{A}$ (définis en section 1.1) vérifient $\mathrm{P}_{\mathcal{A}} \neq \mathrm{NP}_{\mathcal{A}}$. En effet l'ensemble des $\langle a, 1^n \rangle$ tels que l'arbre a a (au moins) un nœud étiqueté 1 à la profondeur n est dans $\mathrm{NP}_{\mathcal{A}} \setminus \mathrm{P}_{\mathcal{A}}$.

Exemple 4. La question $\mathrm{P} = \mathrm{NP}$ est ouverte sur de nombreuses structures intéressantes, par exemple $\mathbb{R}$, $\mathbb{C}$, $\mathbb{R}_{ovs}$ et bien sûr $\mathbb{Z}_2$.

La classe des compléments. Pour $C_{\mathcal{S}}$ une classe de complexité sur $\mathcal{S}$ de domaine M, on note co-$C_{\mathcal{S}}$ l'ensemble des complémentaires des langages de $C_{\mathcal{S}}$:

$$\text{co-}C_{\mathcal{S}} = \{M^\infty \setminus L,\ L \in C_{\mathcal{S}}\}.$$

On vérifie que la classe co-$\mathrm{NP}_{\mathcal{S}}$ est l'ensemble des langages L tels qu'il existe un langage $A \in \mathrm{P}_{\mathcal{S}}$ et un polynôme p tels que

$$x \in L \quad \text{ssi} \quad \forall y \in M^{p(|x|)}\ \langle x, y \rangle \in A.$$

La hiérarchie polynomiale. On définit $\Sigma^0_{\mathcal{S}} = \mathrm{P}_{\mathcal{S}}$. Par récurrence, on définit pour $k \geqslant 1$ les classes $\Sigma^{k+1}_{\mathcal{S}}$ et $\Pi^{k+1}_{\mathcal{S}}$. Un langage L est dans $\Sigma^{k+1}_{\mathcal{S}}$ si et seulement s'il existe un langage $A \in \Pi^k_{\mathcal{S}}$ et un polynôme p tels que pour tout élément $x \in M^\infty$

$$x \in L \quad \text{ssi} \quad \exists y \in M^{p(|x|)}\ \langle x, y \rangle \in A.$$

Pour tout $k \geqslant 0$ on pose $\Pi^k_{\mathcal{S}} = \text{co-}\Sigma^k_{\mathcal{S}}$. Ainsi $\Sigma^1_{\mathcal{S}} = \mathrm{NP}_{\mathcal{S}}$ et $\Pi^1_{\mathcal{S}} = \text{co-}\mathrm{NP}_{\mathcal{S}}$. On pose également $\Delta^k_{\mathcal{S}} = \Sigma^k_{\mathcal{S}} \cap \Pi^k_{\mathcal{S}}$. On définit la hiérarchie polynomiale $\mathrm{PH}_{\mathcal{S}} = \bigcup_{k \in \mathbb{N}} \Sigma^k_{\mathcal{S}}$. Si deux niveaux consécutifs de la hiérarchie $\Sigma^k_{\mathcal{S}}$ et $\Sigma^{k+1}_{\mathcal{S}}$ coïncident, alors $\mathrm{PH}_{\mathcal{S}} = \Sigma^k_{\mathcal{S}}$ et on dit que la hiérarchie polynomiale s'effondre au niveau k.

Non-déterminisme booléen. On définit l'équivalent des classes $\Sigma^k_{\mathcal{S}}$, $\Pi^k_{\mathcal{S}}$ et $\mathrm{PH}_{\mathcal{S}}$ pour des quantifications booléennes. On définit la classe $\mathrm{BNP}_{\mathcal{S}}$ comme ceci : un langage $L \subset M^\infty$ est dans $\mathrm{BNP}_{\mathcal{S}}$ si et seulement s'il existe un langage $A \in \mathrm{P}_{\mathcal{S}}$ et un polynôme p tels que pour tout élément $x \in M^\infty$

$$x \in L \quad \text{ssi} \quad \exists y \in \{0,1\}^{p(|x|)}\ \langle x, y \rangle \in A.$$

On définit de la même manières les classes $\mathrm{B}\Sigma^{\mathrm{k}}_{\mathcal{S}}$ et $\mathrm{B}\Pi^{\mathrm{k}}_{\mathcal{S}}$, et on pose $\mathrm{BPH}_{\mathcal{S}} = \bigcup_{\mathrm{k} \in \mathbb{N}} \mathrm{B}\Sigma^{\mathrm{k}}_{\mathcal{S}}$.

Calcul avec oracle. Considérons A une machine sur la structure $\mathcal{S}$ de domaine M, et soit $L \subset M^\infty$. Une machine A avec l'oracle L, c'est une machine qui dispose d'instructions supplémentaires de type "$C_i(p_i) \leftarrow \mathrm{oracle}(p_j)$". Considérons une machine sur cette instruction et notons $(x_1, \ldots, x_k)$ le uple pointé par p_j (on rappelle que le

premier symbole blanc indique la fin de ce uple). À l'étape, suivante, la case pointée par la tête p_i contient 1 si $(x_1, \ldots, x_k) \in L$, et 0 sinon. Notons bien que cela se fait en un pas de calcul seulement (la complexité de L n'entre pas en jeu). Pour une classe de complexité $C_{\mathcal{S}}$, $\mathcal{S}$ de domaine M, et $L \subset M^\infty$, on note $C_{\mathcal{S}}(L)$ l'ensemble des langages reconnus par les machines de $C_{\mathcal{S}}$ avec l'oracle L.

Important : les langages de $C_{\mathcal{S}}(L)$ ne se déduisent pas de la connaissance de $C_{\mathcal{S}}$ comme partie de $\mathcal{P}(M^\infty)$ et de L, mais du dispositif de calcul définissant $C_{\mathcal{S}}$ et de L. Penser au théorème de Baker, Gill et Solovay [BGS75, BDG90] : il existe un langage L tel que $\mathrm{P}(L) \neq \mathrm{NP}(L)$, et un langage L' tel que $\mathrm{P}(L') = \mathrm{NP}(L')$.

Étant donné une classe de complexité $C_{\mathcal{S}}$ sur une structure $\mathcal{S}$ de domaine M et $\mathcal{L}$ un ensemble de langages sur M, on note $C_{\mathcal{S}}(\mathcal{L}) = \bigcup_{L \in \mathcal{L}} C_{\mathcal{S}}(L)$. En particulier, pour une classe de complexité $\mathcal{D}_{\mathcal{S}}$, on définit $C_{\mathcal{S}}(\mathcal{D}_{\mathcal{S}})$ (où $C_{\mathcal{S}}$ fait référence à un dispositif de calcul alors que $\mathcal{D}_{\mathcal{S}}$ est vu comme un ensemble de langages de M). Si on considère $N \subset M$, tout langage sur N est aussi un langage sur M et on définit $C_{\mathcal{S}}(L)$ comme ci-dessus pour un langage L de N^∞. En particulier, puisqu'on a toujours $\{0,1\} \subset M$, l'utilisation d'oracles booléens est toujours bien définie.

On peut également définir la hiérarchie polynomiale en termes d'oracles comme dans le cas classique [BDG95]. On a $\Sigma_{\mathcal{S}}^{k+1} = \mathrm{NP}_{\mathcal{S}}(\Sigma_{\mathcal{S}}^k)$ et $\Pi_{\mathcal{S}}^{k+1} = \text{co-NP}_{\mathcal{S}}(\Sigma_{\mathcal{S}}^k)$. De même, $\mathrm{B}\Sigma_{\mathcal{S}}^{\mathrm{k}+1} = \mathrm{BNP}_{\mathcal{S}}(\mathrm{B}\Sigma_{\mathcal{S}}^{\mathrm{k}})$, et $\mathrm{B}\Pi_{\mathcal{S}}^{\mathrm{k}} = \text{co-BNP}_{\mathcal{S}}(\mathrm{B}\Sigma_{\mathcal{S}}^{\mathrm{k}})$.

Proposition 1.2 *[BDG95] Soit $A \in \mathrm{BNP}_S$. Par définition il existe un problème associé $B_S \in \mathrm{P}_S$ et un polynôme p tels que*

$$A = \{x \in M^\infty,\ \exists y \in \{0,1\}^{p(|x|)}\ \langle x, y\rangle \in B\}.$$

Il existe un algorithme de $\mathrm{PF}(\mathrm{BNP}_S)$ qui sur l'entrée x détermine si $x \in A$ et calcule en cas de réponse positive un certificat y correspondant.

Démonstration. La méthode est ce qu'on appelle une "recherche préfixe". Définissons

$$\tilde{A} = \{\langle x, y\rangle \mid (x,y) \in M^\infty \times \{0,1\}^\infty, \exists z \in \{0,1\}^\infty\ \langle x, yz\rangle \in A\}.$$

Bien sûr $\tilde{A} \in \mathrm{BNP}_S$. Considérons l'algorithme suivant qui prend en entrée $x \in M^\infty$. Si $\langle x, \lambda\rangle \notin \tilde{A}$ on rejette l'entrée x. Sinon, on pose $y := \lambda$ et on entre dans une boucle. À chaque étape, on teste si $\langle x, y0\rangle \in \tilde{A}$ auquel cas on pose $y := y0$; sinon, on teste si $\langle x, y1\rangle \in \tilde{A}$ et on pose $y := y1$ si c'est le cas. On sort de cette boucle quand aucune de ces deux conditions n'est satisfaite. La valeur de y est alors un certificat pour x. Cet algorithme fonctionne bien sûr en temps polynomial. □

Fonctions conseil. Une fonction conseil est une application $f : \mathbb{N} \to M^\infty$. Pour une fonction conseil f, on définit $C_{\mathcal{S}}/f$ l'ensemble des langages L tels qu'il existe $A \in C_{\mathcal{S}}$ vérifiant

$$x \in L \quad \text{ssi} \quad \langle x, f(|x|)\rangle \in A.$$

Noter que le conseil $f(|x|)$ obtenu sur une entrée x ne dépend que de la longueur de x. Pour $\mathcal{F}$ une famille de fonctions, on définit $C_{\mathcal{S}}/\mathcal{F} = \bigcup_{f \in \mathcal{F}} C_{\mathcal{S}}/f$. En particulier on

définit "poly" l'ensemble des fonctions f à valeurs dans $\{0,1\}^\infty$ vérifiant $|f(n)| = n^{O(1)}$. Il existe une caractérisation intéressante des langages de $\mathrm{P}_\mathcal{S}/\mathrm{poly}$: ce sont les langages qui sont décidés par une suite de circuits sur $\mathcal{S}$ de tailles polynomiales (utilisant un nombre fini de paramètres, les mêmes pour toute taille d'entrée, comme expliqué ci-dessus).

Le temps parallèle polynomial. La classe PAR_S est l'ensemble des langages de M^∞ décidés par une famille P-uniforme de circuits de profondeur polynomiale. On a l'inclusion suivante.

Proposition 1.3 *[BCSS98] Sur les structures* $\mathbb{Z}_2$, $\mathbb{C}$, $\mathbb{R}$, $\mathbb{R}_{ovs}$ *et* $\mathbb{R}_{vs}$, *on a* $\mathrm{NP} \subset \mathrm{PAR}$.

Cette inclusion est triviale dans le cas classique, il suffit pour une entrée donnée d'essayer tous les certificats en parallèle. Cet argument ne tient plus pour une structure de domaine infini. Les inclusions $\mathrm{NP}_\mathbb{C} \subset \mathrm{PAR}_\mathbb{C}$ et $\mathrm{NP}_\mathbb{R} \subset \mathrm{PAR}_\mathbb{R}$ résultent d'algorithmes d'élimination des quantificateurs efficaces sur ces structures.

Notons que dans le cas classique, PAR coïncide avec la classe PSPACE des langages décidés en espace polynomial.

Calcul sans paramètres. D'une manière générale, pour une classe de complexité $C_\mathcal{S}$, on note $C^0_\mathcal{S}$ la classe de complexité correspondante si l'on autorise pas l'utilisation de paramètres.

1.4 Réduction et complétude

La notion de problème complet permet d'identifier les problèmes les plus difficiles au sein d'une classe. Pour cela, on a besoin de comparer la difficulté de deux problèmes : c'est la notion de réduction qui le permet. Dire qu'on peut réduire un problème B à un problème A signifie que résoudre B est "plus facile" que de résoudre A : si on a un algorithme efficace pour résoudre A, alors on en déduit un pour B. Il existe différents types de réductions.

La réduction many-one (ou réduction de Karp). On dit qu'un problème $B \subset M^\infty$ se réduit au problème $A \subset M^\infty$ s'il existe une fonction $f : M^\infty \to M^\infty$ telle que pour tout $x \in M^\infty$, $x \in B$ si et seulement si $f(x) \in A$. Imposer en plus des conditions sur f permet de définir différentes réductions. Si on s'intéresse à la calculabilité sur la structure $\mathcal{S}$, on demande que f soit calculable sur $\mathcal{S}$. Si on veut étudier la classe $\mathrm{NP}_\mathcal{S}$, les réductions intéressantes sont celles qui sont calculables en utilisant moins de ressource que $\mathrm{NP}_\mathcal{S}$. On considère dans la suite la réduction many-one en temps polynomial : on impose que le temps de calcul de f soit polynomialement borné. On note cette réduction $\leqslant_m$, et on écrit $B \leqslant_m A$ si B se réduit à A par une telle réduction.

La réduction Turing (ou réduction de Cook). Un problème B est Turing réductible à A en temps polynomial sur la structure $\mathcal{S}$ si $B \in \mathrm{P}_\mathcal{S}(A)$. On note ceci

$B \leqslant_T A$. Si $B \leqslant_m A$ alors $B \leqslant_T A$. La réduction many-one est donc plus fine que la réduction Turing.

Complétude. Considérons une classe de complexité $C_{\mathcal{S}}$ et une réduction $\leqslant$. Un langage $A \subset M^\infty$ est dit $C_{\mathcal{S}}$-dur pour la réduction $\leqslant$ si tout langage de $C_{\mathcal{S}}$ se réduit à A via $\leqslant$. Un langage A est dit $C_{\mathcal{S}}$-complet pour la réduction $\leqslant$ si A est $C_{\mathcal{S}}$-dur et $A \in C_{\mathcal{S}}$.

Un problème naturellement NP$_{\mathcal{S}}$**-complet.** Soit $H_{\mathcal{S}} \subset M^\infty$ l'ensemble des $\langle A, 1^t\rangle$, où A est la description d'une machine de Turing non-déterministe sur la structure $\mathcal{S}$ (par exemple la description booléenne du graphe des instructions suivi du vecteur de paramètres) tels que la machine A accepte l'entrée vide en au plus t pas de calculs (cela signifie qu'il existe $y \in M^t$ tel que A accepte $\langle \lambda, y\rangle$ en au plus t pas). Comme dans le cas classique [BDG95], ce problème $H_{\mathcal{S}}$ est NP$_{\mathcal{S}}$-complet pour la réduction many-one en temps polynomial. Il existe donc toujours des problèmes NP$_{\mathcal{S}}$-complet.

Remarque. L'argument précédent ne tient plus dans le cas d'une structure de signature infinie. Considérons par exemple la structure $(\mathbb{Z}, 0, 1, \{\lambda_k\}_{k\in\mathbb{N}}, =)$, où λ_k est la fonction $x \mapsto k \times x$. Il est prouvé dans [Gaß97] que cette structure ne possède pas de langage NP-complet pour la réduction many-one en temps polynomial.

Problèmes naturels NP**-complets dans le cas classique.** Il existe une grande diversité de problèmes NP-complets : consulter par exemple le catalogue de Garey et Johnson [GJ79]. Nous donnons ici trois exemples.

◦ Le problème SAT. Donnée : une formule de la logique propositionnelle. Déterminer si elle est satisfaisable.

◦ Le problème du "voyageur de commerce" TSP (Travelling Salesman Problem). Étant donné une matrice $d_{ij} \in \mathbb{N}, 1 \leqslant i, j \leqslant n$ et un entier K, déterminer s'il existe une permutation σ de $\{1, \ldots, n\}$ telle que

$$\sum_{i=1}^{n-1} d_{\sigma(i),\sigma(i+1)} + d_{\sigma(n),\sigma(1)} \leqslant K.$$

La variante où on se restreint à la distance euclidienne sur $\mathbb{N}^2$ est également NP-complète : on se donne les coordonnées entières d'un ensemble de points de $\mathbb{N}^2$ et un entier K, et on demande s'il existe un polygone de longueur au plus K reliant tous ces points.

◦ Le problème du "sac-à-dos" KP (Knapsack Problem) ou problème de la sous-somme. Étant donné des entiers positifs $x_1, \ldots, x_n, y$, déterminer s'il existe $I \subset \{1, \ldots, n\}$ tel que $\sum_{i\in I} x_i = y$.

Problèmes naturels NP**-complets sur les réels et les complexes.** On donne ci-dessous quelques problèmes complets pour la réduction many-one en temps

polynomial sur $\mathbb{R}$ ou $\mathbb{C}$. Précisons que dans les langages que nous allons définir, les polynômes sur $\mathbb{R}$ ou $\mathbb{C}$ sont codés sous forme pleine. Par exemple, un polynôme de degré total d en n variables sera codé $\langle d, n, \bar{a}\rangle$ où $\bar{a}$ est la suite des coefficients de monômes (dans l'ordre lexicographique).

◦ Le problème $\mathrm{HN}_{\mathbb{C}}$ [BCSS98]. Donnée : un système composé de polynômes $f_1, \ldots, f_q \in \mathbb{C}[x_1, \ldots, x_n]$. Les f_i ont-ils un zéro commun sur $\mathbb{C}$? Ce problème est $\mathrm{NP}_{\mathbb{C}}$-complet.

◦ Le problème $\mathrm{4FEAS}_{\mathbb{R}}$ [BSS89, BCSS98]. Donnée : $f \in \mathbb{R}[x_1, \ldots, x_n]$ de degré total au plus 4. Existe-t-il $\bar{a} \in \mathbb{R}^n$ tel que $f(\bar{a}) = 0$? Ce problème est $\mathrm{NP}_{\mathbb{R}}$-complet.

◦ Le problème de la dimension [Koi99b]. Donnée : un système semi-algébrique sur $\mathbb{R}$, c'est-à-dire un ensemble d'équations réelles de type $f_i \Delta 0$ avec $f_i \in \mathbb{R}[x_1, \ldots, x_n]$ et $\Delta \in \{=, <, \leqslant\}$, et un entier d. L'ensemble des points de $\mathbb{R}^n$ solutions de ce système est-il de dimension au moins d ? Ce problème est $\mathrm{NP}_{\mathbb{R}}$-complet.

1.5 Théorème de transfert

Résoudre le problème P = NP sur $\mathbb{R}$, $\mathbb{R}_{ovs}$ ou $\mathbb{C}$, par exemple, semble difficile. À défaut d'être en mesure d'apporter des réponses, on essaie de relier ces problèmes à des questions de complexité sur d'autres structures, en particulier au cas classique.

Nous appelons théorème de transfert toute proposition du type

$$C_{\mathcal{S}} \subset D_{\mathcal{S}} \Rightarrow E_{\mathcal{T}} \subset F_{\mathcal{T}}$$

où $\mathcal{S}$ et $\mathcal{T}$ sont deux structures et C,D,E,F quatre classes de complexité. Dans les sections suivantes, nous décrivons brièvement deux méthodes pour obtenir des théorèmes de transfert et les résultats qui s'y rapportent. De nouvelles méthodes seront présentées aux chapitres 4 et 5.

1.5.1 Parties booléennes

On appelle partie booléenne d'une classe de complexité $C_{\mathcal{S}} \subset \mathcal{P}(M^\infty)$

$$\mathrm{BP}(C_{\mathcal{S}}) = \{L \cap \{0,1\}^\infty \mid L \in C_{\mathcal{S}}\}.$$

Il est alors possible d'obtenir un théorème de transfert en remarquant que

$$C_{\mathcal{S}} \subset D_{\mathcal{S}} \Rightarrow \mathrm{BP}(C_{\mathcal{S}}) \subset \mathrm{BP}(D_{\mathcal{S}}).$$

Le point important est de réussir à trouver des inclusions entre les ensembles $\mathrm{BP}(C_{\mathcal{S}}), \mathrm{BP}(D_{\mathcal{S}})$ et des classes de complexités classiques.

Exemple 1. Dans [Koi97] il est prouvé que $\mathrm{BP}(\mathrm{P}_{\mathbb{C}}) \subset \mathrm{BPP}$. Comme bien sûr $\mathrm{NP} \subset \mathrm{BP}(\mathrm{NP}_{\mathbb{C}})$, ceci donne

$$\mathrm{NP}_{\mathbb{C}} \subset \mathrm{P}_{\mathbb{C}} \Rightarrow \mathrm{NP} \subset \mathrm{BPP}.$$

Remarquons que $\mathrm{NP}_{\mathbb{C}} \subset \mathrm{P}_{\mathbb{C}}$ est équivalent à $\mathrm{P}_{\mathbb{C}} = \mathrm{NP}_{\mathbb{C}}$ – comme sur toute autre structure. Que cela nous apprend-il ? Cela laisse penser que $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$, sinon tout problème de NP serait résolu par un "bon" algorithme probabiliste en temps polynomial. Rappelons aussi [BDG95] que $\mathrm{BPP} \subset \mathrm{P/poly}$. Donc on aurait $\mathrm{NP} \subset \mathrm{P/poly}$, ce qui signifie que tout problème de NP serait résolu par une suite de circuits de tailles polynomiales, et la hiérarchie polynomiale s'effondrerait alors au deuxième niveau [BDG95].

Exemple 2. Cucker et Grigoriev [CG97] ont montré que $\mathrm{BP}(\mathrm{PAR}_{\mathbb{R}}) = \mathrm{PSPACE/poly}$. Comme $\mathrm{BP}(\mathrm{EXP}_{\mathbb{R}}) = \{0,1\}^{\infty}$, ceci prouve que $\mathrm{PAR}_{\mathbb{R}} \neq \mathrm{EXP}_{\mathbb{R}}$.

Exemple 3. Les résultats de Cucker et Koiran sur la hiérarchie polynomiale sur les réels avec addition : voir chapitre 2.

1.5.2 Structures élémentairement équivalentes

Deux structures de même signature sont dites élémentairement équivalentes si elles vérifient les mêmes formules du premier ordre (on rappelle que ce sont les formules obtenues à partir des formules atomiques en utilisant un nombre fini de connecteurs logiques $\wedge, \vee, \neg$ et de quantificateurs existentiels et universels $\exists, \forall$). La méthode suivante est due à Michaux [Mic94] – voir aussi [Hem01]. Soit S et S' deux structures élémentairement équivalentes. Si $\mathrm{P}_S^0 = \mathrm{NP}_S^0$ alors H_S est résolu par une machine M sans paramètre fonctionnant en temps polynomial. Pour une taille n d'entrée fixée, on peut écrire une formule du premier ordre $\phi_n(\bar{x})$ exprimant que la machine M accepte $\bar{x}$ – le point important pour réaliser cette étape est que l'on connaît une borne sur le temps de calcul. On peut également écrire une formule $\psi_n(\bar{x})$ exprimant que $\bar{x} \in H_S$ (par une formule existentielle). Ainsi, la formule

$$\forall x_1, \ldots, x_n \, \phi_n(\bar{x}) \leftrightarrow \psi_n(\bar{x})$$

exprime que, pour les entrées de taille n, la machine M décide H_S. Cette formule est donc vraie sur la structure S. Par équivalence élémentaire, elle est vraie aussi sur S'. Or, $\phi_n(\bar{x})$ sur S' exprime que la machine M (vue comme machine sur S') accepte $\bar{x}$, et $\psi_n(\bar{x})$ exprime que $\bar{x} \in H_{S'}$. La machine M, comme machine de S', fonctionne en temps polynomial et décide donc $H_{S'}$. Ceci prouve que $\mathrm{P}_{S'}^0 = \mathrm{NP}_{S'}^0$. Pour résumer :

Théorème 1.1 *[Mic94] Pour $S \equiv S'$, $\mathrm{P}_S^0 = \mathrm{NP}_S^0$ si et seulement si $\mathrm{P}_{S'}^0 = \mathrm{NP}_{S'}^0$.*

Que se passe-t-il si on autorise les paramètres ? Pour une extension élémentaire $S \preccurlyeq S'$ on obtient

$$\mathrm{P}_S = \mathrm{NP}_S \Rightarrow \mathrm{P}_{S'} = \mathrm{NP}_{S'}.$$

En effet, si on suppose $\mathrm{P}_S = \mathrm{NP}_S$, on a une machine avec paramètres décidant H_S en temps polynomial. Cette machine peut être vue comme une machine de S', et celle-ci décide $H_{S'}$. Cela vient du fait que pour $S \preccurlyeq S'$, $\bar{\alpha} \in S^\infty$ et une formule du premier ordre ϕ, $S \models \phi(\bar{\alpha})$ si et seulement si $S' \models \phi(\bar{\alpha})$. Ceci prouve $\mathrm{P}_{S'} = \mathrm{NP}_{S'}$. Bien sûr si on suppose $\mathrm{P}_{S'} = \mathrm{NP}_{S'}$, alors il existe une machine en temps polynomial décidant $H_{S'}$, et cette machine, restreinte à S^∞, décide H_S. Mais cette machine n'est pas P_S a priori puisqu'elle peut utiliser des paramètres de S'.

Exemple 1. Corps algébriquement clos. La théorie des corps algébriquement clos de caractéristique nulle étant complète, deux modèles de cette théorie sont élémentairement équivalents. Notons $\overline{\mathbb{Q}}$ la clôture algébrique de $\mathbb{Q}$. D'après ce qui précède, pour un corps algébriquement clos de caractéristique nulle K, si $\mathrm{P}_{\overline{\mathbb{Q}}} = \mathrm{NP}_{\overline{\mathbb{Q}}}$ alors $\mathrm{P}_K = \mathrm{NP}_K$. En fait, Blum Cucker Shub et Smale ont montré que les corps algébriquement clos de caractéristique nulle ont l'élimination des paramètres : cela signifie que tout langage décidé en temps polynomial avec paramètres peut être décidé en temps polynomial par une machine n'utilisant pas de paramètre. Cela permet donc de montrer le théorème de transfert suivant.

Théorème 1.2 *[BCSS96] La question* $\mathrm{P} = \mathrm{NP}$ *a la même réponse sur tous les corps algébriquement clos de caractéristique nulle.*

Ce théorème admet l'extension suivante [Koi99a] : la hiérarchie polynomiale s'effondre au même niveau sur tous les corps algébriquement clos de caractéristique nulle, ou ne s'effondre sur aucun.

Exemple 2. Corps différentiellement clos de caractéristique nulle. Portier a montré que la réponse à la question $\mathrm{P} = \mathrm{NP}$ est la même sur toutes ces structures [Por99].

Exemple 3. Corps réels clos. Si on considère une extension de corps réels clos $R \preccurlyeq R'$, alors $\mathrm{P}_R = \mathrm{NP}_R$ implique $\mathrm{P}_{R'} = \mathrm{NP}_{R'}$. Concernant la réciproque, on ne sait pas si les corps réels clos ont l'élimination des paramètres. Pour des travaux sur ce sujet, consulter [Mic94], [CK99] et [BDMM00].

2. Calcul sur les réels avec addition

On s'intéresse dans ce chapitre au cas particulier du calcul sur les réels avec addition, ce qui correspond aux structures $\mathbb{R}_{ovs}$ et $\mathbb{R}_{vs}$. On explique tout d'abord pourquoi la notion d'arrangement d'hyperplans [OT91, AS00] est pertinente dans ce cadre. On présente ensuite des résultats de parties booléennes sur ces structures, permettant d'établir des théorèmes transfert [CK95]. Enfin, le modèle des arbres de décision linéaire, crucial dans les chapitres suivants, est introduit.

Considérons une machine décidant un langage $A \in \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$. Soit t un polynôme majorant le temps de calcul parallèle de cette machine. Sur l'entrée $(x_1, \ldots, x_n)$, tous les tests que cette machine effectue sont de la forme $a_0 + a_1 x_1 + \ldots + a_n x_n > 0$, avec $a_i \in \{-2^{t(n)}, \ldots, 2^{t(n)}\}$ – pour voir cela, imaginer le calcul sur une entrée formelle $X_1, \ldots, X_n$. C'est pourquoi, si on veut étudier les langages de $\mathrm{PAR}^0_{\mathbb{R}_{ovs}}$, ce qui inclut en particulier les langages de $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$, il est fructueux d'étudier la structure géométrique de certains ensembles d'hyperplans de $\mathbb{R}^n$.

Sur une machine avec p paramètres $(\alpha_1, \ldots, \alpha_p)$, tout test sur l'entrée $(x_1, \ldots, x_n)$ est de la forme $a_0 + a_1\alpha_1 + \ldots + a_p\alpha_p + a_{p+1}x_1 + \ldots + a_{p+n}x_n > 0$ avec $a_i \in \{-2^{t(n)}, \ldots, 2^{t(n)}\}$. Dans ce cas, on considère que c'est le calcul en dimension $n + p$ d'une machine sans paramètre sur l'entrée $(x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_p)$. On se ramène ainsi au cas précédent.

On est donc intéressé par les ensembles d'hyperplans suivants : pour $n, q \in \mathbb{N}$, on note $\mathcal{H}^n_q$ l'ensemble des hyperplans de $\mathbb{R}^n$ à coefficients dans $\{-q, \ldots, q\}$. On note $\mathcal{L}^n_q$ l'union de ces hyperplans de $\mathbb{R}^n$. Enfin, pour un polynôme t, on note $\mathcal{H}_t$ la famille $(\mathcal{H}^n_{t(n)})_{n\in\mathbb{N}}$ et $\mathcal{L}_t$ le langage réel $\bigcup_{n\in\mathbb{N}} \mathcal{L}^n_{t(n)}$.

2.1 Arrangements d'hyperplans

Soit $H = \{h_1, \ldots, h_m\}$ un ensemble d'hyperplans de $\mathbb{R}^n$. Pour chaque hyperplan, on appelle $h_i^{\oplus}$ et $h_i^{\ominus}$ les deux demi-espaces ouverts délimités par h_i. Pour un point $x \in \mathbb{R}^n$, on pose $z_i(x) = 0$ si $x \in h_i$, $z_i(x) = 1$ si $x \in h_i^{\oplus}$ et$z_i(x) = -1$ si $x \in h_i^{\ominus}$. On définit

$\varphi(x) = (z_1(x), \ldots, z_m(x))$. On considère la relation d'équivalence $\sim$ sur $\mathbb{R}^n$ donnée par

$$x \sim y \Leftrightarrow \varphi(x) = \varphi(y).$$

On appelle faces de l'arrangement $\mathcal{A}(H)$ les classes d'équivalences de la relation $\sim$. On appelle dimension d'une face la dimension de sa clôture affine. Une face de dimension 0 est appelée un sommet de l'arrangement, une face de dimension n cellule. Un exemple d'arrangement d'hyperplans en dimension 2 est donnée figure 2.1.

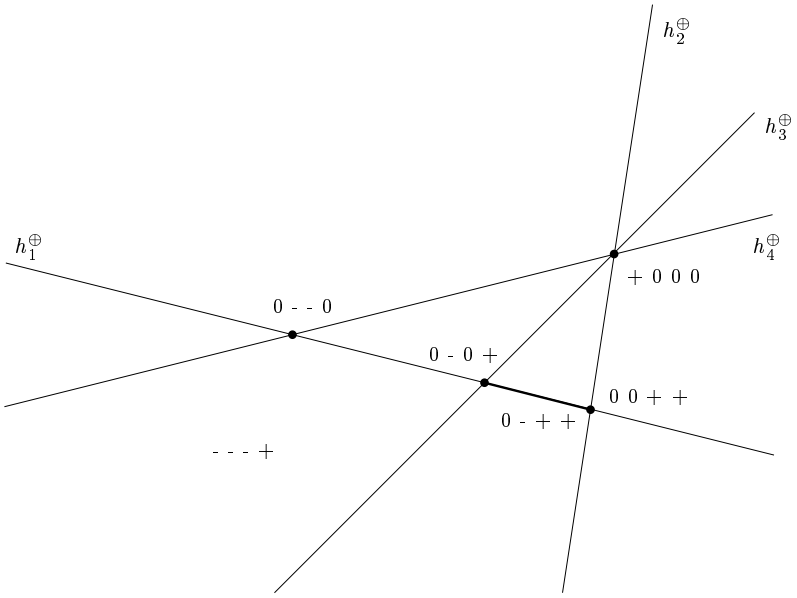

FIG. 2.1 – Un arrangement d'hyperplans en dimension 2

Dénombrement des cellules

Des hyperplans de $\mathbb{R}^n$ sont dit en position générale si toute intersection de $n+1$ de ces hyperplans est vide. Considérons m hyperplans de $\mathbb{R}^n$ en position générale, et notons $\Phi_n(m)$ le nombre de cellules de cet arrangement. Il est facile d'évaluer ce nombre [Mat, Ede87], montrant que celui-ci est indépendant des hyperplans considérés. En supprimant un hyperplan h de l'arrangement, on a $\Phi_n(m-1)$ cellules. Combien de cellules coupent l'hyperplan h que l'on rajoute ? $\Phi_{n-1}(m-1)$ en considérant la trace des autres hyperplans sur h. Comme $\Phi_n(0) = 1$ et $\Phi_1(m) = m+1$, on obtient

$$\Phi_n(m) = \binom{m}{0} + \binom{m}{1} + \ldots + \binom{m}{n}.$$

On a la majoration $\Phi_n(m) \leqslant m^n$ pour $m, n \geqslant 2$. C'est bien sûr en position générale que le nombre de cellules est maximum ; tout arrangement de m hyperplans dans $\mathbb{R}^n$

décompose donc l'espace en $O(m^n)$ cellules.

Exemple. Quel est le nombre de cellules de l'arrangement du sac-à-dos ? En dimension n, on a $2^n - 1$ hyperplans ce qui donne $O(2^{n^2})$ cellules. En fait, on a aussi une borne inférieure en $2^{\Omega(n^2)}$ [BCS97].

Triangulation d'un arrangement

La triangulation d'un polyèdre est une manière de décomposer celui-ci en simplexes. Rappelons d'abord comment est construite une triangulation Δp d'un polyèdre borné p de $\mathbb{R}^n$. Cette construction est récursive. Soit z un sommet de p et $\{f_1, \ldots, f_s\}$ l'ensemble des $(d-1)$-faces de p qui ne sont pas adjacents à z. Soit δf_i une triangulation de f_i (comme polyèdre borné de $\mathbb{R}^{n-1}$). La collection des cellules formant Δp est $\{\text{conv}(z, f), f \in \Delta f_1 \cup \ldots \cup \Delta f_n\}$, où $\text{conv}(z, f)$ est l'intérieur de l'enveloppe convexe de $z \cup f$.

Le cas d'un polyèdre non borné est un peu plus compliqué. On traite le cas d'un polyèdre non borné p ne contenant pas de droite. Pour $x \in p$, on définit $\text{cc}_x(p)$ comme l'ensemble $\{y;\ \forall \lambda \geqslant 0\ x + \lambda y \in p\}$. Cet ensemble, indépendant du point x, est appelé le *cône caractéristique* de p et on le note $\text{cc}(p)$. Tout d'abord on choisit un sommet z de p, et on applique à p l'algorithme de triangulation pour un polyèdre borné. Cependant, les cellules élémentaires obtenues ne couvrent pas entièrement le cône $C = z + \text{cc}(p)$. Le cône C est alors triangulé de la manière suivante. Soit h un hyperplan tel que $h \cap C$ soit un polyèdre borné p' de h. Soit $\Delta p'$ une triangulation de p'. On pose $\Delta C = \{\text{cone}(z, f), f \in \Delta p'\}$, où $\text{cone}(z, f)$ est l'intérieur du cône de sommet z et de base f.

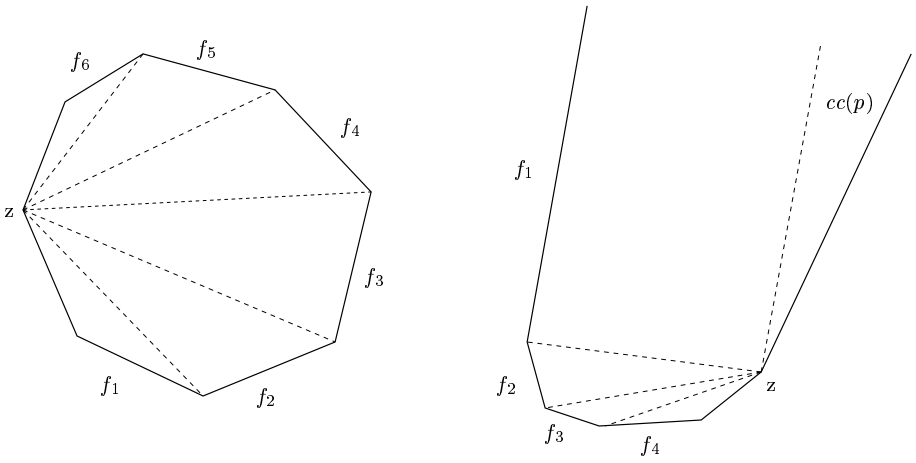

FIG. 2.2 – Triangulation de polyèdres

La programmation linéaire

Considérons le problème suivant : on se donne un ensemble d'inéquations en $(x_1, \ldots, x_n)$ du type $\sum_{i=1}^{n} a_i x_i + a_0 < 0$ et $\sum_{i=1}^{n} a_i x_i + a_0 \leqslant 0$, où les coefficients $a_i \in \mathbb{Z}$ sont donnés en bits. Existe-t-il un point de $\mathbb{R}^n$ solution de ce système ? Ce problème, appelé programmation linéaire, est dans P au sens classique [Sch86, KV00].

2.2 Non-déterminismes réel et booléen

Sur les réels avec addition, avec ou sans ordre, le non-déterminisme booléen est aussi puissant que le non-déterminisme réel dans la hiérarchie polynomiale.

Théorème 2.1 *[CK95] Sur les réels avec addition : pour tout $k \geqslant 0$, $(\mathrm{B}\Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{vs}}})^0 = (\Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{vs}}})^0$ et $\mathrm{B}\Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{vs}}} = \Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{vs}}}$. Et sur les réels avec addition et ordre : pour tout $k \geqslant 0$, $(\mathrm{B}\Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{ovs}}})^0 = (\Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{ovs}}})^0$ et $\mathrm{B}\Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{ovs}}} = \Sigma^{\mathrm{k}}_{\mathbb{R}_{\mathrm{ovs}}}$.*

Ce théorème est basé sur la proposition suivante qu'il est utile de citer.

Proposition 2.1 *[Koi94] Soit P un polyèdre de $\mathbb{R}^n$ défini par des inégalités du type $\sum_{i=1}^{n} a_i x_i + a_0 < 0$ et $\sum_{i=1}^{n} a_i x_i + a_0 \leqslant 0$, où la taille des a_i est majorée par L. Si $P \neq \emptyset$, alors P contient un point à coordonnées rationnelles dont la taille des numérateurs et dénominateurs est $(Ln)^{O(1)}$.*

Ces travaux ont permis d'établir des résultats de parties booléennes.

Théorème 2.2 *[CK95] Sur les réels avec addition et égalité : pour tout $k \geqslant 0$, $\mathrm{BP}((\Sigma^k_{\mathbb{R}_{vs}})^0) = \mathrm{BP}(\Sigma^k_{\mathbb{R}_{vs}}) = \Sigma^k$. On a également $\mathrm{BP}(\mathrm{PAR}^0_{\mathbb{R}_{vs}}) = \mathrm{BP}(\mathrm{PAR}_{\mathbb{R}_{vs}}) = \mathrm{PSPACE}$. Et avec l'ordre : pour tout $k \geqslant 0$, $\mathrm{BP}((\Sigma^k_{\mathbb{R}_{ovs}})^0) = \Sigma^k$ et $\mathrm{BP}(\Sigma^k_{\mathbb{R}_{ovs}}) = \Sigma^k/\mathrm{poly}$. On a également $\mathrm{BP}(\mathrm{PAR}^0_{\mathbb{R}_{ovs}}) = \mathrm{PSPACE}$ et $\mathrm{BP}(\mathrm{PAR}_{\mathbb{R}_{ovs}}) = \mathrm{PSPACE}/\mathrm{poly}$.*

Une technique utile sur les réels avec addition et ordre est l'encodage d'un conseil booléen de taille polynomiale dans les décimales d'un paramètre α. Sur une entrée de taille n, les $n^{O(1)}$ bits correspondant au conseil pour les entrées de taille n peuvent être calculés à partir de α en temps polynomial sur $\mathbb{R}_{ovs}$. Ceci permet de montrer le résultat suivant.

Proposition 2.2 *[Koi94] Pour toute classe de complexité C au-dessus de* P*, en particulier pour $C \in \{\mathrm{P}, \mathrm{NP}, \Sigma^k, \Pi^k, \mathrm{PH}, \mathrm{PAR}\}$, on a $C_{\mathbb{R}_{ovs}}/\mathrm{poly} = C_{\mathbb{R}_{ovs}}$.*

Comme expliqué section 1.5.1, le théorème 2.2 permet d'établir des théorèmes de transfert. Sur les réels avec addition

$$\begin{cases} \mathrm{PH}^0_{\mathbb{R}_{vs}} = (\Sigma^k_{\mathbb{R}_{vs}})^0 \Rightarrow \mathrm{PH} = \Sigma^k \\ \mathrm{PH}_{\mathbb{R}_{vs}} = \Sigma^k_{\mathbb{R}_{vs}} \Rightarrow \mathrm{PH} = \Sigma^k \end{cases}$$

et sur les réels avec addition et ordre

$$\begin{cases} (\mathrm{PH}_{\mathbb{R}_{ovs}})^0 = (\Sigma^k_{\mathbb{R}_{ovs}})^0 \Rightarrow \mathrm{PH} = \Sigma^k \\ \mathrm{PH}_{\mathbb{R}_{ovs}} = \Sigma^k_{\mathbb{R}_{ovs}} \Rightarrow \mathrm{PH}/\mathrm{poly} = \Sigma^k/\mathrm{poly}. \end{cases}$$

Tout l'objet des chapitres 4 et 5 est d'établir des formes de réciproques à ces théorèmes.

2.3 Arbres de décision linéaire

Un arbre de décision linéaire à n variables $x_1, \ldots, x_n$ sur $\mathbb{R}$ décide une partie de $\mathbb{R}^n$ de la façon suivante. Chaque nœud interne est étiqueté par un test du type $\sum_{i=1}^{n} a_i x_i + a_0 < 0$ avec $a_i \in \mathbb{R}$. Si l'entrée vérifie cette inéquation alors on descend au fils gauche sinon on descend au fils droit. Les feuilles sont étiquetées *accepte* ou *rejette*. La mesure de complexité associée à un tel arbre est sa hauteur. Si on note H l'ensemble des hyperplans-tests d'un arbre de décision linéaire, on remarque que l'ensemble décidé par cet arbre est une union de faces de l'arrangement $\mathcal{A}(H)$. Ce modèle de calcul est particulièrement utile pour prouver des bornes inférieures. En effet, si une machine de Turing sur $\mathbb{R}_{ovs}$ décide une partie de $\mathbb{R}^n$ en temps t, alors il existe un arbre de décision linéaire de profondeur au plus t décidant ce même ensemble (il suffit de développer l'arbre de calcul de la machine). Soit H un ensemble d'hyperplans de $\mathbb{R}^n$. On dit qu'un arbre de décision linéaire sur $\mathbb{R}^n$ localise un point dans $\mathcal{A}(H)$ s'il n'existe pas deux points x et y de $\mathbb{R}^n$ appartenant à deux faces distinctes de cet arrangement et suivant le même chemin dans l'arbre (cette notion est donc indépendante des étiquettes des feuilles).

3. Arbres de décision linéaire pour les arrangements d'hyperplans

Nous présentons deux méthodes pour construire des arbres de décision linéaire peu profonds permettant de reconnaître des unions d'hyperplans de $\mathbb{R}^n$. Historiquement, la première méthode est due à Meyer auf der Heide qui construit une famille d'arbres de décision linéaire de profondeur polynomiale décidant le sac-à-dos [MadH84], détruisant ainsi l'espoir de prouver une borne inférieure superpolynomiale pour ce problème par l'intermédiaire de ce modèle de calcul. Ce résultat est ensuite généralisé à d'autres arrangements d'hyperplans [MadH88]. Cependant la profondeur de ces arbres est multiple de l'inverse du rayon de la plus petite boule inscrite dans l'arrangement. Ceci pose la question de savoir si on peut obtenir une borne indépendante de ce rayon. La réponse à cette question, positive, est apportée par Meiser [Mei93] qui propose une construction complètement différente pour reconnaître des unions d'hyperplans de $\mathbb{R}^n$, basée sur la notion de VC-dimension. Cette méthode permet d'établir le théorème suivant.

Théorème 3.1 *[Mei93] Soit $H = \{h_1, \ldots, h_m\}$ un ensemble d'hyperplans de $\mathbb{R}^n$.*

(i) Le problème de localisation dans $\mathcal{A}(H)$ peut être résolu par un arbre de décision linéaire de profondeur $(n \log m)^{O(1)}$.

(ii) De plus, si les coefficients de $h_1, \ldots, h_m$ sont des entiers de $\{-q, \ldots, q\}$, on peut construire un tel arbre avec des fonctions tests à coefficients entiers de tailles $(n \log q)^{O(1)}$.

Nous donnons dans un premier temps une démonstration du théorème 3.1 par la méthode de Meiser. Nous décrivons ensuite la construction de Meyer auf der Heide. Cette méthode, moins générale que la précédente, se révèle plus fructueuse pour les théorèmes de transfert au chapitre 4.

D'après le lemme suivant, construire un arbre localisant un point dans l'arrangement $\mathcal{A}(h_1, \ldots, h_m)$ n'est pas beaucoup plus difficile que de construire un arbre reconnaissant l'union $h_1 \cup \cdots \cup h_m$.

Lemme 3.1 *[MadH88] Soit H un ensemble d'hyperplans de $\mathbb{R}^n$. Si l'union de ces hyperplans peut être décidée par un arbre de décision linéaire $\mathcal{T}_1$ de profondeur d, alors le problème de localisation dans l'arrangement de H peut être résolu par un arbre $\mathcal{T}_2$ de profondeur $2d$ n'utilisant comme hyperplans tests que ceux présents dans l'arbre $\mathcal{T}_1$.*

3.1 La méthode du cutting

Nous allons expliquer la méthode de Meiser pour construire un arbre de décision linéaire reconnaissant l'union d'hyperplans $h_1 \cup \cdots \cup h_m$ dans $\mathbb{R}^n$. Nous en présentons dans un premier temps une version simplifiée permettant d'établir plus simplement le théorème 3.1. Comme il est intéressant d'étudier la profondeur de cet arbre, nous présentons ensuite une méthode plus rapide, proche de celle décrite dans l'article de Meiser mais où nous avons supprimé l'utilisation du produit puisque notre modèle d'arbre ne le permet pas. Précisons également l'utilisation d'une borne récente pour les ε-nets ce qui explique que la hauteur des arbres obtenue ici est inférieure à celle annoncée dans l'article de Meiser.

VC-dimension et ε-nets

Nous avons besoin de définir quelques notions de VC-dimension [KV94, Mat98]. Un système géométrique S, ou *range space*, est un couple (X, R) où X est un ensemble et R un ensemble de parties de X. Soit $A \subset X$, A fini. On note $\Pi_R(A) = \{A \cap r, r \in R\}$. Si $\Pi_R(A) = \mathcal{P}(A)$ (l'ensemble des parties de A), on dit que A est pulvérisé par R. La VC-dimension de S est le sup des $d \in \mathbb{N}$ tels qu'il existe un ensemble A de cardinal d pulvérisé par R.

Soit A une partie finie de X. Soit ε fixé dans $[0, 1]$. On note $R_{A,\varepsilon} = \{r \in R, |A \cap r|/|A| > \varepsilon\}$. L'ensemble $N \subset A$ est un ε-net de A (pour R) s'il contient au moins un point de chaque élément de $R_{A,\varepsilon}$. Cette notion a été introduite par Haussler et Welzl [HW87]. L'existence de *petits* ε-nets pour des systèmes géométriques de VC-dimension finie est le point crucial : pour (X, R) de VC-dimension $d \geqslant 1$, $A \subset X$ fini et $0 < \varepsilon < 1$, il existe un ε-net de A (pour R) de taille $\lceil 8d/\varepsilon \log(8d/\varepsilon) \rceil$. Ce résultat a été amélioré ensuite par Blumer, Ehrenfeucht, Haussler et Warmuth.

Théorème 3.2 *[BEHW89] Soit (X, R) de VC-dimension $d \geqslant 1$. Soit $A \subset X$, A fini, et $0 < \varepsilon < 1$. Il existe un ε-net de A pour R de taille $O(d/\varepsilon \log(1/\varepsilon))$.*

Notons H^n l'ensemble des hyperplans de $\mathbb{R}^n$, et C^n l'ensemble des simplexes de $\mathbb{R}^n$ (éventuellement non bornés, éventuellement ouverts sur certains côtés). Pour $c \in C^n$, on note $S_c = \{h \in H^n, h \cap c \neq \emptyset\}$. Soit $\mathcal{S}^n = \{S_c, c \in C^n\}$.

Lemme 3.2 *[Mei93] VC-dim$(H^n, \mathcal{S}^n) = O(n^2 \log n)$.*

On combinant les résultats de théorème 3.2 et du lemme 3.2, on obtient le théorème suivant qui rend possible la méthode de "diviser pour régner" pour la localisation dans les arrangements d'hyperplans. Consulter également [Cha93] à ce sujet.

Théorème 3.3 *[Mei93, Cla87] Soit ε fixé, $0 < \varepsilon < 1$. Pour tout ensemble H de m hyperplans de $\mathbb{R}^n$, il existe un ensemble $R \subset H$ de cardinal $O(n^2 \log n)$ tel qu'aucun simplexe ouvert de $\mathbb{R}^n$ n'intersectant pas R ne soit intersecté par plus de εm hyperplans de H. En particulier, aucune cellule de l'arrangement triangulé $\Delta\mathcal{A}(R)$ n'est intersectée par plus de εm hyperplans de H. Un tel R est appelé ε-cutting de H.*

Construction de l'arbre par récurrence

L'algorithme se déroule de la manière suivante. Soit R un sous-ensemble de H donné par le théorème 3.3 pour $\varepsilon = 1/2$. Posons $r = |R|$. La position d'une entrée x dans $\mathcal{A}(R)$ peut être déterminée par un arbre de profondeur $2r$ en testant tour à tour la position de x par rapport à chacun des hyperplans de R. Les feuilles correspondant aux hyperplans de R peuvent être étiquetées *accepte*. Si x ne se trouve sur aucun des hyperplans de R, il appartient à une cellule c de $\mathcal{A}(R)$. On cherche alors à déterminer un simplexe n'intersectant pas R et contenant x. Pour cela, on localise x dans une triangulation du polyèdre c : ceci est expliqué ci-dessous. D'après le théorème 3.3, l'ensemble H' des hyperplans de H intersectant un simplexe de la triangulation de c est de cardinal au plus $|H|/2$. On localise alors x dans $\mathcal{A}(H')$ récursivement.

Localisation de point dans un polyèdre triangulé

On sait que le point d'entrée x se trouve dans une cellule c donnée de $\mathcal{A}(R)$, et on veut maintenant déterminer un simplexe s (éventuellement non borné) n'intersectant aucun hyperplan de R, et tel que x appartienne à l'adhérence de s. Plus précisément, on veut que l'ensemble des points de c suivant le même chemin que x dans l'arbre de décision linéaire que l'on va construire soit inclus dans un simplexe n'intersectant pas R. La méthode consiste à localiser x dans une triangulation du polyèdre c. Pour cela, on va descendre dans la hiérarchie d'une triangulation : voici comment on procède si c est un polyèdre borné. Soit z^0 un sommet de c, et $\{f_1^0, \ldots, f_{n_0}^0\}$ l'ensemble des $(n-1)$-faces de c qui ne sont pas adjacentes à z^0. À la première étape, on détermine i tel que $x \in \mathrm{conv}(z^0, f_i^0)$. Comme les n_0 faces considérées sont délimitées par au plus r hyperplans, et $n_0 \leqslant r$, ceci peut être fait par un arbre de profondeur $O(r^2)$. Soit z^1 un sommet de f_i^0, et $\{f_1^1, \ldots, f_{n_1}^1\}$ l'ensemble des $(n-2)$-faces de f_i^0 qui ne sont pas adjacentes à z^1. Dans la deuxième étape, on détermine en profondeur $O(r^2)$ une face f_i^1 tel que $x \in \mathrm{conv}(z^0, z^1, f_i^1)$. On continue à descendre ainsi dans la hiérarchie de la triangulation, et après $n-1$ étapes on détermine finalement un simplexe s inclus dans c tel que x soit dans l'adhérence de s. Si le polyèdre c est non borné, il faut tenir compte des spécificités d'une triangulation de c : on doit en particulier traiter le cas où x est dans le cône caractéristique de c. Dans tous les cas, comme chaque étape se fait en profondeur $O(r^2)$, la profondeur de l'arbre construit ci-dessus localisant x dans c est $O(nr^2)$.

Analyse de l'algorithme

Soit $T(m, n)$ la profondeur de l'arbre décrit ci-dessus décidant l'union de m hyperplans dans $\mathbb{R}^n$. On a $T(m, n) \leqslant O(nr^2) + T(m/2, n)$ pour $m > r$ (où r est

donné par le théorème 3.2), $T(m,n) = O(r)$ sinon. Ainsi $T(m,n) = O(nr^2 \log m) = O(n^5 \log^2 n \log m)$. Ceci achève la démonstration du point (i) du théorème 3.1.

On suppose maintenant que les hyperplans de H ont des coefficients entiers dans $[-q,q]$. Chaque hyperplan apparaissant dans les fonctions test de l'arbre est la clôture affine d'une union de faces de $\mathcal{A}(H)$. Chacun de ces hyperplans est donc la clôture affine de n sommets $p_1,\ldots,p_n$ de $H' = H \cup \{x_1 = 0, x_1 = 1, \ldots, x_n = 0, x_n = 1\}$. Par la règle de Cramer, p_i est un point rationnel $(a_{i1}/u_i, \ldots, a_{in}/u_i)$, avec $|a_{ij}|, |u_i| \leqslant B = q^n n^{n/2}$. Une équation de l'hyperplan h contenant $p_1,\ldots,p_n$ est $\det(x - p_1, p_2 - p_1, \ldots, p_n - p_1) = 0$. En multipliant les colonnes de ce déterminant par $u_1, (u_1u_2), \ldots, (u_1u_n)$ respectivement, on obtient une équation de h à coefficients entiers majorés par $n!(2B)^{2n}$. La longueur de chaque coefficient est majorée par $n^2 \log n + 2n^2 \log q + O(n^2)$. Ceci complète la démonstration du théorème 3.1.

Localisation dans un polyèdre triangulé : méthode rapide

Cette section expose la méthode de localisation dans un polyèdre triangulé telle qu'elle est décrite dans l'article de Meiser. La profondeur de l'arbre et la taille des coefficients de cette nouvelle construction sont ensuite évaluées. On cherche à localiser le point d'entrée $x = x^0$ dans une triangulation du polyèdre f^0 défini par les hyperplans $H = \{h_1,\ldots,h_r\}$. Notons p^0 le point de base servant à trianguler f^0. Pour savoir dans quelle pyramide se trouve x^0, il suffit de déterminer quel est le premier hyperplan de H intersecté par le rayon (p^0, x^0).

Lemme 3.3 *Étant donnés un polyèdre $h_1^{\oplus} \cap \ldots \cap h_r^{\oplus}$ et un de ses sommets p, il existe un arbre de décision linéaire en x de profondeur $O(r)$ qui décide quel est le premier hyperplan de $\{h_1,\ldots,h_r\}$ intersecté par le rayon (p,x).*

Démonstration. Soit $p = (p_1,\ldots,p_n)$ avec $p_i = y_i/e$. Pour l'hyperplan h_i d'équation $a_1x_1 + \ldots + a_nx_n + a_{n+1} = 0$, soit λ_i (s'il existe) tel que $h_i(p + \lambda_i(x-p)) = 0$. On cherche l'indice i_0 tel que $\lambda_{i_0} = \min\{\lambda_i, \lambda_i > 0\}$. L'équation ci-dessus s'écrit $u_i\lambda_i = v_i$ avec $u_i = a_1(x_1 - p_1) + \ldots + a_n(x_n - p_n)$ et $v_i = -a_1p_1 - \ldots - a_np_n - a_{n+1}$. On sait que $v_i \neq 0$ pour tout i car on ne considère que les hyperplans qui ne contiennent pas p. Pour tester l'existence de λ_i, il suffit de tester l'égalité à 0 de u_i. Si $u_i \neq 0$, λ_i existe et on connaît en plus le signe de u_i. On connaît donc le signe de λ_i puisque v_i est une constante. Ensuite, il est facile de comparer λ_i et λ_j en effectuant le test $\beta u_i v_j > u_j v_i$, avec $\beta = \text{signe}(u_iu_j)$. □

Le lemme 3.3 permet de déterminer quel est le premier hyperplan intersecté par le rayon (p^0, x^0). Notons le h_{i_0} et appelons $x^1 = (p^0, x^0) \cap h_{i_0}$. Soit f^1 la face de f^0 contenant x^1, et soit p^1 le point de base pour la triangulation de f^1. On doit maintenant déterminer quel hyperplan est intersecté en premier par le rayon (p^1, x^1) sur h_{i_0}. Cette opération peut s'effectuer sans multiplication comme l'explique le lemme suivant.

Lemme 3.4 *Soit $H = \{h_i\}_{i\in I}$ en ensemble d'hyperplans dans $\mathbb{R}^n$. Soit h un autre hyperplan. Soit p et x deux points distincts de $\mathbb{R}^n$. Soit $\{y\} = (px) \cap h$. Alors pour un point quelconque q de h, on la propriété suivante : le rayon (q,y) intersecte les hyperplans h_i (ou $h_i \cap h$) dans le même ordre que (q,x) intersecte les $\{(h_i \cap h) + \mathbb{R}(px)\}_{i\in I}$.*

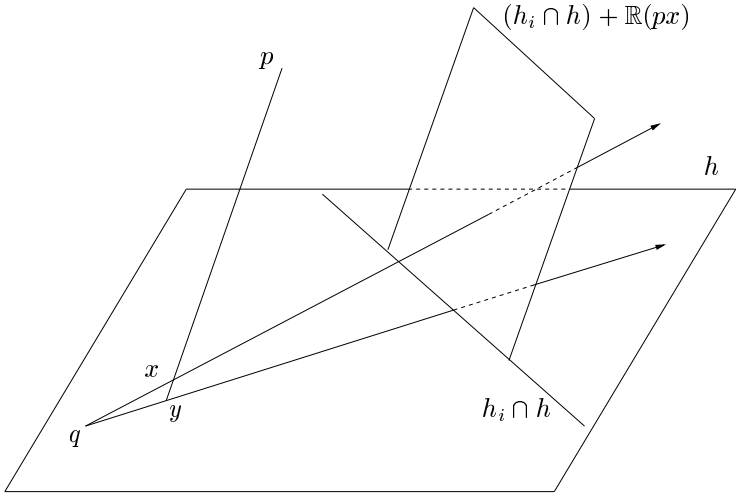

FIG. 3.1 – Localisation rapide dans une triangulation

Démonstration. Notons $t_i = (q, y) \cap h_i$ et $w_i = (q, x) \cap (h_i \cap h + \mathbb{R}px)$ – voir la figure 3.1. Le théorème de Thalès donne $\overline{qw_i}/\overline{qx} = \overline{qt_i}/\overline{qy}$. □

Ceci permet de déterminer l'hyperplan h_{i_2} par un arbre en x. À l'étape j, on cherche à décider quel hyperplan $h_{i_{j-1}}$ parmi $\{h_i\}_{i \in I_j}$ est intersecté en premier par le rayon (p^{j-1}, x^{j-1}). Or, par application successive du lemme 3.4, on sait qu'il suffit de déterminer quel est le premier hyperplan intersecté par (p^{j-1}, x^0) parmi $\{(h_i \cap h_{i_0} \cap \ldots \cap h_{i_{j-2}}) + \text{Vect}\{p^{j-2}x^{j-2}, \ldots, p^1x^1, p^0x^0\}\}_{i \in I_{j-1}}$. Par le lemme 3.3, on sait comment construire un arbre de décision linéaire en x décidant cela. Ceci termine la construction de la méthode rapide de localisation dans un polyèdre triangulé.

Analysons la profondeur de cet arbre. Le nouvel algorithme pour localiser un point dans la triangulation d'un polyèdre donne un arbre de hauteur $O(nr)$. La nouvelle relation de récurrence pour la complexité de l'algorithme total s'écrit donc $T(n, m) = O(nr) + T(n, m/2)$. Avec $r = O(n^2 \log(n))$, on obtient cette fois $T(n, m) = O(n^3 \log n \log m)$. Dans le cas où les hyperplans sont à coefficients entiers dans $[-q, q]$, l'analyse précédente concernant la taille des coefficients est encore valide.

3.2 La méthode de Meyer auf der Heide

Nous décrivons maintenant une autre méthode pour construire un arbre reconnaissant une union d'hyperplans à coefficients entiers dans $\{-q, \ldots, q\}$ d'abord sur $[-1, 1]^n$ puis sur $\mathbb{R}^n$. Nous montrons que l'arbre ainsi construit à des hyperplans test à coefficients de taille $(n \log q)^{O(1)}$.

Décider une union d'hyperplans sur un cube

Soit $q \in \mathbb{N}$. On décrit la construction d'un arbre de décision linéaire pour décider sur $[-1,1]^n$ une union d'hyperplans $\mathcal{H}_q^n$. Étant donné un point $y \in \mathbb{R}^n$ et un ensemble $A \subset \mathbb{R}^n$, on note $\mathrm{Aff}(y,A)$ la clôture affine de $\{y\} \cup A$, et $P(y,A) = \{\lambda y + (1-\lambda)x, x \in A, \lambda < 1\}$ la pyramide de sommet y et de base A. Une récurrence sur la dimension est rendue possible par le lemme suivant.

Lemme 3.5 *[MadH84] Soit $S = \{h_1, \ldots, h_p\}$ un ensemble d'hyperplans de $\mathbb{R}^n$ tel que l'intersection $I = \bigcap_{i=1}^p h_i$ soit non vide. Soit A un polyèdre sur un hyperplan h_0 qui ne contient pas I, et soit s un point de $I \setminus h_0$. Si un arbre de décision linéaire décide $L' = \bigcup_{i=1}^p h_i \cap h_0$ sur A, alors l'arbre obtenu en remplaçant chaque test h' par $\mathrm{Aff}(s,h')$ (avec le signe approprié) décide $L = \bigcup_{i=1}^p h_i$ on $P(s,A)$.*

Démonstration. Soit $x \in P(s,A)$. Remarquer l'équivalence

$$x \in h_1 \cup \ldots \cup h_p \Leftrightarrow (sx) \cap h_0 \in (h_1 \cap h_0) \cup \ldots \cup (h_p \cap h_0)$$

permet de montrer le lemme. □

Un nombre $r > 0$ est une *granularité* d'un ensemble d'hyperplans $h_1, \ldots, h_k$ de $\mathbb{R}^n$ si, pour toute boule B de rayon r, soit $\{h_i, h_i \cap B \neq \emptyset\} = \emptyset$ soit $\bigcap_{h_i \cap B \neq \emptyset} h_i \neq \emptyset$. Notons $r_{n,q}$ une granularité de $\mathcal{H}_q^n$. Il est montré dans [MadH88] que l'on peut prendre $1/r_{n,q} = n^{n^2} q^{2n^2} 2^{O(n^2)}$.

L'arbre est construit par récurrence sur la dimension. Si $n = 1$ on décide si $x \in \mathcal{L}_n$ par une recherche dichotomique. On suppose maintenant $n > 1$, et on pose $H_n^0 = \mathcal{H}_q^n$. On subdivise $C_n^1 = [-1,1]^n$ en petits cubes de rayons plus petits que $r_{n,q}$. On prend par exemple des cubes d'arête la plus grande puissance de 2 inférieure à $2r_{n,q}/\sqrt{n}$. Par une recherche dichotomique sur chaque coordonnée, on détermine dans quel petit cube c_n^1 se trouve x. Soit $H_n^1 = \{h \in H_n^0,\ h \cap c_n^1 \neq \emptyset\}$. Deux cas se présentent :

(i) $H_n^1 = \emptyset$.

(ii) Sinon, $\bigcap_{h \in H_n^1} h \neq \emptyset$ par définition de la granularité.

Dans le premier cas, on rejette l'entrée x. Sinon, soit $s_n^1 \in \bigcap_{h \in H_n^1} h$. On détermine une $(n-1)$-face f_n^1 du cube C_n^1 telle que $x \in P(s_n^1, f_n^1)$: le cube C_n^1 de dimension n a $2n$ faces de dimension $n-1$, on peut donc faire tous ces tests tour à tour. Cette face est sur un hyperplan g d'équation $x_i = 1$ ou $x_i = -1$. Par hypothèse de récurrence, on construit un arbre de décision linéaire qui décide si un point z donné en entrée est sur l'union des hyperplans $\{h \cap g,\ h \in H_n^1\}$ sur le cube $f_n^1 = C_n^1 \cap g$ de dimension $n-1$. Par le lemme 3.5, ceci permet d'obtenir un arbre qui décide $\bigcup_{h \in H_n^1} h$ sur $P(s_n^1, f_n^1)$.

Décider une union d'hyperplans sur $\mathbb{R}^n$

Comment reconnaître une union d'hyperplans sur tout $\mathbb{R}^n$ à partir de l'algorithme précédent ? La méthode consiste à cône r l'arrangement. Soit $\tilde{H}_n$ la famille d'hyperplans de $\mathbb{R}^{n+1}$ définis par les équations de la forme $\sum_{i=1}^n a_i x_i - b x_{n+1} = 0$ où b et les a_i sont des entiers majorés en valeur absolue par $2^{t(n)}$ et $(a_1, \ldots, a_n, b) \neq 0$. $\tilde{H}_n$ est obtenu à

partir de H_n en transformant chaque hyperplan $h \in H_n$ de $\mathbb{R}^n$ d'équation $\sum_{i=1}^n a_i x_i = b$ en l'hyperplan $\tilde{h}$ de $\mathbb{R}^{n+1}$ d'équation $\sum_{i=1}^n a_i x_i - b x_{n+1} = 0$. Pour décider $\tilde{H}_n$ sur $\mathbb{R}^{n+1}$, on procède de la manière suivante. On détermine une n-face f du cube $[-1,1]^{n+1}$ tel que $x \in P(0,f)$. Il suffit ensuite d'appliquer le lemme 3.5 à un arbre décidant sur f si $x \cap f \in \tilde{H}_n$. En posant $x_{n+1} = 1$ dans cet arbre, on obtient un arbre décidant H_n sur $\mathbb{R}^n$.

Hauteur de l'arbre

Soit $T(n,q)$ la profondeur d'un arbre ainsi construit reconnaissant une union d'hyperplans de $\mathcal{H}_q^n$ dans $\mathbb{R}^n$. Pour évaluer précisément la hauteur de l'arbre, notons $T(n,q,q')$ la hauteur d'un arbre construit ci-dessus reconnaissant une union d'hyperplans de $\mathbb{R}^n$ d'équations $\sum_{i=1}^n a_i x_i + b = 0$ avec $|a_i| \leqslant q$ et $|b| \leqslant q'$. On a $T(n,q,q') \leqslant n \log(1/r_{n,Q}) + O(n^2) + T(n-1, q, q+q')$ où $Q = \max(q,q')$. Le premier terme correspond aux n dichotomies successives, le second à la localisation du point d'entrée dans les pyramides, et le troisième à l'étape de récursion. De plus $T(1,q) = O(\log(1/r_{n,q}))$. Ceci donne $T(n,q,q) \leqslant n^2 log(1/r_{n,nq}) + O(n^3)$. On obtient $T(n,q) = O(n^4(\log n + \log q))$.

Taille des coefficients des tests

On a besoin d'une borne sur la taille des coefficients des hyperplans de $\mathbb{R}^n$ obtenus comme clôture affine de points de $\mathbb{R}^n$.

Lemme 3.6 *Soit $v_1, \ldots, v_{n-1}$ des vecteurs linéairement indépendants de $\mathbb{R}^n$ à coordonnées entières majorées par A en valeur absolue. L'hyperplan* $\mathrm{Vect}(v_1, \ldots, v_{n-1})$ *a une équation à coefficients entiers majorés par $A^{n-1}(n-1)^{(n-1)/2}$ en valeur absolue.*

Démonstration. Une équation de cet hyperplan est $\det(v_1, \ldots, v_{n-1}, x) = 0$. Ses coefficients sont des mineurs de tailles $(n-1) \times (n-1)$ dont les coefficients sont bornés par A en valeur absolue. Appliquer l'inégalité de Hadamard donne le résultat. □

Corollaire 3.1 *Soit $a_1, \ldots, a_n$ des points de $\mathbb{R}^n$ tels que leur clôture affine h soit un hyperplan.*

(i) Si les a_i ont des coordonnées entières majorées par B en valeur absolue, h a une équation à coefficients entiers majorés par $B^n n^{n/2}$ en valeur absolue.

(ii) Si les a_i ont des coordonnées rationnelles à numérateurs et dénominateurs majorés par C en valeur absolue, h a une équation avec des coefficients entiers majorés par $C^{n^2+n} n^{n/2}$ en valeur absolue.

Démonstration. Pour (i), soit $\overline{a_i}$ un point de $\mathbb{R}^{n+1}$ avec ses n premières coordonnées égales à celles de a_i, et sa dernière coordonnée égale à 1. Par le lemme 3.6, $\mathrm{Vect}(\overline{a_1}, \ldots, \overline{a_n})$ a une équation de la forme $\alpha_1 x_1 + \cdots + \alpha_n x_n + \alpha_{n+1} x_{n+1} = 0$ où les α_i sont des entiers bornés par $B^n n^{n/2}$ en valeur absolue. Une équation de h est alors $\alpha_1 x_1 + \cdots + \alpha_n x_n + \alpha_{n+1} = 0$.

Pour le point (ii), soit $a_i = (p_{ij}/q_{ij})_{1 \leqslant j \leqslant n}$ où p_{ij} et q_{ij} sont des entiers majorés par C en valeur absolue. On fait maintenant le changement de variable $y_j = \prod_{i=1}^n q_{ij} \cdot x_j$

pour $j = 1, \ldots, n$. Dans les coordonnées y, les composantes des a_i sont des entiers majorés par C^n en valeur absolue. Par le point (i), dans ces nouvelles coordonnées, h a une équation de la forme $\alpha_1 y_1 + \cdots + \alpha_n y_n + c = 0$ avec des coefficients majorés par $C^{n^2} n^{n/2}$ en valeur absolue. On obtient la borne annoncée en repassant dans les coordonnées en x. □

Dans un arbre en dimension n, les hyperplans tests dans les nœuds sont de la forme $f_n \circ \ldots \circ f_{s+1}(h_s)$ avec $f_i : h \mapsto \mathrm{Aff}(z_i, h)$ et h_s un hyperplan crée en dimension s. On appelle "hyperplan initial" un tel hyperplan. Les sommets des pyramides sont les points y qui apparaissent dans les fonctions $f : h \mapsto \mathrm{Aff}(y, h)$. Un tel point y est dans $I = \bigcap_{h \in H} h$, pour un certain $H \subset \mathcal{H}_q^n$. Si $\dim I > 0$, on ajoute des équations de la forme $x_i = 0$ pour obtenir une intersection réduite à un point. On obtient par la règle de Cramer : $y = (n_1/d, \ldots, n_s/d)$ où $|d|, |n_i| \leqslant (2^q)^n n^{n/2}$. Les hyperplans initiaux sont de trois types :

(a) L'intersection x_0 de $h \in \mathcal{H}_q^n$ avec un espace affine de dimension un de la forme $\{x_{i_1} = \varepsilon_1, \ldots, x_{i_{n-1}} = \varepsilon_{n-1}\}$, où $\varepsilon_i \in \{-1, 1\}$.

(b) Les hyperplans utilisés pour tester dans quelle pyramide le point se trouve. Ils sont de la forme $h = \mathrm{Aff}(y, f)$ où f est l'intersection de deux faces du cube $[-1, 1]^s$ dans l'espace affine de dimension s défini par $\{x_{i_1} = \varepsilon_{i_1}, \ldots, x_{i_{n-s}} = \varepsilon_{i_{n-s}}\}$, où $\varepsilon_i \in \{-1, 1\}$.

(c) Les hyperplans utilisés pour décomposer le cube unité en petits cubes.

Comme la borne (c) domine les bornes de (a) et (b), on conclut que dans $\mathbb{R}^n$ chaque hyperplan test est la clôture affine de n points à coordonnées rationnelles à numérateurs et dénominateurs majorés par $\sqrt{n}/2(1/r_n)$. Appelons C cette borne. Par le corollaire 3.1, chaque hyperplan test a une équation à coefficients entiers majorés par $C^{n^2+n} n^{n/2}$ en valeur absolue. Les longueurs de ces coefficients sont donc $(n \log q)^{O(1)}$.

3.3 Bornes inférieures sur la hauteur des arbres

Il existe également des bornes inférieures sur la hauteur des arbres de décision linéaire décidant certains langages. Citons le résultat de Dobkin et Lipton.

Théorème 3.4 *[DL78] Soit $A \subset \mathbb{R}^n$ constructible avec l'addition et l'ordre (et des paramètres). Soit N le maximum du nombre de composantes connexes de A et de $\mathbb{R}^n \setminus A$. Alors tout arbre de décision linéaire décidant A a une profondeur au moins* $\log N$.

La profondeur minimale l_n d'un arbre décidant $\mathrm{KP}_{\mathbb{R}} \cap \mathbb{R}^n$ vérifie donc

$$\Omega(n^2) \leqslant l_n \leqslant O(n^4 \log n).$$

Il n'existe pas d'encadrement plus précis de l_n connu pour l'instant. Cependant, il existe des travaux établissant des bornes inférieures du même ordre pour le sac-à-dos sur des modèles d'arbres plus puissants. Citons le résultat de Ben-Or sur la hauteur des arbres de calcul avec multiplication [BO83], celui de Yao concernant des arbres de calcul parallèles avec multiplication [Yao82], ainsi que ceux de Grigoriev et Karpinski sur les

arbres de calcul probabilistes [GK97]. Citons pour terminer [Gri97] sur un algorithme presque optimal en dimension fixée pour décider une union d'hyperplans sur un modèle de calcul parallèle sur le corps ordonné des réels.

4. Transferts sur les réels avec addition et ordre

Dans ce chapitre, nous montrons comment simuler les constructions de Meyer auf der Heide et de Meiser avec des algorithmes uniformes sur les réels utilisant des oracles booléens. Ceci permet d'établir des théorèmes de transfert. Les résultats de ce chapitre proviennent de [FK98], [Fou98] et [FK00].

4.1 Le problème de la localisation de point dans un arrangement

Soit t un polynôme. On dit qu'un algorithme résout le problème de localisation associé à la famille d'arrangements d'hyperplans $\mathcal{H}_t$ si, sur une entrée $(x_1, \ldots, x_n) \in \mathbb{R}^n$, il calcule un système

$$\mathcal{S} = \begin{cases} f_i(y) < 0 & i = 1, \ldots, p \\ g_j(y) = 0 & j = 1, \ldots, r \end{cases}$$

composé de $p + r = n^{O(1)}$ équations ou inéquations affines à coefficients entiers de tailles $n^{O(1)}$ tel que l'ensemble des points $P_{\mathcal{S}}$ de $\mathbb{R}^n$ satisfaisant $\mathcal{S}$ soit inclus dans une face de $\mathcal{A}(\mathcal{H}^n_{t(n)})$, et $x \in P_{\mathcal{S}}$.

La méthode de perturbation

Dans la suite on cherche à localiser un point $x = (x_1, \ldots, x_n)$ dans $\mathcal{A}(\mathcal{H}^n_{t(n)})$ en simulant les méthodes données par les constructions des arbres du chapitre précédent. Or on n'a pas explicité de manière constructive comment passer d'un arbre reconnaissant $\mathcal{L}^n_{t(n)}$ à un arbre localisant un point dans l'arrangement $\mathcal{A}(\mathcal{H}^n_{t(n)})$. Pour localiser un point en ne simulant que la partie "reconnaissance d'une union d'hyperplans", on utilise une méthode de perturbation. Soit $\tilde{x} = (x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n)$ où $\varepsilon_1, \ldots, \varepsilon_n$ sont des infiniment petits positifs vérifiant $\varepsilon_1 \ll \varepsilon_2 \ll \cdots \ll \varepsilon_n$. Que se passe-t-il si on lance sur l'entrée

$\tilde{x}$ notre algorithme simulant un arbre reconnaissant l'union d'hyperplans $\mathcal{L}^n_{t(n)}$? Bien sûr cette entrée est rejetée mais la collection de tous les tests effectués constitue un système localisant $\tilde{x}$. Notons $\tilde{S} = \{f_1(y) < 0, \ldots, f_q(y) < 0\}$ ce système. Alors pour chaque i, on teste si $f_i(x) < 0$ ou $f_i(x) = 0$. Ceci nous donne un nouveau système S qui localise le point x dans $\mathcal{A}(\mathcal{H}^n_{t(n)})$. De plus, ce système est bien composé de $n^{O(1)}$ équations et inéquations affines (puisque la profondeur de l'arbre est polynomiale) à coefficients entiers de tailles $n^{O(1)}$ (par l'analyse de la taille des coefficients).

Notons qu'on ne dispose pas d'éléments infiniment petits dans notre structure (dont l'ensemble de base est $\mathbb{R}$). Les calculs dans $\mathrm{Vect}\{1, \varepsilon_1, \ldots, \varepsilon_n\}$ sont simulés en représentant un élément $a_0 + a_1\varepsilon_1 + \ldots + a_n\varepsilon_n$ par le $(n+1)$-uple de réels $(a_0, \ldots, a_n)$. L'ordre se traduit sur cette représentation par l'ordre lexicographique.

Calcul formel

Soit A un langage de $C^0_{\mathbb{R}_{ovs}}$, avec $C \in \{\mathrm{P}, \Sigma^k, \mathrm{PAR}\}$. Considérons le problème de décision booléen associé $\hat{A}$. Ce problème prend en entrée $\langle 1^n, S\rangle$, où S est un système d'inéquations linéaires à coefficients entiers dans $\mathbb{R}^n$. Notons P_S l'ensemble des points de $\mathbb{R}^n$ solutions de ce système. Le langage $\hat{A}$ doit accepter les entrées telles que $P_S \subset A$, et refuser celles qui vérifient $P_S \cap A = \emptyset$. La réponse dans les autres cas est indifférente.

Lemme 4.1 *Pour $A \in C^0_{\mathbb{R}_{ovs}}$ avec $C \in \{\mathrm{P}, \Sigma^k, \mathrm{PAR}\}$, il existe un langage $\hat{A}$ associé dans $C_{\mathbb{Z}_2}$.*

Démonstration. Considérons A un langage de $\mathrm{P}^0_{\mathbb{R}_{ovs}}$. Soit S un système de $\mathbb{R}^n$ l'entrée de l'algorithme. La méthode consiste à simuler le calcul de la machine réelle décidant A sur l'entrée formelle $X_1, \ldots, X_n$. Le contenu d'une case de la machine réelle est représentée par l'équation $\sum_{i=1}^n a_i X_i + a_0$, où les a_i sont des entiers (majorés par 2^j en valeur absolue après j pas de calcul). Simuler l'addition et la soustraction de deux cases du ruban est aisé. Pour simuler le test $\sum_{i=1}^n a_i X_i + a_0 > 0$, il suffit de vérifier si

$$P_{\mathcal{S}} \cap \{(x_1, \ldots, x_n) \in \mathbb{R}^n,\ \sum_{i=1}^n a_i x_i + a_0 > 0\}$$

est vide ou non (le test est positif dans ce dernier cas). C'est un problème de programmation linéaire, ce problème est donc dans P. Ainsi on a un langage $\hat{A} \in \mathrm{P}$.

Cette preuve se traduit directement au cas $C = \mathrm{PAR}$. Traitons maintenant le cas $C = \Sigma^k$, $k \geqslant 1$. Soit $A \in (\Sigma^k_{\mathbb{R}_{ovs}})^0$. Par l'équivalence des non-déterminisme réel et booléen (théorème 2.1), il existe un polynôme p et $B \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$ tels que pour tout $x \in \mathbb{R}^n$, $x \in A$ si et seulement si

$$Q_1 y_1 \in \{0,1\}^{p(n)} \ldots Q_k y_k \in \{0,1\}^{p(n)} \langle x, y_1, \ldots, y_k\rangle \in B$$

où les quantificateurs Q_i alternent avec $Q_1 = \exists$. On applique le résultat précédent à $B(y_1, \ldots, y_k)$, et comme la méthode précédente se paramétrise bien en $y_1, \ldots, y_k$ on obtient un algorithme uniforme $\hat{B}(y_1, \ldots, y_k)$. Donc pour un système S dans $\mathbb{R}^n$ définissant

un polyèdre inclus dans A ou dans son complémentaire, $\langle 1^n, S\rangle \in \hat{A}$ si et seulement si

$$Q_1 y_1 \in \{0,1\}^{p(n)} \dots Q_k y_k \in \{0,1\}^{p(n)} \langle 1^n, S\rangle \in \hat{B}(y_1, \dots, y_n).$$

Ceci montre que $\hat{A} \in \Sigma^k_{\mathbb{Z}_2}$. □

4.2 Exploration d'un arbre de décision linéaire

Soit t un polynôme fixé. On rappelle que $\mathcal{L}_t$ est le langage $\bigcup_{n\in\mathbb{N}} \mathcal{L}^n_{t(n)}$, où $\mathcal{L}^n_{t(n)}$ est l'union de tous les hyperplans de $\mathbb{R}^n$ à coefficients majorés par $t(n)$ en valeur absolue. À ce langage on associe le langage $\tilde{\mathcal{L}}_t$. Une instance de $\tilde{\mathcal{L}}_t$ se compose de trois entiers n, L, d tous trois écrits en unaire, et d'un système éventuellement vide $\mathcal{S}$ d'inégalités de la forme $l(x) \geqslant 0$ ou $l(x) < 0$. Les coefficients de ces inégalités sont écrits en binaire, et la variable x vit dans $\mathbb{R}^n$. Le système définit un polyèdre $P_{\mathcal{S}} \subset \mathbb{R}^n$. Une instance de $\tilde{\mathcal{L}}_t$ est positive s'il existe un arbre de décision linéaire T de profondeur au plus d à coefficients de tailles (en bits) au plus L tel que T décide $\mathcal{L}_t$ sur $P_{\mathcal{S}}$ (c'est-à-dire $\mathcal{L}_t \cap P_{\mathcal{S}} = E \cap P_{\mathcal{S}}$, où E est le sous-ensemble de $\mathbb{R}^n$ décidé par T).

On a besoin d'un algorithme pour résoudre $\tilde{\mathcal{L}}_t$ et pour une instance positive de ce problème, on veut en plus calculer l'étiquette l_r de la racine d'un arbre T correspondant (cet arbre n'est peut-être pas unique mais n'importe quelle solution convient). Ainsi l_r est simplement une valeur booléenne si T est réduit à une feuille, et un hyperplan affine de la forme $l(x) \geqslant 0$ sinon.

Lemme 4.2 *On a $\tilde{\mathcal{L}}_t \in$ PSPACE. De plus, pour une instance positive, l_r peut être construit en espace polynomial.*

Démonstration. On doit d'abord déterminer si T peut être de profondeur nulle, c'est-à-dire réduit à une feuille. Dans ce cas, T reconnaît $\mathbb{R}^n$ ou bien $\emptyset$ selon l'étiquette de cette feuille. L'étiquette 1 n'est *pas* acceptable si et seulement si

$$\exists x \in \mathbb{R}^n \ x \in P_{\mathcal{S}} \setminus \mathcal{L}_t \,. \tag{4.1}$$

Comme $\mathcal{L}_t \in \mathrm{NP}^0_{\mathbb{R}_{ovs}}$, le problème de décider si, étant donnés x et $\mathcal{S}$, ils vérifient $x \in P_{\mathcal{S}} \setminus \mathcal{L}_t$ est co-$\mathrm{NP}^0_{\mathbb{R}_{ovs}}$. Décider (4.1) est donc un problème Σ^2, donc PSPACE. L'étiquette 0 n'est *pas* acceptable si et seulement si $\exists x \in \mathbb{R}^n \ x \in P_{\mathcal{S}} \cap \mathcal{L}_t$, ce qui est également un problème PSPACE. S'il existe une solution de profondeur 0, on accepte l'instance de $\tilde{\mathcal{L}}_t$ et on donne en sortie l'étiquette correspondante. Sinon, pour $d > 0$ on cherche les solutions de profondeurs entre 1 et d (pour $d = 0$ on rejette l'instance). Pour cela nous énumérons (dans l'ordre lexicographique) toutes les inégalités linéaires possibles de la forme $l(x) \geqslant 0$ où les coefficients de l sont de tailles (en bits) au plus L. Pour chacune de ces inégalités on procède de la manière suivante.

1. Décider de manière récursive si $(n, L, d-1, \mathcal{S} \cup \{l(x) \geqslant 0\})$ est une instance positive de $\tilde{\mathcal{L}}_t$.
2. Décider de manière récursive si $(n, L, d-1, \mathcal{S} \cup \{l(x) < 0\})$ est une instance positive de $\tilde{\mathcal{L}}_t$.

3. En cas de réponse positive aux deux questions, sortir de la boucle, accepter et répondre $l_r = l$.

L'instance est rejetée si elle n'est pas acceptée au cours de cette procédure d'énumération. En plus de l'espace nécessaire pour résoudre le cas de la profondeur nulle, on a juste à maintenir une pile pour garder la trace des appels récursifs. Cet algorithme fonctionne donc en espace polynomial, ce qui montre $\tilde{\mathcal{L}}_t \in \mathrm{PSPACE}$. De plus, dans le cas d'une instance positive, cet algorithme fournit l_r comme cela était demandé. □

Théorème 4.1 $\mathrm{PAR}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{PSPACE})$.

Démonstration. Soit $A \in \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$, et t un polynôme majorant le temps parallèle de A. Par le théorème 3.1, l'ensemble $A \cap \mathbb{R}^n$ peut être résolu par un arbre dont la hauteur et la taille des coefficients sont majorées par an^b, où a et b sont des constantes. L'idée est d'utiliser le lemme 4.2 pour descendre dans l'arbre. De plus, pour une instance positive, on peut construire l_r en temps polynomial avec un conseil de taille polynomiale (pour être plus précis, on peut par exemple dire que chaque bit de l_r est dans PSPACE). On pose donc $L = d = an^b$ et $\mathcal{S} = \emptyset$. Par hypothèse $(n, L, d, \mathcal{S})$ est une instance positive de $\tilde{\mathcal{L}}_t$ et donc l_r peut être calculé en temps polynomial (avec un conseil de taille polynomiale). Si l_r est une valeur booléenne alors on s'arrête et on renvoie cette valeur. Sinon l_r est une fonction affine, et on peut déterminer en temps polynomial si l'entrée $x \in \mathbb{R}^n$ de $\mathcal{L}_t$ satisfait $l_r(x) \geqslant 0$. Si c'est le cas, on pose $\mathcal{S}' = \mathcal{S} \cup \{l_r \geqslant 0\}$. Sinon on pose $\mathcal{S}' = \mathcal{S} \cup \{l_r < 0\}$. Dans les deux cas, on pose $d' = d - 1$, et on lance l'algorithme décidant $\tilde{\mathcal{L}}_t$ sur l'entrée $(n, L, d', \mathcal{S}')$. Cette procédure se poursuit jusqu'à ce qu'une feuille soit atteinte, ce qui demande au plus an^b étapes. Ceci montre que $A \in \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{PSPACE})$. □

Corollaire 4.1 $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$ *si et seulement si* $\mathrm{P} = \mathrm{PSPACE}$.

Démonstration. De droite à gauche, c'est une conséquence directe du théorème 4.1. Dans l'autre sens, c'est une conséquence du théorème 2.2 sur les parties booléennes. □

Corollaire 4.2 $\mathrm{P}_{\mathbb{R}_{ovs}} = \mathrm{PAR}_{\mathbb{R}_{ovs}}$ *si et seulement si* $\mathrm{P/poly} = \mathrm{PSPACE/poly}$.

Démonstration. De gauche à droite, c'est une conséquence des parties booléennes (théorème 2.2). Considérons maintenant A un problème de $\mathrm{PAR}_{\mathbb{R}_{ovs}}(\alpha_1, \ldots, \alpha_k)$. Ce problème est la restriction d'un autre problème $\mathrm{PAR}^0_{\mathbb{R}_{ovs}}$ de plus grande dimension. Autrement dit, il existe $B \in \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$ tel que pour tout $x \in \mathbb{R}^n$, $x \in A$ si et seulement si $(x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_k) \in B$. Par le théorème 4.1, $\mathrm{P/poly} = \mathrm{PSPACE/poly}$ implique $B \in \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{P/poly})$. Bien sûr $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{P/poly}) = \mathrm{P}^0_{\mathbb{R}_{ovs}}/\mathrm{poly}$. Or $\mathrm{P}^0_{\mathbb{R}_{ovs}}/\mathrm{poly} \subset \mathrm{P}_{\mathbb{R}_{ovs}}$ par la proposition 2.2, ce qui montre $A \in \mathrm{P}_{\mathbb{R}_{ovs}}$ puisque c'est la restriction d'un problème $\mathrm{P}_{\mathbb{R}_{ovs}}$. □

4.3 Simulation de la méthode du cutting

La simulation de la méthode de Meiser permet d'établir le théorème de localisation 4.2. Rappelons que la classe #P est l'ensemble des fonctions de comptage associées aux problèmes NP – consulter par exemple le catalogue de classes de complexité de Johnson

[Joh90]. Plus précisement une fonction $f : \{0,1\}^\infty \to \mathbb{N}$ est dans #P si il existe $A \in \mathrm{P}$ et p un polynôme tels que, pour tout x

$$f(x) = |\{y \in \{0,1\}^{p(|x|)},\ \langle x, y\rangle \in A\}.$$

Nous avons besoin d'approximer les fonctions de comptage. Pour cela nous utilisons le résultat suivant dû à Stockmeyer (consulter aussi [MR95]).

Lemme 4.3 *[Sto85] Soit $f \in \#\mathrm{P}$. Soit $\alpha, d > 0$. Alors il existe $g \in \mathrm{PF}(\Sigma^2)$ telle que pour tout x de longueur n, $f(x)/(1+\alpha n^{-d}) \leqslant g(x) \leqslant f(x)(1+\alpha n^{-d})$.*

Voici comment on calcule un ε-cutting.

Lemme 4.4 *Soit t un polynôme et $0 < \varepsilon < 1$. Il existe une fonction dans $\mathrm{PF}(\Sigma^4)$ qui, sur l'entrée $\langle 1^n, A\rangle$ où A est un algorithme en temps polynomial (décidant un sous-ensemble de $\mathcal{H}_t$), calcule un ε-cutting de $H_n = \{\langle a_1, \ldots, a_{n+1}\rangle \in A \cap \mathcal{H}_t\}$.*

Démonstration. Par le théorème 3.3, on sait qu'il existe un $\varepsilon/8$-cutting de H_n de taille $q(n)$, où q est un polynôme. On va calculer un ε-cutting de taille $q(n)$. On devine un ensemble R_n de $q(n)$ hyperplans de H_n. Reste à vérifier que c'est un ε-cutting. Pour que ce soit un ε-cutting il suffit qu'aucune cellule de $\Delta\mathcal{A}(R_n)$ ne soit intersectée par plus de $\varepsilon|H_n|$ hyperplans de H_n. Soit c une cellule de $\Delta\mathcal{A}(R_n)$. Savoir s'il existe un $h \in H_n$ tel que $h \cap c \neq \emptyset$ est un problème NP. La fonction de comptage qui à $\langle c, 1^n\rangle$ associe $|\{h \in H_n, h \cap c \neq \emptyset\}|$ est donc dans #P. Voici donc comment on procède : on devine R_n, puis on vérifie si aucune cellule c de $\Delta\mathcal{A}(R_n)$ n'est intersectée par trop d'hyperplans. On commence par calculer une valeur approchée b de $|H_n|$ telle que $b/2 \leqslant |H_n| \leqslant 2b$ par le lemme 4.3 ($\alpha = d = 1$). Puis, pour une cellule c devinée, on calcule une valeur approchée du nombre d'hyperplans de H_n intersectant c ($\alpha = d = 1$ dans le lemme 4.3). Si cette valeur est inférieure à $\varepsilon/2.b/2$, alors il y a certainement moins de $\varepsilon|H_n|$ hyperplans qui intersectent cette cellule. Si ceci est vrai pour toute cellule c, on a un ε-cutting. De plus, on est sûr que cette méthode fonctionne pour un R_n puisque $\varepsilon/2.b/2 \leqslant \varepsilon/8|H_n|$ et l'on cherche un cutting de cardinal suffisamment grand pour savoir qu'il existe un $\varepsilon/8$-cutting. La méthode ci-dessus pour deviner un cutting et vérifier cette hypothèse est donc dans $\mathrm{NP}(\text{co-NP}(\Delta_3))$, c'est-à-dire dans Σ^4. Par la méthode de recherche préfixe (proposition 1.2 page 11), il existe un algorithme $\mathrm{PF}(\Sigma^4)$ calculant un cutting. □

Théorème 4.2 *Étant donné un polynôme t, le problème de localisation pour H_t est dans $\mathrm{PF}^0_{\mathbb{R}_{ovs}}(\Sigma^4)$. Ceci signifie qu'un système localisant le point d'entrée peut être calculé en temps polynomial par une machine de Turing sur $\mathbb{R}_{ovs}$ avec un oracle booléen Σ^4.*

Démonstration. On va maintenant décrire l'algorithme de localisation de x dans $\mathcal{A}(\mathcal{H}^n_{t(n)})$. Notons $H_1 = \mathcal{H}^n_{t(n)}$. À la première étape, on calcule un 1/2-cutting R_1 de H_1 par le lemme 4.4, où A accepte tous les hyperplans de H_1. Puis on détermine dans quelle cellule c_1 de $\Delta\mathcal{A}(R_1)$ se trouve x. On a donc maintenant un système de $n^{O(1)}$ inégalités à coefficients de tailles $n^{O(1)}$ décrivant c_1. On sait que la moitié au plus des hyperplans de H_1 intersecte c_1. Notons cet ensemble d'hyperplans H_2 (on ne cherche pas à le déterminer). La seconde étape consiste à calculer un 1/2-cutting de H_2. Pour cela, on utilise encore le lemme 4.4, mais l'algorithme A accepte cette fois uniquement

les hyperplans de H_2. Vérifier qu'un hyperplan se trouve dans H_2 est un problème de programmation linéaire qui se traite donc en temps polynomial. En $O(\log(|H_1|)) = n^{O(1)}$ étapes, on est ramené au problème de localiser un point dans $\Delta\mathcal{A}(R_k)$ avec $|R_k| = n^{O(1)}$, ce qui se fait en temps polynomial. □

Corollaire 4.3 $\mathrm{NP}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\Sigma^4)$.

Démonstration. Soit $L \in \mathrm{NP}^0_{\mathbb{R}_{ovs}}$. Par l'équivalence des non-déterminismes réel et booléen (théorème 2.1), il existe $A \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$ et un polynôme r tels que

$$(x_1, \ldots, x_n) \in L \quad \text{ssi} \quad \exists z \in \{0,1\}^{r(n)} \langle x, z\rangle \in A.$$

Comme il est expliqué au chapitre 2, $L \cap \mathbb{R}^n$ est une union de faces de $\mathcal{A}(\mathcal{H}^n_{t(n)})$. L'algorithme $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\Sigma^4)$ décidant L marche en deux étapes. L'entrée $x = (x_1, \ldots, x_n)$ est d'abord localisée dans $\mathcal{A}(\mathcal{H}^n_{t(n)})$. Par le théorème 4.2, ceci peut être fait en temps $\mathrm{PF}^0_{\mathbb{R}_{ovs}}(\Sigma^4)$. La sortie est un système $\mathcal{S}$ de $n^{O(1)}$ équations et inéquations de la forme $h_i(x) < 0$ or $h_i(x) = 0$. Les h_i ont des coefficients de tailles polynomiales et l'ensemble $P_{\mathcal{S}}$ des points de $\mathbb{R}^n$ satisfaisant $\mathcal{S}$ est inclus dans une face de $\mathcal{A}(\mathcal{H}^n_{t(n)})$. Ensuite il reste à vérifier si $P_{\mathcal{S}}$ est inclus dans L ou dans son complément. Ceci peut être fait par un algorithme NP standard par la méthode de calcul formel (lemme 4.1). □

Il existe une autre méthode pour terminer la démonstration précédente. Une fois que l'on a déterminé un polyèdre $P_{\mathcal{S}}$ auquel le point d'entrée x appartient, on peut calculer, par le non-déterminisme, un point $\tilde{x}$ à petites coordonnées rationnelles se trouvant dans $P_{\mathcal{S}}$ (proposition 2.1). Il reste alors à déterminer si $\tilde{x}$ est accepté, par un algorithme classique, en devinant un certificat booléen z et en lançant l'algorithme T sur $\langle q, z\rangle$. Cette nouvelle méthode montre la nature élémentaire du résultat précédent, qui n'utilise pas le fait que la programmation linéaire est dans P. Cependant, la première démonstration montre bien où le non-déterminisme est utilisé, et où on sait s'en passer.

Théorème 4.3 $\mathrm{P} = \mathrm{NP} \Leftrightarrow \mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{NP}^0_{\mathbb{R}_{ovs}}$.

Démonstration. L'implication de droite à gauche résulte des parties booléennes (théorème 2.2). La réciproque est conséquence immédiate du corollaire 4.3 puisque $\mathrm{P} = \mathrm{NP}$ implique $\Sigma^4 = \mathrm{P}$. □

Corollaire 4.4 $\mathrm{P/poly} = \mathrm{NP/poly} \Leftrightarrow \mathrm{P}_{\mathbb{R}_{ovs}} = \mathrm{NP}_{\mathbb{R}_{ovs}}$.

De droite à gauche, le résultat vient des parties booléennes. Dans l'autre sens, soit $A \in \mathrm{NP}_{\mathbb{R}_{ovs}}$. C'est la restriction d'un problème $B \in \mathrm{NP}^0_{\mathbb{R}_{ovs}}$. L'hypothèse $\mathrm{P/poly} = \mathrm{NP/poly}$ implique $\mathrm{P/poly} = \mathrm{PH}/\mathit{poly}$. L'algorithme de localisation est donc dans $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{P/poly})$. On en déduit $B \in \mathrm{P}^0_{\mathbb{R}_{ovs}}/\mathit{poly} \subset \mathrm{P}_{\mathbb{R}_{ovs}}$ en codant la fonction conseil dans un nouveau paramètre (proposition 2.2). Ceci prouve $A \in \mathrm{P}_{\mathbb{R}_{ovs}}$. □

4.4 Simulation de l'arbre de Meyer auf der Heide

On améliore le théorème de localisation en simulant la méthode de Meyer auf der Heide.

Théorème 4.4 *Pour tout polynôme t, le problème de localisation dans $\mathcal{H}_t$ est dans* $\mathrm{PF}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$.

Démonstration. Les quatre phases de l'algorithme sont les suivantes.

(a) Trouver le petit cube c dans lequel le point d'entrée x se trouve. Ceci se fait avec une dichotomie successive selon chacune des coordonnées. Cette partie de l'algorithme est donc $\mathrm{P}^0_{\mathbb{R}_{ovs}}$.

(b) Une fois que ce petit cube c est trouvé, il faut décider si l'ensemble H des hyperplans de $\mathcal{H}^n_{t(n)}$ intersectant c est non vide. Si c'est le cas, il faut en plus calculer un point y de $\bigcap_{h\in H} h$. C'est pour cette étape que l'oracle NP est nécessaire. Les détails algorithmiques sont exposés ci-dessous.

(c) Enfin, quand on a calculé un tel point y, s'il existe, il faut déterminer dans quelle pyramide de sommet y et de base une face f du grand cube se trouve x. Ceci est à nouveau un algorithme $\mathrm{P}^0_{\mathbb{R}_{ovs}}$: il y a $2n$ pyramides à tester en dimension n, chacune étant délimitée par $2(n-1)$ côtés.

(d) Il reste à localiser par récurrence $x' = (yx) \cap f$ dans f. Ceci s'obtient en remplaçant chaque test h d'un algorithme localisant x' dans f par le test $\mathrm{Aff}(y,h)$ (appliqué à x). L'étude menée sur la taille des coefficients de l'arbre prouve que les calculs sont dans $\mathrm{P}^0_{\mathbb{R}_{ovs}}$: les coefficients de chaque test s'obtiennent avec un nombre polynomial d'opérations sur $(\mathbb{Z}, +, -, \times)$, chaque entier restant de taille polynomiale au cours du calcul.

Connaissant c, comment déterminer un point de $\bigcap_{h\in H} h$ à l'aide d'un oracle NP ? Tout d'abord on demande à l'aide d'un oracle NP si $\dim\left(\bigcap_{h\in H} h\right) = 0$. Il suffit pour cela de savoir s'il existe n hyperplans à coefficients entiers majorés par $2^{t(n)}$ en valeur absolue, intersectant le petit cube c, et dont l'intersection est de dimension nulle. Étant donnés les coefficients de ces n hyperplans, tout le reste est de l'algèbre linéaire et se calcule en temps polynomial. Si la réponse est négative, alors on se demande si $\dim\left(\bigcap_{h\in H} h\right) = 1$, et ainsi de suite. En n étapes au plus, on sait si $H = \emptyset$. Et si ce n'est pas le cas, on connaît $\dim\left(\bigcap_{h\in H} h\right)$. On peut alors déterminer par recherche préfixe (proposition 1.2) $n-k$ hyperplans de H dont l'intersection est de dimension k. Il est alors aisé de calculer un point y de cette intersection.

L'algorithme exposé ci-dessus simule l'arbre reconnaissant le langage $\mathcal{L}_t$. En utilisant la méthode de perturbation décrite dans ce chapitre, on obtient un algorithme produisant un système localisant un point dans $\mathcal{H}_t$. □

Théorème 4.5 $\mathrm{NP}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$.

Démonstration. La démonstration est la même que celle du corollaire 4.3, mais la procédure de localisation est maintenant dans $\mathrm{PF}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ par le théorème 4.4. □

Ce théorème est plus puissant que les précédents, et permet de retrouver tous les résultats de ce chapitre. D'autres inclusions peuvent être obtenues par cette méthode. L'algorithme de localisation de point décrit ci-dessus peut être utilisé pour tout langage réel L dans une classe $C^0_{\mathbb{R}_{ovs}} \subset \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$. Ensuite il reste à vérifier si le polyèdre résultant $P_{\mathcal{S}}$ est inclus dans L ou son complément. Si C est "raisonnable", ce calcul est dans

$C_{\mathbb{Z}_2}$ par la méthode de calcul formel (lemme 4.1). Par exemple, ceci donne $\mathrm{PAR}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{PSPACE})$. On a aussi $(\Sigma^k_{\mathbb{R}_{ovs}})^0 \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\Sigma^k)$ pour $k \in \mathbb{N}$ et $(\Pi^k_{\mathbb{R}_{ovs}})^0 \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\Pi^k)$. On peut même établir pour tout $k \geqslant 0$

$$\begin{cases} \mathrm{PH} = \Sigma^k \Leftrightarrow \mathrm{PH}^0_{\mathbb{R}_{ovs}} = (\Sigma^k_{\mathbb{R}_{ovs}})^0 \\ \mathrm{PH/poly} = \Sigma^k\mathrm{/poly} \Leftrightarrow \mathrm{PH}_{\mathbb{R}_{ovs}} = \Sigma^k_{\mathbb{R}_{ovs}}. \end{cases}$$

On a aussi $\mathrm{PP}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{PP})$. Pour BPP, on obtient seulement $\mathrm{BPP}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP} \oplus \mathrm{BPP})$ où $\oplus$ est l'union disjointe.

On va maintenant établir un résultat de complétude. Pour un langage réel $L \subset \mathbb{R}^\infty$, définissons la *partie entière* de L comme

$$\mathrm{IP}(L) = \bigcup_{n\in\mathbb{N}} \{\langle p_1, \ldots, p_n\rangle,\ (p_1, \ldots, p_n) \in L \text{ et } p_1, \ldots, p_n \in \mathbb{Z}\}.$$

Pour une classe de complexité réelle C, on pose $\mathrm{IP}(C) = \{\mathrm{IP}(L),\ L \in C\}$.

Lemme 4.5 *Soit $A \subset \mathbb{R}^\infty$ tel que $\mathrm{IP}(A)$ soit Turing NP-difficile. Alors A est Turing $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$-difficile.*

Démonstration Par le théorème 4.5, $\mathrm{NP}^0_{\mathbb{R}_{ovs}} \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$. Comme $\mathrm{IP}(A)$ est NP-difficile, $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP}) \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{IP}(A))$, et bien sûr $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{IP}(A)) \subset \mathrm{P}^0_{\mathbb{R}_{ovs}}(A)$ puisqu'on peut calculer un entier à partir de sa représentation en bits en temps polynomial.

□

Le résultat suivant est une conséquence directe du lemme 4.5.

Théorème 4.6 *Les langages $\mathrm{KP}_\mathbb{R}$ et $\mathrm{TSP}_\mathbb{R}$ sont $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$-complets pour la réduction Turing en temps polynomial.*

4.5 Vers les modèles multiplicatifs

Nous citons ici deux théorèmes de transfert sur les structures $\mathbb{R}$ et $\mathbb{C}$ (avec produit). Le premier est obtenu en appliquant la méthode du parcours de l'arbre exposée section 4.2 à une structure M dont les formules Σ^2 sans paramètre peuvent être décidées dans PSPACE, ce qui inclut en particulier les structures $\mathbb{R}$ et $\mathbb{C}$. Ceci dit, il n'existe pas de résultat sur l'existence d'arbre de calcul (arbre de décision dont la complexité du calcul des tests est prise en compte) de profondeur polynomiale pour décider un problème NP-complet sur $\mathbb{R}$ où $\mathbb{C}$. Le théorème obtenu est donc celui-ci.

Théorème 4.7 *[Koi00a] S'il existe une famille d'arbres de calcul sur $\mathbb{R}$ de profondeurs polynomiales (avec un nombre fini de paramètres) décidant $4\mathrm{FEAS}_\mathbb{R}$, alors on a le théorème de transfert $\mathrm{P} = \mathrm{PSPACE} \Rightarrow \mathrm{P}_\mathbb{R} = \mathrm{NP}_\mathbb{R}$. Sinon $\mathrm{P}_\mathbb{R} \neq \mathrm{NP}_\mathbb{R}$ bien sûr. De même, s'il existe une famille d'arbres de calcul de profondeurs polynomiales sur $\mathbb{C}$ (avec un nombre fini de paramètres) décidant $\mathrm{HN}_\mathbb{C}$, alors $\mathrm{P} = \mathrm{PSPACE} \Rightarrow \mathrm{P}_\mathbb{C} = \mathrm{NP}_\mathbb{C}$. Sinon $\mathrm{P}_\mathbb{C} \neq \mathrm{NP}_\mathbb{C}$.*

Le deuxième résultat concerne les structures qui admettent une "énumération efficace des conditions de signe", ce qui est le cas de $\mathbb{R}$ et $\mathbb{C}$ (et en fait de tout corps, et de tout corps ordonné).

Théorème 4.8 *[Koi00b] Si* P = PSPACE, *alors* $\mathrm{B}\Sigma^2_{\mathbb{R}} = \mathrm{B}\Pi^2_{\mathbb{R}}$ *et* $\mathrm{B}\Sigma^2_{\mathbb{C}} = \mathrm{B}\Pi^2_{\mathbb{C}}$.

5. Transferts sur les réels avec addition et égalité

La situation sur la structure sans ordre $\mathbb{R}_{vs}$ (les réels avec addition, soustraction et égalité) est assez différente de celle sur $\mathbb{R}_{ovs}$ puisqu'il est possible de prouver la séparation inconditionnelle $\mathrm{P}_{\mathbb{R}_{vs}} \neq \mathrm{NP}_{\mathbb{R}_{vs}}$ – voir l'article de Meer [Mee92] à ce sujet. Il serait intéressant d'obtenir d'autres résultats de séparation sur cette structure. Malheureusement, pour diverses questions comme l'effondrement de la hiérarchie polynomiale $\mathrm{PH}_{\mathbb{R}_{vs}}$ ou la séparation de $\mathrm{PH}_{\mathbb{R}_{vs}}$ et de $\mathrm{PAR}_{\mathbb{R}_{vs}}$, cela s'avère impossible avec les techniques actuelles : les théorèmes de transfert de la section 5.3 montrent que ces questions sont aussi difficiles que certains problèmes ouverts de complexité classique. Les résultats de ce chapitre proviennent de [FK00]. Pour des résultats de même nature sur les groupes infinis d'exposant 2, consulter l'article de Bourgade [Bou01].

5.1 La méthode du chemin générique

Le résultat $\mathrm{P}_{\mathbb{R}_{vs}} \neq \mathrm{NP}_{\mathbb{R}_{vs}}$ a été prouvé par Meer [Mee92]. Dans cette section, on utilise des arguments similaires pour séparer les niveaux les plus bas de la hiérarchie polynomiale sur $\mathbb{R}_{vs}$. Séparer les niveaux supérieurs de la hiérarchie s'avère extrêmement difficile à cause d'un théorème de transfert établi en section 5.3. Ces résultats sont basés sur des observations élémentaires concernant la structure des sous-ensembles de $\mathbb{R}^n$ définissable dans $\mathbb{R}_{vs}$ (voir lemme 5.2 et lemme 5.3 en particulier).

Dans la suite, ces sous-ensembles sont simplement appelés "ensembles définissables". Rappelons que les ensembles définissables le sont sans quantificateurs puisque $\mathbb{R}_{vs}$ admet l'élimination des quantificateurs. En conséquence, un ensemble définissable n'est rien d'autre qu'une combinaison booléenne d'hyperplans. Dans le reste de ce chapitre, on travaille avec des machines sans paramètres sauf indication contraire. On peut facilement généraliser ces théorèmes au cas des machines avec paramètres (remarque 5.2 page 48).

Rappelons tout d'abord la méthode du chemin générique. Soit M une machine sur

$\mathbb{R}_{vs}$ s'arrêtant sur toute entrée, et L un langage décidé par M. Étant donné $n \in \mathbb{N}\backslash\{0\}$, on pose $L_n = L \cap \mathbb{R}^n$. Le chemin générique de la machine M pour les entrées de taille n est le chemin obtenu en répondant *non* à tous les tests de la forme $''h(x) = 0\ ?''$ (sauf si $h = 0$, auquel cas on répond *oui*). Cette définition est effective au sens où le chemin générique peut être suivi par une machine M sur une entrée formelle $(X_1, \ldots, X_n)$. Si M est une machine sans paramètres, ce calcul peut être mené sur une machine de Turing classique. De plus, si M travaille en temps $t(n)$, la méthode du chemin générique fonctionne en temps $O(nt(n)^2)$ et les tests effectués le long de ce chemin sont calculés effectivement. Notons $\{h_1, \ldots, h_r\}$ ces tests. Bien sûr $r \leqslant t(n)$. De plus, ces hyperplans ont la propriété suivante : si les entrées suivant le chemin générique sont rejetées, $L_n \subset h_1 \cup \ldots \cup h_r$; sinon ces entrées sont acceptées et $L_n^c \subset h_1 \cup \ldots \cup h_r$. Notons que la méthode du chemin générique peut être appliquée à un sous-espace affine $X \subset \mathbb{R}^n$ au lieu de $\mathbb{R}^n$, auquel cas on répond *oui* à un test $''h(x) = 0\ ?''$ si et seulement si $X \subset h$. Dans ce cas particulier où X est un espace affine, un ensemble définissable A de $\mathbb{R}^n$ est dense dans X si et seulement si $\dim A \cap X = \dim X$. Ces observations sont résumées dans le lemme suivant. Dans ce lemme, Sys^n représente l'ensemble des systèmes d'équations affines en n variables avec des coefficients dans $\mathbb{Z}$. Pour $S \in \mathrm{Sys}^n$, P_S représente le sous-espace affine de $\mathbb{R}^n$ défini par S.

Lemme 5.1 *Soit A un langage de $\mathbb{R}^\infty$ et $A^n = A \cap \mathbb{R}^n$. Notons L^n l'ensemble des systèmes $S \in \mathrm{Sys}^n$ tels que A^n est dense dans P_S, et L le langage $\bigcup_{n \geqslant 1} L^n$. Supposons $A \in C^0_{\mathbb{R}_{vs}}$, avec $C = \mathrm{PAR}$ ou $C = \Sigma^k$ pour $k \in \mathbb{N}$. Alors $L \in C_{\mathbb{Z}_2}$.*

Démonstration. Notons que A^n est définissable pour tout $A \in C^0_{\mathbb{R}_{vs}}$ (c'est en fait le cas pour tout langage récursif de ${\mathbb{R}_{vs}}^\infty$). Cet ensemble est dense dans P_S si et seulement si un point générique de P_S appartient à A. On peut donc appliquer la méthode du point générique décrite ci-dessus. Plus précisément, considérons le cas $A \in \mathrm{P}^0_{\mathbb{R}_{vs}}$. Étant donné un hyperplan test h, on peut décider en temps polynomial si $P_S \subset h$ par des techniques d'algèbre linéaire. La même remarque s'applique dans le cas $C = \mathrm{PAR}$ puisque les hyperplans tests ont toujours des coefficients de taille polynomiale. On conclut que L est dans P si $A \in \mathrm{P}^0_{\mathbb{R}_{vs}}$, et L est dans $\mathrm{PAR}_{\mathbb{Z}_2} = \mathrm{PSPACE}$ si $A \in \mathrm{PAR}^0_{\mathbb{R}_{vs}}$. Si $A \in (\Sigma^k_{\mathbb{R}_{vs}})^0$ pour un $k \geqslant 1$ on utilise l'équivalence entre quantifications réelles et booléennes pour $\mathbb{R}_{vs}$ (théorème 2.1) : il existe un polynôme p et $B \in \mathrm{P}^0_{\mathbb{R}_{vs}}$ tels que pour tout $x \in \mathbb{R}^n$, $x \in A$ si et seulement si

$$Q_1 y_1 \in \{0,1\}^{p(n)} \cdots Q_k y_k \in \{0,1\}^{p(n)} \langle x, y_1, \ldots, y_k \rangle \in B$$

(les quantificateurs Q_i alternent, avec $Q_1 = \exists$). L'ensemble A^n est dense dans P_S si et seulement si on a

$$Q_1 y_1 \in \{0,1\}^{p(n)} \cdots Q_k y_k \in \{0,1\}^{p(n)} F_n(y_1, \ldots, y_k) \tag{5.1}$$

où $F_n(y_1, \ldots, y_k)$ signifie : "$\{x \in \mathbb{R}^n;\ \langle x, y_1, \ldots, y_k \rangle \in B\}$ est dense dans P_S". Comme $B \in \mathrm{P}^0_{\mathbb{R}_{vs}}$, on sait que $F_n(y_1, \ldots, y_k)$ peut être décidé en temps polynomial par la méthode du point générique. Donc 5.1 montre que $L \in \Sigma^k$. □

Remarquons que dans le cas $C = \Sigma^k$ il est vraiment nécessaire de passer aux quantifications booléennes avant d'appliquer la méthode du point générique (penser par exemple à l'ensemble des points $x \in \mathbb{R}$ défini par la formule $\exists y\ x = y$).

5.2 Séparation des premiers niveaux

Le problème "Twenty Questions" est un candidat potentiel pour la séparation $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$ – voir [SS96]. Ce problème est défini de la manière suivante :

$$\mathrm{TQ} = \bigcup_{n\in\mathbb{N}} \{(x_1, \ldots, x_n) \in \mathbb{R}^n,\ x_1 \in \{0, \ldots, 2^n - 1\}\}.$$

Ici on montre que Twenty Questions peut être utilisé pour séparer $\Sigma^1_{\mathbb{R}_{vs}}$ de $\Pi^1_{\mathbb{R}_{vs}}$.

Proposition 5.1 $\mathrm{TQ} \in \Sigma^1_{\mathbb{R}_{vs}} \setminus \Pi^1_{\mathbb{R}_{vs}}$.

Démonstration On rappelle [SS96] que $\mathrm{TQ} \in \Sigma^1_{\mathbb{R}_{vs}}$ puisque la restriction de ce problème à $\mathbb{R}^n$ est $\mathrm{TQ}_n = \{(x_1, \ldots, x_n),\ \exists u_1, \ldots, u_n \in \{0,1\}\ x_1 = \sum_i 2^{i-1} u_i\}$. Supposons par l'absurde que $\mathrm{TQ} \in \Pi^1_{\mathbb{R}_{vs}}$: TQ_n est donc l'intersection d'un nombre fini d'ensembles A_i de la forme $A_i = L_i \cap \mathbb{R}^n$ où $L_i \in \mathrm{P}^0_{\mathbb{R}_{vs}}$. Puisque TQ_n n'est pas dense dans $\mathbb{R}^n$, il existe un ensemble A_{i_0} qui n'est pas dense dans $\mathbb{R}^n$. La méthode du chemin générique montre que A_{i_0} est inclus dans l'union de $n^{O(1)}$ hyperplans. Ceci est impossible puisque TQ_n est composé de l'union de 2^n hyperplans distincts.
□

Proposition 5.2 $(\Sigma^2_{\mathbb{R}_{vs}} \cap \Pi^2_{\mathbb{R}_{vs}}) \setminus (\Sigma^1_{\mathbb{R}_{vs}} \cup \Pi^1_{\mathbb{R}_{vs}}) \neq \emptyset$.

Démonstration. Considérons le langage suivant :

$$L = \bigcup_{n\in\mathbb{N}} \begin{array}{l} \{(x_1, \ldots, x_n),\ \exists u_1, \ldots, u_n \in \{0,1\}\ \forall v_1, \ldots, v_n \in \{0,1\} \\ x_1 = \sum_i 2^{i-1} u_i \text{ et } x_2 \neq \sum_j 2^{j-1} v_j\}. \end{array}$$

On peut échanger les deux blocs de quantificateurs, donc $L \in \Sigma^2_{\mathbb{R}_{vs}} \cap \Pi^2_{\mathbb{R}_{vs}}$. Supposons maintenant que $L \in \Sigma^1_{\mathbb{R}_{vs}}$. La restriction $L_n = L \cap \mathbb{R}^n$ est donc une union finie de A_i de la forme $L_i \cap \mathbb{R}^n$ où $L_i \in \mathrm{P}^0_{\mathbb{R}_{vs}}$. Un de ces ensembles, disons A_{i_0}, doit être dense dans le sous-espace $S = \{x \in \mathbb{R}^n, x_1 = 0\}$. La méthode du chemin générique appliquée à L_{i_0} sur ce sous-espace montre que $A^c_{i_0} \cap S$ est inclus dans l'union de $n^{O(1)}$ hyperplans. Ceci est impossible puisque $L^c \cap S$ est l'union de 2^n hyperplans. Un argument similaire appliqué à $S' = \{x \in \mathbb{R}^n, x_2 = 1/2\}$ montre que $L \notin \Pi^1_{\mathbb{R}_{vs}}$. □

5.3 Transferts dans la hiérarchie polynomiale

Dans cette section on montre que la séparation des niveaux supérieurs de $\mathrm{PH}_{\mathbb{R}_{vs}}$ est reliée à des séparations dans la hiérarchie booléenne. On commence avec deux lemmes. Le lemme 5.2 est une remarque concernant la structure des sous-ensembles définissables de $\mathbb{R}^n$. Dans le lemme 5.3, on construit une formule $\Sigma^2_{\mathbb{R}_{vs}}$ *générique* décidant un ensemble définissable A à l'aide du prédicat $\dim S \cap A = \dim S$ (la variable S représentant un espace affine de $\mathbb{R}^n$).

Lemme 5.2 *Tout ensemble définissable non vide $A \subset \mathbb{R}^n$ est de la forme*

$$A = E_k \setminus (E_{k-1} \setminus (\ldots \setminus E_0))$$

où E_i est une union finie d'espaces affines, $E_{i-1} \subsetneq E_i$, $E_i = \overline{E_i \setminus E_{i-1}}$, et $k \leqslant \dim A$.

Démonstration. Si $\dim A = 0$ le résultat est clairement vrai puisque A est une union finie de points. Supposons par récurrence que le résultat est vrai pour tous les ensembles définissables de dimension au plus $d-1$, et soit A un ensemble définissable de dimension d. L'adhérence $\overline{A}$ de A est une union finie de sous-espaces affines. Si $A = \overline{A}$ on pose $k = 0$ et $E_k = A$. Sinon, considérons l'ensemble définissable $A_1 = \overline{A} \setminus A$. Puisque $\dim A_1 \leqslant d-1$, pour un $k \leqslant d$ on peut écrire par hypothèse de récurrence $A_1 = E_{k-1} \setminus (E_{k-2} \setminus (\cdots \setminus E_0))$ avec E_i une union finie d'espaces affines, $E_{i-1} \subsetneq E_i$, $E_i = \overline{E_i \setminus E_{i-1}}$. Comme $A = \overline{A} \setminus A_1$, on peut prendre $E_k = \overline{A}$. □

Lemme 5.3 *Pour tout ensemble définissable $A \subset \mathbb{R}^n$ on a :*

$$\begin{aligned}(x_1, \ldots, x_n) \in A \Leftrightarrow \exists S_1 \forall S_2 \\ x \in S_1 \wedge \dim A \cap S_1 = \dim S_1 \\ \wedge (\dim A \cap S_1 \cap S_2 < \dim S_1 \cap S_2 \Rightarrow x \notin S_1 \cap S_2)\end{aligned}$$

où S_1 et S_2 sont des espaces affines de $\mathbb{R}^n$.

Démonstration. Le résultat est bien sûr vrai si $A = \emptyset$. Sinon, on écrit $A = E_k \setminus (E_{k-1} \setminus (\ldots \setminus E_0))$ comme dans le lemme 5.2. Soit $x \in A$ et $i_0 = \min\{i;\ i = k \bmod 2,\ x \in E_i\}$. Alors $x \notin E_{i_0-1}$: si x appartenait à E_{i_0-1}, comme $x \in A$ il existerait $i < i_0$ tel que $i = k \bmod 2$ et $x \in E_i$. Ceci serait en contradiction avec la minimalité de i_0. On montre d'abord l'implication de gauche à droite : soit S_1 un sous espace affine de E_{i_0} maximal contenant x. Comme $x \in S_1$ et $x \notin E_{i_0-1}$, on a l'inclusion stricte $S_1 \cap E_{i_0-1} \subsetneq S_1$. Donc $\dim S_1 \setminus E_{i_0-1} = \dim S_1$ et $\dim A \cap S_1 = \dim S_1$. Enfin, si $\dim A \cap S_1 \cap S_2 < \dim S_1 \cap S_2$, alors $S_1 \cap S_2 \subset E_{i_0-1}$. Ainsi $x \notin S_1 \cap S_2$. Réciproquement, supposons maintenant que x satisfait la formule pour $S_1 = S$. Comme $A \cap S$ est définissable, par le lemme 5.2 on peut écrire $A \cap S = E_k \setminus (E_{k-1} \setminus (\ldots \setminus E_0))$. Ici $E_k = \overline{A \cap S} = S$ – la seconde égalité venant de $\dim A \cap S = \dim S$. E_{k-1} est une union finie d'espaces affines. Pour tout sous-espace S_2 de cette union on a $\dim A \cap S \cap S_2 < \dim S \cap S_2$, donc $x \notin S \cap S_2$. Ceci montre que $x \notin E_{k-1}$, en conséquence de quoi $x \in A \cap S$. □

Remarque 5.1 *Dans le lemme précédent, si l'ensemble définissable A est une combinaison booléenne d'hyperplans à coefficients dans $\mathcal{D} \subset \mathbb{R}$, alors on peut quantifier seulement sur les espaces affines définis par des systèmes d'équations affines à coefficients dans $\mathcal{D}$.*

On peut maintenant énoncer et prouver notre théorème de transfert pour $\mathbb{R}_{vs}$. Remarquons qu'on a un décalage de deux niveaux dans les théorèmes 5.1 et 5.2.

Théorème 5.1 $\mathrm{P} = \mathrm{PSPACE} \Rightarrow \mathrm{PAR}^0_{\mathbb{R}_{vs}} = \left(\Sigma^2_{\mathbb{R}_{vs}}\right)^0 \cap \left(\Pi^2_{\mathbb{R}_{vs}}\right)^0$.

Démonstration. Supposons que P = PSPACE, et soit $L \in \mathrm{PAR}^0_{\mathbb{R}_{vs}}$ un langage décidé par une famille P-uniforme de circuits de profondeurs $t(n)$. Comme $\mathrm{PAR}^0_{\mathbb{R}_{vs}}$ est clos par complément, il est suffisant de montrer que $\mathrm{PAR}^0_{\mathbb{R}_{vs}} = (\Sigma^2_{\mathbb{R}_{vs}})^0$. Par le lemme 5.3, L est décidé par la formule $(\Sigma^2_{\mathbb{R}_{vs}})^0$ suivante :

$$\begin{aligned}(x_1,\ldots,x_n) \in L \Leftrightarrow\ & \exists \mathcal{S}_1 \forall \mathcal{S}_2 \\ & x \in P_{\mathcal{S}_1} \wedge \dim L^n \cap P_{\mathcal{S}_1} = \dim P_{\mathcal{S}_1} \\ & \wedge (\dim L^n \cap P_{\mathcal{S}_1 \cup \mathcal{S}_2} < \dim P_{\mathcal{S}_1 \cup \mathcal{S}_2} \Rightarrow x \notin P_{\mathcal{S}_1 \cup \mathcal{S}_2})\end{aligned}$$

où $L^n = L \cap \mathbb{R}^n$, $\mathcal{S}_1$ et $\mathcal{S}_2$ sont des systèmes composés d'au plus n équations affines à coefficients dans $\{-2^{t(n)},\ldots,2^{t(n)}\}$ (remarque 5.1), et $P_{\mathcal{S}}$ est le sous-espace de $\mathbb{R}^n$ défini par $\mathcal{S}$. Par le lemme 5.1 la condition $\dim L_n \cap P_{\mathcal{S}} = \dim P_{\mathcal{S}}$ peut être vérifiée dans PSPACE, et par conséquence dans P par hypothèse. □

Théorème 5.2 *Pour tout $k \geqslant 0$:*

$$\mathrm{PH} = \Sigma^k \Rightarrow \mathrm{PH}^0_{\mathbb{R}_{vs}} = (\Sigma^{k+2}_{\mathbb{R}_{vs}})^0.$$

Démonstration. Considérons un langage $L \in \mathrm{PH}^0_{\mathbb{R}_{vs}}$: on a $L \in (\Sigma^q_{\mathbb{R}_{vs}})^0$ pour un $q \geqslant 0$. Comme dans la démonstration du théorème précédent, on utilise la formule Σ^2 du lemme 5.3. Comme $\mathrm{PH}^0_{\mathbb{R}_{vs}} \subset \mathrm{PAR}^0_{\mathbb{R}_{vs}}$, par la remarque 5.1 on peut encore se contenter de quantifier sur des systèmes composés d'équations à coefficients dans $\{-2^{p(n)},\ldots,2^{p(n)}\}$, pour un polynôme p. Par le lemme 5.1 la condition $\dim P_S \cap L_n = \dim P_S$ peut être vérifiée dans Σ^q, et donc dans Σ^k par hypothèse. En mettant la formule obtenue sous forme prénexe, on obtient $L \in (\Sigma^{k+2}_{\mathbb{R}_{vs}})^0$. □

Le dernier théorème de transfert est basé sur une méthode un peu différente.

Théorème 5.3 $\mathrm{P} = \mathrm{NP} \Rightarrow (\Sigma^1_{\mathbb{R}_{vs}})^0 \cap (\Pi^1_{\mathbb{R}_{vs}})^0 = \mathrm{P}^0_{\mathbb{R}_{vs}}$.

Considérons un langage $L \in (\Sigma^1_{\mathbb{R}_{vs}})^0 \cap (\Pi^1_{\mathbb{R}_{vs}})^0$. Comme $L \in (\Sigma^1_{\mathbb{R}_{vs}})^0$, il existe $A \in \mathrm{P}^0_{\mathbb{R}_{vs}}$ et un polynôme p tels que

$$L = \{x \in \mathbb{R}^\infty;\ \exists u \in \{0,1\}^{p(|x|)}\ \langle x,u\rangle \in A\}.$$

Il existe aussi un problème $B \in \mathrm{P}^0_{\mathbb{R}_{vs}}$ et un polynôme q tels que

$$L = \{x \in \mathbb{R}^\infty;\ \forall v \in \{0,1\}^{q(|x|)}\ \langle x,v\rangle \in B\}$$

puisque $L \in (\Pi^1_{\mathbb{R}_{vs}})^0$. Soit $A^n_i = \{x \in \mathbb{R}^n; \langle x,i\rangle \in A\}$ et $B^n_j = \{x \in \mathbb{R}^n; \langle x,j\rangle \in B\}$. Avec ces notations, on a $L^n = \bigcup_{i\in\{0,1\}^{p(n)}} A^n_i = \bigcap_{j\in\{0,1\}^{q(n)}} B^n_j$ où $L^n = L \cap \mathbb{R}^n$. Les langages A et B sont décidables en temps polynomial ; soit t un polynôme tel que $t(n)$ borne le temps de calcul nécessaire à :

(i) décider A sur une entrée de la forme $\langle x,u\rangle$ avec $x \in \mathbb{R}^n$ et $u \in \mathbb{R}^{p(n)}$;

(ii) décider B sur une entrée de la forme $\langle x,v\rangle$ avec $x \in \mathbb{R}^n$ et $v \in \mathbb{R}^{q(n)}$.

Considérons deux cas : $\dim L^n < n$ ou $\dim L^n = n$. Dans le premier cas, l'un des B_i^n, par exemple $B_{i_0}^n$, doit vérifier $\dim B_{i_0}^n < n$ puisque L^n est une intersection (finie) de B_i^n. Comme $L^n \subset B_{i_0}^n$, l'argument du chemin générique nous montre que L^n est inclus dans l'union de $t(n)$ espaces affines h_k de dimensions $n-1$. Dans le cas $\dim L^n = n$, il doit exister $A_{i_0}^n$ tel que $\dim A_{i_0}^n = n$. Mais $(L^n)^c = \bigcap_i (A_i^n)^c \subset (A_{i_0}^n)^c$. Le complément de L^n est alors inclus dans l'union de $t(n)$ espaces affines h_k de dimensions $n-1$. De plus, la restriction de L^n à un hyperplan h_k (ou même à n'importe que hyperplan) a la même structure.

On explique maintenant comment L peut être décidé dans $\mathrm{P}^0_{\mathbb{R}_{vs}}$ sous l'hypothèse $\mathrm{P} = \mathrm{NP}$. Soit x une entrée de $\mathbb{R}^n$. A la première étape, on utilise la méthode du chemin générique pour vérifier si $\dim L^n < n$ ou $\dim L^n = n$ (ceci est un problème NP). Premier cas : $\dim L^n < n$. Alors on devine $j \in \{0,1\}^{q(n)}$ tel que $\dim B_j^n < n$. En utilisant la méthode du chemin générique, on calcule un ensemble d'au plus $t(n)$ hyperplans dont l'union contient L^n (cet algorithme est dans P). On vérifie ensuite si x appartient à l'un de ces hyperplans : si ce n'est pas le cas, on rejette ; sinon, on passe à l'étape suivante. Le cas $\dim L^n = n$ est similaire. On devine $i \in \{0,1\}^{p(n)}$ tel que $\dim A_i^n = n$. La méthode du chemin générique produit au plus $t(n)$ hyperplans dont l'union contient $(L^n)^c$. Il reste à vérifier si x appartient à l'un de ces hyperplans. Si ce n'est pas le cas on accepte, sinon on passe à l'étape suivante. Au début l'étape k, on a calculé un système d'équations (à coefficients entiers) définissant un espace affine de dimension $n-k+1$ auquel x appartient. On sait aussi qu'une restriction de L^n à cet espace a la même structure que dans la première étape : L^n ou son complément, restreint à cet espace, est inclus dans l'union d'au plus $t(n)$ espaces affines de dimensions $n-k$. En utilisant un oracle NP il est possible de déterminer ces hyperplans. Ensuite x est accepté, ou rejeté, ou alors on passe à l'étape $k+1$. Cet algorithme termine après au plus n étapes. □

Remarque 5.2 *Les trois théorèmes de transfert obtenus dans cette partie peuvent être étendus aux machines avec paramètres. Par exemple, montrons que* $\mathrm{PH} = \Sigma^k \Rightarrow \mathrm{PH}_{\mathbb{R}_{vs}} = \Sigma^{k+2}_{\mathbb{R}_{vs}}$. *Pour un problème* $L \in \mathrm{PH}_{\mathbb{R}_{vs}}$ *il existe des paramètres* $\alpha_1, \ldots, \alpha_p$ *et un problème* $L' \in \mathrm{PH}^0_{\mathbb{R}_{vs}}$ *tel que* $(x_1, \ldots, x_n) \in L$ *si et seulement si* $(x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_p) \in L'$. *Par le théorème 5.2,* $L' \in (\Sigma^{k+2}_{\mathbb{R}_{vs}})^0$ *sous l'hypothèse* $\mathrm{PH} = \Sigma^k$. *Ceci implique* $L \in \Sigma^{k+2}_{\mathbb{R}_{vs}}$.

6. Existence de problèmes creux NP-complets

6.1 Présentation du problème

Les recherches autour de l'existence de problèmes creux NP-complets ont débuté à la fin des années soixante-dix. Nous nous contenterons de rappeler quelques résultats. Pour des développements récents dans ce domaine, on pourra consulter l'article de Cai et Ogihara [CO97]. Un langage $L \subset \{0,1\}^\infty$ est dit creux si $|L \cap \{0,1\}^n| = n^{O(1)}$. Parmi les premiers résultats, citons le théorème de Karp-Lipton [KL82] : s'il existe un langage creux NP-dur pour la réduction Turing, alors la hiérarchie polynomiale s'effondre au deuxième niveau. A peu près en même temps, Mahaney [Mah82] montrait qu'il ne peut exister de langage creux NP-dur pour la réduction many-one sous l'hypothèse P $\neq$ NP.

Cucker, Koiran et Matamala ont étudié une question similaire sur les réels avec addition et égalité [CKM97]. Rappelons que $\mathbb{R}_{vs}$ désigne la structure $(\mathbb{R}, 0, 1, +, -, =)$, et $\mathbb{R}_{ovs}$ la structure $(\mathbb{R}, 0, 1, +, -, \leqslant)$. Nous dirons qu'un langage réel L est définissable sur une structure S si pour tout n, l'ensemble $L \cap \mathbb{R}^n$ est définissable sur S. Rappelons la définition d'un langage réel creux tel qu'il est défini dans [CKM97].

Définition 6.1 *Un langage $L \subset \mathbb{R}^\infty$ définissable sur $\mathbb{R}_{vs}$ ou $\mathbb{R}_{ovs}$ est creux si* $\dim(L \cap \mathbb{R}^n) = (\log n)^{O(1)}$.

Il est montré dans [CKM97] qu'il n'existe pas de langage creux définissable sur $\mathbb{R}_{vs}$ et $\mathrm{NP}_{\mathbb{R}_{vs}}$-dur pour la réduction many-one. Nous nous intéressons ici à des questions analogues. Les résultats de ce chapitre proviennent de [Fou01b]. Dans la section 6.2, nous obtenons un résultat similaire en ce qui concerne la réduction Turing. Le cas des réels avec addition et ordre est traitée dans la section 6.3, où nous exhibons un problème creux Turing-complet. En ce qui concerne la réduction many-one, nous conjecturons qu'il n'existe pas de problème creux complet. La figure 6.1 résume cette situation. Citons également l'approche de Matamala et Meer [MM99], où la notion de langage creux est remplacée par celle de langage bien structuré. Un langage L est dit bien structuré si le

	many-one	Turing
$(\mathbb{R}, 0, 1, +, -, =)$	non	non
$(\mathbb{R}, 0, 1, +, -, <)$	?	oui

FIG. 6.1 – Existence de problèmes creux NP-complets

nombre de composantes connexes de $L \cap \mathbb{R}^n$ est polynomialement borné et s'il existe un algorithme en temps polynomial décidant l'appartenance de deux points de L à la même composante connexe. Ils obtiennent entre autres un équivalent du théorème de Mahaney : il n'existe pas de langage bien structuré NP-complet (pour la réduction many-one en temps polynomial) sur $\mathbb{R}_{ovs}$ si P $\neq$ NP sur cette structure.

Enfin, pour une notion de problème creux identique à celle considérée ici, basée sur la dimension des langages, Cucker et Grigoriev [CG01] ont montré qu'il n'existe pas de problème creux NP-complet (pour la réduction many-one en temps polynomial) sur le modèle des machines faibles.

6.2 Cas des réels avec égalité

Rappelons la définition du sac-à-dos réel $\mathrm{KP}_{\mathbb{R}}$:

$$\mathrm{KP}_{\mathbb{R}} \cap \mathbb{R}^{n+1} = \{(y, x_1, \ldots, x_n),\ \exists I \subset \{1, \ldots, n\} \sum_{i \in I} x_i = y\}.$$

Il est clair que ce langage appartient à la classe $\mathrm{NP}_{\mathbb{R}_{vs}}$, et donc aussi à $\mathrm{NP}_{\mathbb{R}_{ovs}}$. Géométriquement, $\mathrm{KP}_{\mathbb{R}} \cap \mathbb{R}^{n+1}$ se compose de l'union de 2^n hyperplans. Nous établissons tout d'abord une borne sur le nombre de ces hyperplans contenant un sous-espace linéaire donné.

Lemme 6.1 *Soit A un sous-espace linéaire de dimension k de $\mathbb{R}^{n+1}$. Alors il existe au plus 2^{n-k} hyperplans de $\mathrm{KP}_{\mathbb{R}} \cap \mathbb{R}^{n+1}$ dont la direction contient A.*

Démonstration. Soit $\{v_1, \ldots, v_k\}$ une base de A. On note $\tilde{V}$ la matrice dont les lignes sont $\{v_1, \ldots, v_k\}$. A un vecteur (colonne) $U = (u_1, \ldots, u_n) \in \{0,1\}^n$ correspond l'hyperplan d'équation $\sum_i u_i x_i - y = 0$; on note $\tilde{U} = (-1, u_1, \ldots, u_n)$. Cet hyperplan contient A si $\tilde{V}\tilde{U} = 0$, ce qu'on peut aussi écrire $VU = B$. En choisissant k colonnes indépendantes dans V, cela donne une matrice inversible V' et un système $V'U' = B - V''U''$. On trouve alors une solution U' au plus pour chaque $U'' \in \{0,1\}^{n-k}$. □

Proposition 6.1 *Il n'existe pas de langage creux $\mathbb{R}_{vs}$-définissable $\mathrm{NP}_{\mathbb{R}_{vs}}$-dur pour la réduction Turing.*

Démonstration. Supposons que $\mathrm{NP}_{\mathbb{R}_{vs}} \leqslant_T S$ avec S creux et définissable sur $\mathbb{R}_{vs}$. Soit φ un algorithme de $\mathrm{P}_{\mathbb{R}_{vs}}(S)$ décidant $\mathrm{KP}_{\mathbb{R}}$, et t un polynôme bornant son temps de calcul. On définit le chemin générique γ pour $\mathbb{R}^n$: à chaque fois que le programme atteint un point de branchement, que ce soit un test d'égalité ou une question à l'oracle, le chemin suivi est celui qui est suivi par un ensemble de dimension n des points d'entrée.

Pour un chemin α, on note X_α l'ensemble des points d'entrée de $\mathbb{R}^n$ qui suivent ce chemin, et C_α son complémentaire. Pour un ensemble $A \subset \mathbb{R}^n$, soit $H(A)$ l'ensemble des hyperplans de KP_n inclus dans $\overline{A}$ (l'adhérence de A) après une éventuelle translation, et $N(A) = |H(A)|$. Posons $N_\alpha = |H(C_\alpha)|$. Enfin, on note D_α l'ensemble des points qui quittent le chemin générique au pas de calcul suivant α.

Soit α un préfixe strict du chemin générique pour $\mathbb{R}^n$, relativement à la machine φ. Soit α' le chemin générique correspondant à un pas de calcul supplémentaire. Si ce pas de calcul est composé d'une opération ou d'un test, alors D_α est contenu dans un hyperplan, donc $N(D_\alpha) \leqslant 1$. Considérons maintenant le cas où ce pas de calcul correspond à une question posée à l'oracle : cette question est donnée par une application affine des coordonnées d'entrée $x_1, \ldots, x_n$. Cette application ne dépend que de α. Soit L la partie linéaire de cette application. Soit $A = L^{-1}(S)$.

Premier cas : $\dim A \leqslant n-1$. Le chemin générique prend la branche correspondant au *non* et $D_\alpha = A + p$ pour un point $p \in \mathbb{R}^n$. Majorons maintenant $N(A)$. L'application L est dans $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ avec $m \leqslant t(n)$. On rappelle que $\dim S_n \leqslant a(\log n)^b$. Si $\dim ImL > a(\log t(n))^b + 1$ alors $\dim A \leqslant n-2$ et $N(A) = 0$. Sinon $\dim ImL \leqslant a(\log t(n))^b + 1$, et donc $\dim KerL \geqslant n - a(\log t(n))^b - 1$. Tous les hyperplans de $H(A)$ contiennent $KerL$. Par le lemme 6.1, le nombre de tels hyperplans est majoré par $2^{n-dim(KerL)}$, ce qui montre $N(A) \leqslant 2^{(\log n)^{O(1)}}$.

Second cas : $\dim A = n$, et donc $D_\alpha = A^c$. Comme S est creux, on a $\dim ImL \leqslant a(\log t(n))^b$ et pour les mêmes raisons que ci-dessus $N(A^c) \leqslant 2^{(\log n)^{O(1)}}$.

Il reste à remarquer que pour A et B définissables, $N(A \cup B) \leqslant N(A) + N(B)$. Ainsi, $N_{\alpha'} = N(C_{\alpha'}) = N(C_\alpha \cup D_\alpha) \leqslant N_\alpha + N(D_\alpha)$. Appliqué au chemin générique γ (de longueur majorée par $t(n)$), ceci donne $N_\gamma \leqslant 2^{(\log n)^{O(1)}} = o(2^n)$. Ceci montre que $X_\gamma \cap \mathrm{KP}_n \neq \emptyset$ quand n est assez grand, ce qui est absurde puisque les points de X_γ sont rejetés (car $\dim X_\gamma = n$). □

6.3 Cas des réels avec ordre

Rappelons tout d'abord ceci : un langage de $\mathrm{NP}_{\mathbb{R}_{ovs}}$ peut être décidé par un algorithme en temps polynomial sur $\mathbb{R}_{ovs}$ avec l'aide d'un oracle booléen de NP (c'est le théorème 4.5). Ceci implique que n'importe quel problème NP-complet $L \subset \{0,1\}^\infty \subset \mathbb{R}^\infty$ est $\mathrm{NP}_{\mathbb{R}_{ovs}}$-complet pour la réduction Turing. Bien sûr un tel langage vérifie $\dim(L \cap \mathbb{R}^n) = 0$, donc il existe des langages creux NP-complets pour $\leqslant_T$ sur $\mathbb{R}_{ovs}$. On peut même énoncer la proposition suivante reposant sur la méthode d'exploration d'arbre (section 4.2).

Proposition 6.2 *Étant donnée une structure S, les langages de NP_S sont décidables par des suites d'arbres de calcul de profondeurs polynomiales si et seulement si il existe un problème booléen NP_S-dur pour la réduction Turing en temps polynomial.*

Par contre, on conjecture qu'il n'existe pas de langage creux $\mathrm{NP}_{\mathbb{R}_{ovs}}$-complet pour la réduction many-one. Introduisons une variante $\mathrm{KP}'_{\mathbb{R}}$ du sac-à-dos réel. Étant donnée

une application t, on définit $\mathrm{KP}'_{\mathbb{R}}[t]$ par

$$\mathrm{KP}'_{\mathbb{R}}[t] \cap \mathbb{R}^{n+1} = \{(y, x_1, \ldots, x_n) \in \mathrm{KP}_{\mathbb{R}} \mid \forall (a_1, \ldots, a_n) \in \{-2^{t(n)}, \ldots, 2^{t(n)}\}^n \setminus \{0\} \ \sum_{i=1}^{n} a_i x_i \neq 0\}.$$

Ainsi $\mathrm{KP}'_{\mathbb{R}}[t] \cap \mathbb{R}^{n+1}$ est l'union de 2^n hyperplans avec des trous de dimension $n-1$.

Proposition 6.3 *Si un langage L tel que $\dim(L \cap \mathbb{R}^n) = O(\log n)$ est $\mathrm{NP}_{\mathbb{R}_{ovs}}$-dur pour la réduction many-one, alors il existe un polynôme t et un langage réel A tel que $\mathrm{KP}'_{\mathbb{R}}[t] \subset A \subset \mathrm{KP}_{\mathbb{R}}$ et $A \in \mathrm{P}_{\mathbb{R}_{ovs}}$.*

Démonstration. Supposons qu'il existe un tel langage L. Alors $\dim(L \cap \mathbb{R}^n) \leqslant a \log n$. Comme $\mathrm{KP}_{\mathbb{R}} \in \mathrm{NP}_{\mathbb{R}_{ovs}}$, il existe une réduction φ en temps polynomial de $\mathrm{KP}_{\mathbb{R}}$ à L. Soit t un polynôme majorant le temps de calcul de φ. Nous allons maintenant décrire un algorithme qui décide A en temps polynomial, avec $\mathrm{KP}'_{\mathbb{R}}[t] \subset A \subset \mathrm{KP}_{\mathbb{R}}$. Soit $x \in \mathbb{R}^n$. Son image par φ est dans $\mathbb{R}^{\leqslant t(n)}$. Soit P_x l'ensemble des points qui suivent le même chemin que x dans φ. En faisant du calcul symbolique le long de ce chemin, on obtient un système d'équations affines décrivant P_x. De plus, on peut calculer en temps polynomial la dimension de P_x. Si $\dim P_x \leqslant n-2$ on rejette l'entrée. Si $\dim P_x = n-1$, on calcule alors une équation de l'adhérence g de P_x. Si g est un hyperplan de $\mathrm{KP}_{\mathbb{R}}$ et $x \in g$ on accepte ; sinon on rejette. Supposons maintenant $\dim P_x = n$. La restriction de φ à P_x est une application affine. Notons L_x sa partie linéaire : $L_x \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ avec $m \leqslant bn^c$. L'algorithme cherche maintenant les hyperplans de $\mathrm{KP}_{\mathbb{R}}$ qui intersectent P_x avec une dimension $n-1$. Supposons qu'il existe un tel hyperplan, et notons h sa direction. Par hypothèse, il existe une constante a telle que $\dim(L \cap \mathbb{R}^n) \leqslant a \log n$. Tout d'abord il est impossible que $\mathrm{rg}\, L_x > a\log(m)+1$. En effet, ceci impliquerait que $\dim L_x(h) > a\log(m)$ et $L_x(h)$ ne pourrait pas être complètement accepté. Donc $\mathrm{rg}\, L_x \leqslant a\log(m)+1 \leqslant a\log(bn^c)+1$, ce qui signifie $\dim \mathrm{Ker} L_x \geqslant n - a\log(bn^c) - 1$. Remarquons que $\mathrm{Ker} L_x \subset h$ sinon tous les points de P_x seraient acceptés. Il reste à trouver tous les hyperplans de $\mathrm{KP}_{\mathbb{R}}$ dont la direction contient $\mathrm{Ker} L_x$. Ceci peut être fait en appliquant l'algorithme suggéré dans la démonstration du lemme 6.1. L'algorithme accepte si x appartient à l'un de ces hyperplans, dont le nombre est majoré par $2^{n-(n-a\log(bn^c)-1)} = n^{O(1)}$. □

Remarque 6.1 *Une variante consiste à accepter le point x quand $\dim P_x \leqslant n-2$, décidant ainsi un langage qui contient $\mathrm{KP}_{\mathbb{R}}$.*

Nous conjecturons que, sous l'hypothèse $\mathrm{P} \neq \mathrm{NP}$, il n'existe pas de t et A tels que $\mathrm{KP}'_{\mathbb{R}}[t] \subset A \subset \mathrm{KP}_{\mathbb{R}}$ et $A \in \mathrm{P}_{\mathbb{R}_{ovs}}$. A présent cette conjecture n'est pas prouvée même pour $t(n) = 1$.

On peut remarquer que l'adhérence de $\mathrm{KP}'_{\mathbb{R}}[t] \cap \mathbb{R}^n$ est $\mathrm{KP}_{\mathbb{R}} \cap \mathbb{R}^n$. On pourrait imaginer qu'il est peut-être facile de calculer l'adhérence d'un langage de $\mathrm{P}^0_{\mathbb{R}_{ovs}}$. La proposition suivante prouve le contraire.

Proposition 6.4 *Les deux énoncés suivants sont équivalents.*
i) Pour tout $A \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$, $\overline{A} \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$.
ii) $\mathrm{P} = \mathrm{NP}$.

Démonstration. Supposons que P = NP et soit $A \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$. On a

$$\overline{A} \cap \mathbb{R}^n = \{(x_1, \ldots, x_n), \forall \varepsilon > 0, \exists (y_1, \ldots, y_n) \in \mathbb{R}^n, \sum_{i=1}^{n} |x_i - y_i| < \varepsilon \wedge (y_1, \ldots, y_n) \in A\}.$$

Ceci montre que $\overline{A} \in (\Pi^2_{\mathbb{R}_{ovs}})^0$. Mais sous l'hypothèse P = NP, on a $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{NP}^0_{\mathbb{R}_{ovs}}$ par le théorème 4.3, et donc $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{PH}^0_{\mathbb{R}_{ovs}}$. Réciproquement, supposons qu'on a le point i). Notons $s(x)$ le signe d'un réel x : $s(x) = 1$ pour $x > 0$ et $s(x) = 0$ pour $x \leqslant 0$. Soit L le langage défini par $L \cap \mathbb{R}^{2n} = \emptyset$ et

$$L \cap \mathbb{R}^{2n+1} = \{(x_1, \ldots, x_n, y_1, \ldots, y_n, y_{n+1}), y_{n+1} \neq 0 \wedge \sum_{i=1}^{n} s(x_i) y_i = y_{n+1}\}.$$

Bien sûr $L \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$. Montrons que $(0, \ldots, 0, y_1, \ldots, y_n, y_{n+1}) \in \overline{L}$ si et seulement si $(y_1, \ldots, y_{n+1}) \in \mathrm{KP}_{\mathbb{R}}$. En effet, si $(y_1, \ldots, y_{n+1}) \in \mathrm{KP}_{\mathbb{R}}$, alors il existe $(\alpha_1, \ldots, \alpha_n) \in \{-1, 1\}^n$, tel que $\sum_{i=1}^{n} s(\alpha_i) y_i = y_{n+1}$. Mais alors $(\varepsilon\alpha_1, \ldots, \varepsilon\alpha_n, y_1, \ldots, y_{n+1}) \in L$ pour tout $\varepsilon > 0$, donc $(0, \ldots, 0, y_1, \ldots, y_{n+1}) \in \overline{L}$. Réciproquement, si $(0, \ldots, 0, y_1, \ldots, y_{n+1}) \in \overline{L}$, alors $(y_1, \ldots, y_n, y_{n+1}) \in \overline{\mathrm{KP}_{\mathbb{R}}} = \mathrm{KP}_{\mathbb{R}}$. Ceci montre que $\mathrm{KP}_{\mathbb{R}} \leqslant_T \overline{L}$ sur $\mathbb{R}_{ovs}$. Comme $L \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$, $\overline{L} \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$ par hypothèse. Or $\mathrm{KP}_{\mathbb{R}}$ est $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$-complet (théorème 4.6), donc $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{NP}^0_{\mathbb{R}_{ovs}}$, et par le théorème 4.3 on obtient P = NP. □

7. Modèle fini plongé dans une structure infinie

On présente dans ce chapitre le modèle des bases de données contraintes [KKR95, KLP00], et quelques résultats sur les fondements théoriques de ce modèle.

7.1 Le modèle des bases de données contraintes

Une signature L est un ensemble de symboles de fonctions f_i et de relations R_i ainsi que la donnée de leurs arités. Une L-structure S est une ensemble M, appelé domaine de la structure, et une interprétation des symboles de L : à chaque symbole de fonction f d'arité n correspond une fonction de $M^n \to M$ et à chaque symbole de relation d'arité n une partie de M^n. Nous n'imposons pas les restrictions du chapitre 1 sur ces structures ; en particulier nous distinguons les fonctions des relations.

Exemples de bases. Les réels avec addition et ordre $(\mathbb{R}, 0, +, -, <)$, le corps ordonné des réels $(\mathbb{R}, 0, 1, +, -, \times, <)$, le corps des complexes $(\mathbb{C}, 0, 1, +, -, \times, =)$.

Autre exemple : la structure ternaire aléatoire $\mathcal{RT}$. C'est l'unique structure à isomorphisme près sur la signature composée d'une relation ternaire qui vérifie les trois propriétés suivantes :

- elle est dénombrable,
- tout modèle fini sur une relation ternaire se plonge dans cette structure,
- tout isomorphisme entre parties finies de cette structure se prolonge en un automorphisme de cette structure.

Cette construction est due à Fraïssé, consulter par exemple [Hod97].

On dit qu'une structure S de domaine M a l'élimination des quantificateurs si pour toute formule du premier ordre $\varphi(\bar{x})$ sur S il existe une formule du premier ordre sans

quantificateurs $\psi(\bar{x})$ sur S telle que

$$S \models \forall \bar{x}\ \varphi(\bar{x}) \leftrightarrow \psi(\bar{x}).$$

On rappelle que les corps algébriquement clos ont l'élimination des quantificateurs. C'est également le cas des corps réels clos (par exemple $\mathbb{R}$). La structure ternaire aléatoire a aussi l'élimination des quantificateurs.

Nous décrivons maintenant le modèle des bases de données contraintes. On se donne les trois objets suivants.

(a) Une signature relationnelle $\mathcal{B}$ composé de symboles de relations $R_1, \ldots, R_s$ et de leurs arités.

(b) Une L-structure infinie S de domaine M.

(c) Une logique $\mathcal{L}$, par exemple la logique du premier ordre.

Étant donné un entier naturel k, une requête Q d'arité k est une application des instances $\mathcal{B}$ vers les modèles d'une relation d'arité k ; si $k = 0$, on parle d'une requête booléenne. Dans l'exemple ci-dessous, la première requête est binaire et la seconde booléenne. Sur ce modèle de base de données, les requêtes s'expriment par des formules de $\mathcal{L}(\mathcal{B} \cup L)$. Toutes les requêtes ne sont bien sûr pas exprimables en général.

Les motivations à l'origine de ce modèle sont les suivantes. D'une part les contraintes permettent d'exprimer des requêtes naturelles dont on a besoin. Dans un contexte géographique par exemple, les contraintes polynomiales sur $\mathbb{R}$ permettent d'exprimer la distance euclidienne. Ensuite, on peut grâce aux contraintes représenter des objets infinis : toujours dans un contexte géographique, cela permet de manipuler uniformément des régions infinies du plan et des objets ponctuels. Enfin, on peut grâce aux contraintes représenter certains objets de manière plus compacte.

Exemple. On prend $\mathcal{B} = \{A, B, V\}$ trois relations binaires, et comme structure infinie sous-jacente le corps ordonné des réels (et la logique du premier ordre). Les symboles A et B correspondent à deux régions du plan et V à un ensemble de villes. Si on cherche l'ensemble des villes de la région A, la requête se formule

$$A(x, y) \wedge V(x, y).$$

Autre requête : existe-t-il une ville de la région A à une distance au plus 1 de la région B ? Ceci s'exprime par la formule

$$\exists x, y A(x, y) \wedge V(x, y) \wedge \exists x', y' B(x', y') \wedge (x - x')^2 + (y - y')^2 \leqslant 1.$$

Si on plonge un graphe dans $\mathbb{R}^k$ avec les contraintes polynomiales et l'ordre, l'usage est de garder comme structure sous-jacente $(\mathbb{R}, 0, 1, +, \times, <)$, et de multiplier l'arité des symboles de la signature des graphes par k. Ainsi le symbole des sommets devient d'arité k et celui des arêtes d'arité $2k$. Cette remarque est bien entendu valable pour d'autres structures et signatures relationnelles.

Dans toute la suite, on se restreint au *cas fini*. Cela signifie que chaque symbole de $\mathcal{B}$ d'arité r interprète un ensemble fini de M^r. Ceci dit, une technique exposée dans

[BST99] permet d'étendre les résultats aux cas où les symboles de $\mathcal{B}$ interprètent des ensembles définis par des formules sans quantificateurs de M. On se limitera également aux requêtes booléennes.

7.2 La question du gain d'expressivité

Nous prenons pour cette section la logique du premier ordre. Que gagne-t-on comme pouvoir d'expression en plongeant un modèle fini dans une structure S ? On peut exprimer de nouvelles requêtes à l'aide de la structure S. Ceci découle directement des opérations et des relations de la structure S. On peut par exemple exprimer des propriétés sur la distance entre les sommets d'un graphe plongé dans le corps ordonné des réels, ce qu'on ne peut pas faire sans les opérations de $\mathbb{R}$. En un sens, ce type de gain d'expressivité est souvent le seul obtenu. Pour préciser ceci, introduisons la notion de requête générique.

Une requête booléenne Q est dite *générique* si pour toute permutation σ de M et toute interprétation finie B de $\mathcal{B}$ on a $Q(\sigma(B)) = Q(B)$. On sait qu'un ordre total permet d'exprimer plus de requêtes que le premier ordre pur (proposition 7.2). C'est pourquoi dans le cas d'une structure totalement ordonnée, on se pose la question du gain d'expressivité qu'apportent les autres opérations. On définit pour cela la notion de requête booléenne localement générique. Une requête booléenne Q est dite *localement générique* si pour toute base de donnée finie B et toute application strictement croissante σ du domaine actif D de B vers M, on a $Q(\sigma(B)) = Q(B)$.

Exemple 1. Considérons un graphe plongé dans $\mathbb{R}$. Les requêtes concernant la connexité, le nombre de sommets, le nombre de composantes connexes du graphe sont des requêtes booléennes génériques. L'ensemble des sommets qui sont dans la même composante connexe qu'un sommet donné en entrée est aussi une requête générique. Par contre, demander si 2 est un sommet, ou s'il existe trois sommets x, y, z vérifiant $x = y + z$ sont des requêtes non génériques.

Exemple 2. Considérons un modèle fini de signature $\mathcal{B}$ composée de deux prédicats unaires A et B, plongé dans une structure totalement ordonnée. Demander si tous les éléments de A sont plus petits que ceux de B est une requête localement générique mais pas générique.

On dit qu'une L-structure a l'*effondrement générique* si toute requête générique qui s'exprime au premier ordre sur $L \cup \mathcal{B}$ s'exprime au premier ordre sur $\mathcal{B}$ seulement. De même, une L-structure totalement ordonnée par $<$ a l'*effondrement localement générique* si toute requête localement générique qui s'exprime au premier ordre sur $L \cup \mathcal{B}$ s'exprime au premier ordre sur $\mathcal{B} \cup \{<\}$.

Proposition 7.1 *Si une structure S a l'effondrement générique, alors toute structure élémentairement équivalente à S a l'effondrement générique. La même remarque vaut pour l'effondrement localement générique.*

De nombreux travaux ont été menés pour chercher des propriétés suffisantes pour l'effondrement (localement) générique. Les propriétés de théorie des modèles ont permis d'identifier une large classe de structures présentant ce comportement. Les résultats les plus généraux dans ce domaine ont été montrés par Baldwin et Benedikt. On rappelle qu'une structure est dite stable s'il n'existe pas une formule $\phi(\bar{x}, \bar{y})$ et une suite de uples $(\bar{a}_i)_{i\in\mathbb{N}}$ vérifiant $\phi(\bar{a}_m, \bar{a}_n) \leftrightarrow m < n$.

Théorème 7.1 *[BB00] Les structures stables ont l'effondrement générique.*

On dit qu'une structure a la propriété d'indépendance s'il existe une formule $\phi(\bar{x}, \bar{y})$ telle que pour tout $m \in \mathbb{N}$, il existe une suite de uples $\bar{b}_1, \ldots, \bar{b}_m$ vérifiant : pour tout $X \subset \{1, \ldots, m\}$, il existe $\bar{a}$ tel que $\phi(\bar{b}_i, \bar{a})$ si et seulement si $i \in X$. Une autre formulation est la suivante : une structure de domaine M a la propriété d'indépendance s'il existe une formule $\phi(\bar{x}, \bar{y})$ tel que le système géométrique $(M^k, \{\phi(M^k, \bar{a}) \mid \bar{a} \in M^n\})$ soit de VC-dimension infinie – où $\phi(M^k, \bar{a})$ est $\{\bar{x} \in M^k, \phi(\bar{x}, \bar{a})\}$.

Théorème 7.2 *[BB00] Les structures totalement ordonnées (par un prédicat $<$) sans la propriété d'indépendance ont la propriété d'effondrement localement générique.*

Une structure totalement ordonnée par un prédicat $<$ est dite o-minimale si tout ensemble définissable est une union finie d'intervalles. Les structures sans la propriété d'indépendance englobent les structures o-minimales, le corps $\mathbb{R}$ par exemple.

On connaît également des structures qui n'ont pas l'effondrement générique, par exemple la structure ternaire aléatoire. On peut en effet exprimer au premier ordre qu'un ensemble fini I plongé dans cette structure est pair [BL00]. Cela vient de l'existence de sommets témoignant de toute relation binaire sur un ensemble fini quelconque de sommets. Cela permet donc d'exprimer qu'il existe une relation de couplage sur I, ce qui est équivalent à la parité de $|I|$.

7.3 Quantifications actives

On considère une signature relationnelle $\mathcal{B} = \{R_1, R_2, \ldots, R_s\}$ et une L-structure S de domaine M. Dans cette section encore, on ne s'intéresse qu'aux modèles finis de $\mathcal{B}$ (chaque relation de $\mathcal{B}$ interprète un ensemble fini). Étant donnée un modèle fini de B, on notera D le domaine actif de B, c'est-à-dire l'ensemble des coordonnées des points de B.

On ne prend plus dans cette section la logique du premier ordre, mais une variante de celle-ci : la logique du premier ordre active FO_{act}. On définit pour cela deux nouveaux quantificateurs, ce sont $\exists^a$ et $\forall^a$. Ces quantificateurs parcourent le domaine actif uniquement : autrement dit, $\exists^a x$ signifie $\exists x \in D$. Les quantificateurs parcourant l'ensemble du domaine M ne sont plus permis. Pour le reste, les formules actives du premier ordre sont construites comme les formules usuelles (appelées à partir de maintenant formules naturelles).

Considérons un modèle fini de $\mathcal{B}$ plongé dans une structure S avec la logique active (du premier ordre). Bien sûr tout ce que l'on peut exprimer au premier ordre sur $\mathcal{B}$ peut

être exprimé par une formule active sur $\mathcal{B} \cup L$. De même, tout ce qu'on peut exprimer au premier ordre actif sur $\mathcal{B} \cup L$ peut s'exprimer par une formule naturelle. Pour résumer

$$\mathrm{FO}(\mathcal{B}) \subset \mathrm{FO}_{act}(\mathcal{B} \cup L) \subset \mathrm{FO}(\mathcal{B} \cup L).$$

Pour une structure S donnée, quelles inclusions sont strictes ?

La première question est pertinente si on se restreint encore une fois aux requêtes génériques. Considérons pour commencer le cas d'une structure S composée uniquement d'un ordre total $<$ sur son domaine M. L'ordre permet d'exprimer de nouvelles requêtes génériques.

Proposition 7.2 *[AHV95, BST99, BB00] Il existe une requête générique qu'on ne peut exprimer au premier ordre pur, mais qu'on peut exprimer au premier ordre en présence d'un ordre total.*

Démonstration. On prend pour $\mathcal{B}$ la signature des algèbres de Boole. La requête est la suivante : on demande que le modèle de $\mathcal{B}$ soit une algèbre de Boole engendrée par un nombre pair d'atomes. Notons qu'une quantification sur la structure permet de simuler le second ordre monadique (c'est-à-dire de quantifier sur les relations unaires) *sur les atomes*. La requête ci-dessus n'est pas exprimable au premier ordre pur. Par contre, en présence d'un ordre total, on peut exprimer cette requête. On peut par exemple exprimer qu'il existe un ensemble des atomes qui contient le plus petit atome, qui ne contient pas le plus grand, et qui, parmi deux atomes successifs quelconques, en contient toujours exactement un. □

Ceci dit, ce gain venant de l'ordre est le seul possible. Ce résultat, dû à Otto et Van den Bussche, repose sur la notion de suite indiscernable. Un ensemble $X \subset M$ totalement ordonné par $<$ est appelé une *suite indiscernable* (pour l'ordre $<$) si deux uples quelconques de X strictement croissants (pour $<$) vérifient les mêmes formules du premier ordre.

Théorème 7.3 (Ehrenfeucht-Mostowski) *[Hod97, CK90] Toute théorie complète T avec un ordre total $<$ admet un modèle contenant une suite indiscernable infinie pour l'ordre $<$.*

Théorème 7.4 *[OVdB96] Toute requête générique qu'on peut exprimer au premier ordre actif avec une structure sous-jacente peut en fait s'exprimer au premier ordre à l'aide d'un ordre total uniquement.*

Démonstration. Soit S une L-structure de domaine M et $\phi \in \mathrm{FO}_{act}(\mathcal{B} \cup L)$ une formule exprimant une requête générique Q. On peut supposer sans perte de généralité qu'il n'y a pas de formules atomiques de ϕ faisant intervenir à la fois des symboles de $\mathcal{L}$ et des symboles de $\mathcal{B}$ (en ajoutant des variables qu'il suffit de quantifier activement, puisqu'elles apparaissent dans une relation de $\mathcal{B}$). Considérons $L' = L \cup \{\prec\}$. Par le théorème 7.3, il existe une L'-structure S' dont la restriction à L est élémentairement équivalente à S, et contenant une séquence indiscernable infinie $(I, \prec)$. Par équivalence

élémentaire, la formule ϕ exprime encore Q pour un modèle fini plongé dans S'. Et ceci en particulier quand on se restreint au cas où les coordonnées du modèle fini sont dans I. Mais dans ce cas, on peut trouver une formule active du premier ordre exprimant Q mais n'utilisant que l'ordre $\prec$ et les symboles de $\mathcal{B}$. En effet, pour chaque relation $R \in L$, savoir si $S' \models R(a_1, \ldots, a_k)$, pour des éléments $a_i \in I$ ne dépend que de l'ordre des a_i pour $\prec$. La requête Q s'exprime donc bien au premier ordre à l'aide d'un ordre total uniquement. □

Un ensemble $X \subset M$ est appelé *ensemble indiscernable* si deux quelconques uples de X de même longueur, chacun de ces uples ayant des coordonnées toutes distinctes, vérifient les mêmes formules du premier ordre. Par le même raisonnement que celui mis en œuvre au théorème 7.4, on sait que sur une théorie T admettant un ensemble indiscernable infini X, toute requête générique qui s'exprime au premier ordre actif (à l'aide d'une structure modèle de T) s'exprime au premier ordre pur.

Exemple. Soit K un corps algébriquement clos de degré de transcendance infini. Une base de transcendance de K est clairement un ensemble indiscernable de K. On peut donc en conclure que toute requête générique exprimable au premier ordre actif avec comme structure sous-jacente un corps algébriquement clos s'exprime en fait au premier ordre pur. Plus généralement, toute suite indiscernable sur une structure stable est un ensemble (totalement) indiscernable [Poi00].

La propriété de Ramsey, basée comme le théorème 7.3 sur le théorème de Ramsey [GRS90, Juk01] mais se limitant à une seule formule, permet de traiter par cette même méthode d'autres logiques (logique de point fixe par exemple). Consulter [BL00] et [KLP00] à ce sujet.

On s'intéresse maintenant à l'inclusion $\mathrm{FO}_{act}(\mathcal{B} \cup L) \subset \mathrm{FO}(\mathcal{B} \cup L)$. On dit qu'une structure S a l'*effondrement actif-naturel* si, pour toute signature relationnelle $\mathcal{B}$ et toute formule du premier ordre ϕ sur $\mathcal{B} \cup L$, il existe une formule active du premier ordre ψ sur $\mathcal{B} \cup L$ telle que pour toute base de données finie B, on ait $(S, B) \models \phi \leftrightarrow \psi$. Cette propriété implique bien sûr l'élimination des quantificateurs.

Proposition 7.3 *Si une structure S a l'effondrement actif-naturel, alors toute structure élémentairement équivalente à S a l'effondrement actif-naturel.*

Exemple. On prend une signature relationnelle composée d'une relation binaire E, et pour structure le corps $\mathbb{R}$. Les points de E sont-ils alignés ? Cette requête s'exprime par la formule naturelle

$$\exists (a,b,c) \neq (0,0,0)\ \forall x, y E(x,y) \rightarrow ax + by + c = 0.$$

Est-il possible d'exprimer cette même requête à l'aide d'une formule active ? Oui, une façon de faire est la suivante :

$$|E| \leqslant 1 \vee \exists^a x_1, y_1, x_2, y_2\ (x_1, y_1) \neq (x_2, y_2) \wedge E(x_1, y_1) \wedge E(x_2, y_2) \wedge \forall^a x, y\ E(x,y) \rightarrow \begin{vmatrix} x - x_1 & x_2 - x_1 \\ y - y_1 & y_2 - y_1 \end{vmatrix} = 0.$$

L'exemple précédent n'est pas une exception comme le montre ce théorème de Benedikt et Libkin.

Théorème 7.5 *[BL96] Le corps ordonné des réels admet l'effondrement actif-naturel.*

On connaît également des structures qui n'ont pas la propriété d'effondrement actif-naturel. Par exemple la structure ternaire aléatoire [BL00], ou encore $(\mathbb{N}, +, \times)$.

Une condition suffisante pour l'effondrement actif-naturel, due à Flum et Ziegler, est basée sur la NFCP (No Finite Cover Property). Une structure M a la NFCP si pour toute formule $\phi(x, y)$ il existe une borne b vérifiant la propriété suivante. Soit $\mathcal{C}$ l'ensemble des $\phi(a, M)$; si tout sous-ensemble de $\mathcal{C}$ de cardinal au plus b a une intersection non vide, alors $\mathcal{C}$ a une intersection non vide.

Théorème 7.6 *[FZ99] Une structure éliminant les quantificateurs qui a la NFCP admet la propriété d'effondrement actif-naturel.*

Notons que ces résultats sont basés sur des jeux de va-et-vient. Une preuve *constructive* de l'effondrement actif-naturel sur les structures o-minimales avec élimination des quantificateurs est donnée dans [BL97, BL00]. D'autres résultats d'effondrement actif-naturel basés sur cette méthode sont traités au chapitre 9.

Une structure qui a la NFCP est stable (mais la réciproque n'est pas vraie). On ne peut pas généraliser le théorème 7.6 au cas des structures stables [FZ99]. En effet, la théorie d'une relation d'équivalence R ayant exactement une classe de chaque cardinal entier $n \geqslant 1$ est stable (elle est ω-stable) mais n'admet pas l'effondrement actif-naturel : on ne peut pas exprimer par une formule active du premier ordre que l'ensemble fini interprété par un prédicat unaire I est une classe d'équivalence de R.

8. Rang de quantification pour la parité et la connexité

De nombreux travaux ont été menés concernant le gain d'expressivité obtenu en plongeant un modèle fini dans une structure infinie M. De tels résultats sont rappelés au chapitre 7. Par exemple, la requête demandant si le cardinal d'un ensemble fini $\mathcal{I}$ plongé dans une structure M est pair est générique, et on ne peut donc pas l'exprimer au premier ordre sur de nombreuses structures M telles que les corps des réels et des complexes (théorèmes 7.1 et 7.2). Cependant, si on se restreint au cas où $|\mathcal{I}|$ est plus petit qu'une borne donnée, alors la requête de parité peut bien sûr s'exprimer par une formule. On se demande alors quel est le rang de quantification minimum possible pour cette formule. Peut-on faire mieux que dans le cas d'un modèle fini sans structure englobante ? Ce chapitre provient de [Fou01a].

8.1 Introduction

Les principaux résultats sont résumé figure 8.1. La première colonne présente le rang de quantification nécessaire pour exprimer qu'un ensemble de cardinal majoré par n est pair, quand cet ensemble est plongé dans l'une des structures à lire sur la gauche. La seconde colonne donne des résultats concernant la connexité d'un graphe plongé dans une structure, n étant cette fois une borne sur le nombre de sommets du graphe. Les deux premières lignes sont des bornes bien connues de théorie des modèles finis. En les comparant aux lignes suivantes, on peut comprendre comment l'addition et le produit permettent ou non d'améliorer ces bornes. Prenons l'exemple de la parité sur $(\mathbb{C}, =)$. Si on ajoute l'addition, on fait baisser le rang de quantification de n à $\lceil \log n \rceil + 1$. Par contre, ajouter ensuite la multiplication ne mène à aucun gain de rang de quantification. Sur les réels avec ordre, l'addition permet de diminuer le rang de quantification pour la parité de $\log n + \Theta(1)$ à $\Theta(\sqrt{\log n})$. Il serait intéressant de préciser ces bornes dans le cas des corps réels clos, par exemple $(\mathbb{R}, +, -, \times, \leqslant)$ – voir à ce propos la question

8.1. Notons que les résultats obtenus redémontrent que la parité et la connexité ne sont pas exprimables au premier ordre sur les structures considérées. Enfin, il est montré dans [GS97] que la parité et la connexité sont expressibles sur $(\mathbb{Q}, +, -, \times, =)$: ceci repose sur la définissabilité des entiers sur cette structure [Rob49, FW91], permettant d'obtenir le pouvoir de l'arithmétique.

Remarque. Il est souvent naturel de considérer un graphe plongé dans M^k au lieu de M – penser aux bases de données géographiques et à $M = \mathbb{R}$. Dans ce cas, l'usage est de considérer la structure M munie des opérations usuelles sur M et de multiplier par k les arités des prédicats des sommets et des arêtes du graphe. Les résultats de ce chapitre sont encore valables dans ce cas : les bornes inférieures restent inchangées, et les bornes supérieures sont multipliées par k.

	pair	connexe
$(\mathbb{Q}, =)$, $(\mathbb{R}, =)$, $(\mathbb{C}, =)$	n	$\Theta(\log n)$
$(\mathbb{Q}, <)$, $(\mathbb{R}, <)$	$\log n + \Theta(1)$	$\Theta(\log n)$
$(\mathbb{Q}, +, -, =)$, $(\mathbb{R}, +, -, =)$, $(\mathbb{C}, +, -, =)$	$\lceil \log n \rceil + 1$	$\Theta(\sqrt{\log n})$
$(\overline{\mathbb{Q}}, +, -, \times, =)$, $(\mathbb{C}, +, -, \times, =)$	$\lceil \log n \rceil + 1$	$\Theta(\sqrt{\log n})$
$(\mathbb{Q}, +, -, <)$, $(\mathbb{R}, +, -, <)$	$\Theta(\sqrt{\log n})$	$\Theta(\sqrt{\log n})$
$(\mathbb{R}, +, -, \times, <)$	$\Omega(\log\log n), O(\sqrt{\log n})$	$\Omega(\log\log n), O(\sqrt{\log n})$
$(\mathbb{Q}, +, -, \times, =)$, $(\mathbb{Q}, +, -, \times, <)$	$\Theta(1)$	$\Theta(1)$

FIG. 8.1 – Rang de quantification pour la parité et la connexité

La question qui nous intéresse dans ce chapitre est définie précisément dans la section 8.2. Dans la section 8.3, on montre des bornes inférieures et des bornes supérieures pour la parité sur les corps algébriquement clos de caractéristique nulle et sur les $\mathbb{Q}$-espaces vectoriels (par exemple les réels avec addition). La section 8.4 traite le cas des corps réels clos (par exemple les réels avec addition et ordre). On déduit de ces résultats des bornes sur la connexité d'un graphe section 8.5. Enfin, la dernière section établit des relations entre ces bornes et la notion d'effondrement actif-naturel.

8.2 Notations et premières remarques

On s'intéresse au problème suivant. On plonge un ensemble fini $\mathcal{I}$ dans un corps algébriquement clos ou dans un $\mathbb{Q}$-espace vectoriel ordonné : on va appeler M cette structure. En plus de la signature de M, on a donc un nouveau prédicat unaire I qui interprète $\mathcal{I}$. On va tout particulièrement s'intéresser à la requête Pair, qui demande si $|\mathcal{I}|$ est pair et Card_m, qui demande si $|\mathcal{I}| \geqslant m$. Pour une requête Q, on note $\mathrm{QR}_M(Q, n)$ le minimum des rang de quantification des formules du premier ordre exprimant Q sous l'hypothèse $|\mathcal{I}| \leqslant n$ – autrement dit, on demande que la formule exprime Q si $|\mathcal{I}| \leqslant n$ et

on se moque de ce qu'elle exprime quand $|\mathcal{I}| > n$. Notre but est de trouver des bornes sur $\mathrm{QR}_M(\mathrm{Pair}, n)$. On rappelle que le rang de quantification $\mathrm{qr}(\phi)$ d'une formule ϕ est définie par récurrence sur sa structure. Si ϕ est une formule atomique, $\mathrm{qr}(\phi) = 0$. Sinon $\mathrm{qr}(\phi \vee \psi) = \mathrm{qr}(\phi \wedge \psi) = \max(\mathrm{qr}(\phi), \mathrm{qr}(\psi))$ et $\mathrm{qr}(\exists x \phi) = \mathrm{qr}(\forall x \phi) = 1 + \mathrm{qr}(\phi)$. Commençons par la remarque suivante.

Lemme 8.1 *Si deux structures M et M' sont élémentairement équivalentes, alors pour tout n_0, $\mathrm{QR}_M(\mathrm{Pair}, n_0) = \mathrm{QR}_{M'}(\mathrm{Pair}, n_0)$.*

Démonstration. Soit n_0 fixé et supposons qu'on a une formule du premier ordre ϕ tel que si $|\mathcal{I}| \leqslant n_0$, $(M, \mathcal{I}) \models \phi$ si et seulement si $|\mathcal{I}|$ est pair. Soit M' une structure élémentairement équivalente à M, et $n \leqslant n_0$. Soit $\tilde{\phi}(x_1, \ldots, x_n)$ la formule ϕ dans laquelle $I(x)$ a été remplacé par $\bigvee_{i=1}^{n} x = x_i$, et $\psi_n = \forall x_1, \ldots, x_n \ \bigwedge_{i<j} x_i \neq x_j \to \tilde{\phi}(x_1, \ldots, x_n)$. Si n est pair, $M' \models \psi_n$ puisque $M \models \psi_n$; et si n est impair, $M' \models \neg\psi_n$. Ainsi si $\mathcal{I}' \subset M'$ avec $|\mathcal{I}'| \leqslant n_0$, $(M', \mathcal{I}') \models \phi$ si $|\mathcal{I}'|$ est pair et $(M', \mathcal{I}') \models \neg\phi$ si $|\mathcal{I}'|$ est impair. Donc $\mathrm{QR}_{M'}(\mathrm{Pair}, n_0) \leqslant \mathrm{QR}_M(\mathrm{Pair}, n_0)$ et par symétrie $\mathrm{QR}_{M'}(\mathrm{Pair}, n_0) = \mathrm{QR}_M(\mathrm{Pair}, n_0)$. □

Bien sûr la remarque précédente s'applique aussi à Card_m. Ceci justifie les notations $\mathrm{QR}_T(\mathrm{Pair}, n)$ et $\mathrm{QR}_T(\mathrm{Card}_m, n)$ pour une théorie complète T. Introduisons des notations pour les théories qui vont nous intéresser, et donnons quelques exemples de modèles de ces théories.

- corps algébriquement clos de caractéristique nulle : ACF_0
  exemples : $(\overline{\mathbb{Q}}, +, -, \times, =)$, $(\mathbb{C}, +, -, \times, =)$.
- $\mathbb{Q}$-espace vectoriel : $\mathbb{Q}vs$
  exemples : $(\mathbb{Q}, +, -, =)$, $(\mathbb{R}, +, -, =)$, $(\mathbb{C}, +, -, =)$.
- $\mathbb{Q}$-espace vectoriel ordonné : $\mathcal{O}vs$
  exemples : $(\mathbb{Q}, +, -, <)$, $(\mathbb{R}, +, -, <)$.

Notons que nous traitons en fait le cas de tout corps algébriquement clos et de tout espace vectoriel infini (remarque 8.1). Notre outil principal sera les jeux de va-et-vient d'Ehrenfeucht et Fraïssé [EF95, Imm98, Hod97, Poi00]. Considérons M et N deux $\mathcal{L}$-structures. Un jeu de longueur n entre deux structures M et N se déroule comme ceci. Au coup i, le premier joueur choisit un point de M ou de N ; le second joueur doit alors choisir un point de l'autre structure. Appelons a_i le point de M choisi à l'étape i, et b_i celui de N choisi à cette même étape. Après n coups la partie s'arrête, et le second joueur a gagné si et seulement si les mêmes formules atomiques sont vraies dans les structures $(M, a_1, \ldots, a_n)$ et $(N, b_1, \ldots, b_n)$. On dit que le second joueur a une stratégie gagnante pour le jeu de va-et-vient de longueur n entre M et N s'il peut gagner quelle que soit la façon dont le premier joueur joue. On va utiliser la propriété fondamentale des jeux d'Ehrenfeucht-Fraïssé.

Fait 8.1 *Si le second joueur a une stratégie gagnante pour gagner le jeu de va-et-vient de longueur n entre les deux structures M et N, alors les mêmes formules de rang de quantification au plus n sont vraies dans M et N.*

La méthode fonctionne comme ceci. Si on veut montrer que $\mathrm{QR}_T(\mathrm{Pair}, n) > B$ pour une théorie complète T de signature $\mathcal{L}$, on choisit deux modèles M et N de T et deux

ensembles finis $\mathcal{I} \subset M$ et $\mathcal{J} \subset N$ de cardinaux au plus n, avec $|\mathcal{I}|$ pair et $|\mathcal{J}|$ impair. Maintenant si on prouve que le second joueur a une stratégie gagnante pour le jeu de longueur B entre les deux $\mathcal{L} \cup \{I\}$-structures $(M, \mathcal{I})$ et $(N, \mathcal{J})$, alors aucune formule du premier ordre sur $\mathcal{L} \cup \{I\}$ de rang de quantification au plus B ne peut exprimer la parité restreinte recherchée.

Examinons les bornes que l'on obtient lorsqu'on n'a pas de structure sous-jacente. Nous noterons $\mathrm{QR}_<(\mathrm{Pair}, n)$ le rang de quantification minimal pour exprimer la parité (jusqu'à n) dans le cas où la structure est munie d'un ordre total, et $\mathrm{QR}_=(\mathrm{Pair}, n)$ en absence d'ordre. Par des jeux de va-et-vient classique [EF95, Imm98], on a $\mathrm{QR}_=(\mathrm{Pair}, n) = n$ et $\mathrm{QR}_<(\mathrm{Pair}, n) = \log n + \Theta(1)$.

8.3 Parité sur une structure sans ordre

Dans cette section, on prouve les résultats concernant la parité sur les corps algébriquement clos de caractéristique nulle et les $\mathbb{Q}$-espaces vectoriels. Comme un corps algébriquement clos de caractéristique nulle est un $\mathbb{Q}$-espace vectoriel, $\mathrm{QR}_{ACF_0}(\mathrm{Pair}, n) \leqslant \mathrm{QR}_{\mathbb{Q}vs}(\mathrm{Pair}, n)$. Cette section est donc divisée en deux parties : une borne inférieure sur $\mathrm{QR}_{ACF_0}(\mathrm{Pair}, n)$, puis une borne supérieure sur $\mathrm{QR}_{\mathbb{Q}vs}(\mathrm{Pair}, n)$.

8.3.1 Borne inférieure sur un corps algébriquement clos

On note $\overline{A}$ la clôture algébrique de $A \subset \mathbb{C}$. Dans cette section, on prouve la borne inférieure suivante.

Théorème 8.1 $\mathrm{QR}_{ACF_0}(\mathrm{Pair}, n) \geqslant \lceil \log n \rceil + 1$.

Démonstration. Par le lemme 8.1, il est suffisant de prouver cette borne dans un corps algébriquement clos de caractéristique nulle donné. On va travailler dans le corps des complexes $\mathbb{C}$. Soit $\mathcal{M}$ un ensemble de 2^{n-1} éléments algébriquement indépendants de $\mathbb{C}$, et $\mathcal{N}$ un ensemble de $2^{n-1}+1$ éléments algébriquement indépendants de $\mathbb{C}$. On va prouver que le second joueur peut gagner le jeu de va-et-vient de longueur n entre $(\mathbb{C}, \mathcal{M})$ et $(\mathbb{C}, \mathcal{N})$.

Au début du jeu, posons $E_n = F_n = \overline{\mathbb{Q}}$ et $\varphi = \mathrm{Id}_{\overline{\mathbb{Q}}}$. Dans la suite, φ est une application partielle de la première structure $(\mathbb{C}, \mathcal{M})$ vers la seconde structure $(\mathbb{C}, \mathcal{N})$, son domaine (resp. image) contenant les points choisis par les deux joueurs dans la première (resp. seconde) structure. À chaque pas on étend φ de sorte que son domaine et son image contiennent les points nouvellement choisis. Quand il reste j coups à jouer, on note E_j le corps où φ est défini et $F_j = \varphi(E_j)$. À chaque étape, φ est un isomorphisme de corps algébriquement clos "avec points" de E_j sur F_j : ceci signifie que pour tout x dans E_j, $x \in \mathcal{M}$ si et seulement si $\varphi(x) \in \mathcal{N}$. On maintient aussi la propriété $\mathcal{P}_j$ suivante.

Tout d'abord $|\mathcal{M} \setminus E_j|, |\mathcal{N} \setminus F_j| \geqslant 2^{j-1}$. *De plus, s'il existe* $a \in \mathcal{M} \setminus E_j$ *et* $A \subset \mathcal{M} \setminus (E_j \cup \{a\})$ *tels que* $a \in \overline{E_j \cup A}$, *alors* $|A| \geqslant 2^{j-1}$. *Et la propriété correspondante dans* $(\mathbb{C}, \mathcal{N})$ *: s'il existe* $a \in \mathcal{N} \setminus F_j$ *et* $A \subset \mathcal{N} \setminus (F_j \cup \{a\})$ *tels que* $a \in \overline{F_j \cup A}$,

alors $|A| \geqslant 2^{j-1}$.

Vérifions que la propriété $\mathcal{P}_n$ est vérifiée au début du jeu. On a $|\mathcal{M} \setminus E_n| = |\mathcal{M}| \geqslant 2^{n-1}$. De plus, comme les éléments de $\mathcal{M}$ sont algébriquement indépendants sur $\mathbb{Q}$, il n'existe pas de $a \in \mathcal{M}$ avec $a \in \overline{\mathbb{Q} \cup A}$ tel que $A \subset \mathcal{M}$ et $a \notin A$. Ce qui est également vrai sur $(\mathbb{C}, \mathcal{N})$.

Supposons que $n - j - 1$ coups ont été joués. L'isomorphisme φ est défini sur E_{j+1} et il reste $j + 1$ coups à jouer. La propriété $\mathcal{P}_{j+1}$ est vérifiée par hypothèse d'induction. Par symétrie, on peut supposer que le point est choisi dans $(\mathbb{C}, \mathcal{M})$. Notons v ce point. On peut aussi supposer que $v \notin E_{j+1}$. Deux cas sont possibles.

Premier cas : $v \in \overline{E_{j+1} \cup \{a_1, \ldots, a_r\}}$ avec $a_i \in \mathcal{M} \setminus E_{j+1}$ distincts et $r \leqslant 2^{j-1}$. Alors on choisit des éléments distinct $b_1, \ldots, b_r$ dans $\mathcal{N} \setminus F_{j+1}$et on définit $\varphi(a_i) = b_i$. Ainsi $E_j = \overline{E_{j+1} \cup \{a_1, \ldots, a_r\}}$. On pose $F_j = \varphi(E_j)$ et on étend φ en un isomorphisme de corps de E_j sur F_j. Montrons que la propriété $\mathcal{P}_j$ est vérifiée. S'il existe $d \in \mathcal{M} \setminus E_j$, avec $d \in \overline{E_j \cup \{c_1, \ldots, c_l\}}$, $c_i \in \mathcal{M} \setminus (E_j \cup \{d\})$ et $l \leqslant 2^{j-1} - 1$, alors $d \in \overline{E_{j+1} \cup \{a_1, \ldots, a_r, c_1, \ldots, c_l\}}$. Mais $r + l \leqslant 2^{j-1} + 2^{j-1} - 1 = 2^j - 1$. En conséquence on devrait avoir $d \in E_{j+1}$ par la propriété $\mathcal{P}_{j+1}$, ce qui est absurde. On a la même propriété dans $(\mathbb{C}, \mathcal{N})$. De plus, $|\mathcal{M} \setminus E_j|, |\mathcal{N} \setminus F_j| \geqslant 2^j - 2^{j-1} = 2^{j-1}$ ce qui achève de montrer que $\mathcal{P}_j$ est vérifiée. Exactement de la même façon, on montre qu'il n'existe pas d'autres points de $\mathcal{M} \setminus E_{j+1}$ dans E_j en plus des a_i : si $d \in (\mathcal{M} \cap E_j) \setminus (E_{j+1} \cup \{a_1, \ldots, a_r\})$, alors $d \in \overline{E_{j+1} \cup \{a_1, \ldots, a_r\}}$ et on conclut avec $\mathcal{P}_{j+1}$. C'est aussi vrai dans $(\mathbb{C}, \mathcal{N})$, et cela montre que φ est un isomorphisme.

Second cas : si on n'est pas dans le premier cas, soit $f \notin \overline{F_{j+1} \cup \mathcal{N}}$. Soit $\varphi(v) = f$. On pose $E_j = \overline{E_{j+1} \cup \{v\}}$, $F_j = \varphi(E_j)$ et on étend φ en un isomorphisme de corps de E_j sur F_j. Montrons que $\mathcal{P}_j$ est vérifiée. Soit $a \in \mathcal{M} \setminus E_j$ tel que $a \in \overline{E_j \cup A}$ pour $A \subset \mathcal{M} \setminus (E_j \cup \{a\})$ avec $|A| < 2^{j-1}$. Ainsi $a \in \overline{E_{j+1} \cup \{v\} \cup A}$. Ceci montre que $v \in \overline{E_{j+1} \cup \{a\} \cup A}$, parce que $a \in \overline{E_{j+1} \cup A}$ est impossible par $\mathcal{P}_{j+1}$. Mais alors on devrait être dans le premier cas puisque $|A \cup \{a\}| \leqslant 2^{j-1}$. Cela est aussi vrai dans $(\mathbb{C}, \mathcal{N})$ par choix de f. De plus, il n'existe pas de point de $\mathcal{M}$ dans $E_j \setminus E_{j+1}$ parce que si $a \in \mathcal{M} \cap E_j \setminus E_{j+1}$, alors $a \in \overline{E_{j+1} \cup \{v\}}$ et comme $a \notin E_{j+1}$ on aurait $v \in \overline{E_{j+1} \cup \{a\}}$ ce qui est absurde. Ceci est aussi vrai dans $(L, \mathcal{N})$ par choix de f, et φ reste donc un isomorphisme. De plus, $|\mathcal{M} \setminus E_j| = |\mathcal{M} \setminus E_{j+1}| \geqslant 2^{j-1}$ ce qui finit de prouver $\mathcal{P}_j$.

Ceci clôt le jeu de va-et-vient. On a ainsi prouvé $\mathrm{QR}_{ACF_0}(\mathrm{Pair}, 2^{n-1}+1) > n$. Comme $\mathrm{QR}_{ACF_0}(\mathrm{Pair}, \cdot)$ est une fonction croissante, on obtient $\mathrm{QR}_{ACF_0}(\mathrm{Pair}, n) \geqslant \lceil \log n \rceil + 1$. □

8.3.2 Borne supérieure dans un $\mathbb{Q}$-espace vectoriel

La démonstration est construite en trois étapes.

- On montre tout d'abord qu'il est possible d'exprimer que $|\mathcal{I}| \geqslant m$ avec une formule de rang de quantification $\lceil \log m \rceil + 2$ dans le cas particulier où les éléments de $\mathcal{I}$ sont linéairement indépendants sur $\mathbb{Q}$.
- On généralise ensuite cette borne au cas général.

– Enfin, on montre comment diminuer de 1 le rang de quantification de ces formules.

On a besoin de construire des formules $S_{(\alpha_1,\ldots,\alpha_p)}(x)$ pour $p \geqslant 1$ et $(\alpha_1,\ldots,\alpha_p) \in \mathbb{N}^p$. Ces formules doivent satisfaire ceci :

$$(M,\mathcal{I}) \models S_{(\alpha_1,\ldots,\alpha_p)}(x) \iff \exists x_1,\ldots,x_p \in \mathcal{I}\ x = \sum_{i=1}^{p} \alpha_i x_i.$$

De plus, on va construire de telles formules avec un petit rang de quantification. On les définit de la manière suivante : on prend $S_{(1)}(x) := I(x)$, et $S_{(\alpha_1)}(x) := \exists y I(y) \wedge x = y + \ldots + y$ (où y est ajouté a lui-même α_1 fois) pour $\alpha_1 \neq 1$. Enfin, on définit

$$S_{(\alpha_1,\ldots,\alpha_p)}(x) := \exists y\ S_{(\alpha_1,\ldots,\alpha_{\lfloor p/2 \rfloor})}(y) \wedge S_{(\alpha_{\lceil p/2 \rceil},\ldots,\alpha_p)}(x-y).$$

On vérifie que le rang de quantification de $S_{(\alpha_1,\ldots,\alpha_p)}(x)$ est majoré par $\lceil \log p \rceil + 1$.

Proposition 8.1 *Dans un $\mathbb{Q}$-espace vectoriel, si on se restreint au cas où les éléments de $\mathcal{I}$ sont linéairement indépendants sur $\mathbb{Q}$, alors on peut exprimer que $|\mathcal{I}| \geqslant m$ avec une formule de rang de quantification $\lceil \log m \rceil + 2$.*

Preuve. Soit $\bar{\alpha}_m = (1,1,\ldots,1) \in \mathbb{N}^m$ et $\bar{\beta}_m = (2,1,1,\ldots,1) \in \mathbb{N}^{m-1}$. Définissons $F_m = \exists x\ S_{\bar{\alpha}_m}(x) \wedge \neg S_{\bar{\beta}_m}(x)$. Remarquons que $\mathrm{qr}(F_m) \leqslant \lceil \log m \rceil + 2$. Montrons que F_m exprime $|\mathcal{I}| \geqslant m$. En effet si F_m est vraie, alors il existe x qui est somme de m éléments *différents* de $\mathcal{I}$: ces éléments doivent être différents puisque la seconde partie de F_m assure que x n'est pas combinaison linéaire de $m-1$ éléments de $\mathcal{I}$ avec pour coefficients $(2,1,\ldots,1)$. Réciproquement, si $|\mathcal{I}| \geqslant m$, prenons pour s la somme de m éléments différents de $\mathcal{I}$. Ce point satisfait la formule $S_{\bar{\alpha}_m}(s) \wedge \neg S_{\bar{\beta}_m}(s)$ puisque les éléments de $\mathcal{I}$ sont linéairement indépendants, et F_m est donc vraie. □

Proposition 8.2 $\mathrm{QR}_{\mathbb{Q}vs}(\mathrm{Card}_m, n) \leqslant \lceil \log m \rceil + 2$.

Démonstration. On se place dans $\mathbb{Q}$. On suppose $|\mathcal{I}| \leqslant n$. Remarquons que si la formule décrite dans la démonstration précédente est vraie, alors $|\mathcal{I}| \geqslant m$. Par contre, si elle n'est pas satisfaite, on ne peut pas conclure – on n'a plus l'hypothèse d'indépendance linéaire. Pour supprimer cette hypothèse, la méthode consiste à pondérer la somme dans la démonstration précédente par des coefficients entiers. Remarquons que $S_{\bar{\alpha}}(x)$ est équivalente à $S_{\bar{\alpha}'}(x)$ où $\bar{\alpha}'$ est obtenue à partir de $\bar{\alpha}$ en permutant les éléments. C'est pourquoi on ne considère que les uples croissants (au sens large). Pour un uple $\bar{\alpha} = (\alpha_1,\ldots,\alpha_p) \in \mathbb{N}^p$, définissons $s(\bar{\alpha})$ comme l'ensemble des uples croissants de $\mathbb{N}^{p-1}$ obtenus en remplaçant dans $\bar{\alpha}$ deux éléments quelconques α_i et α_j par leur somme. Par exemple, $s((1,4,7)) = \{(5,7),(4,8),(1,11)\}$ et $s((1,2,2,3)) = \{(2,3,3),(2,2,4),(1,3,4),(1,2,5)\}$. Définissons les formules

$$J_{\bar{\alpha}} = \exists x S_{\bar{\alpha}}(x) \wedge \bigwedge_{\bar{\beta} \in s(\bar{\alpha})} \neg S_{\bar{\beta}}(x).$$

Pour les raisons exposées ci-dessus, si $J_{(\alpha_1,\ldots,\alpha_p)}$ est vraie alors $|\mathcal{I}| \geqslant p$. On est prêt à construire une formule exprimant $|\mathcal{I}| \geqslant m$ sous l'hypothèse $|\mathcal{I}| \leqslant n$. Soit $N = m! n^{m-1}$.

Soit $\mathcal{A}$ un ensemble de $Nm+1$ éléments de $\mathbb{N}^m$ en position générale : cela signifie qu'aucun hyperplan de $\mathbb{R}^m$ ne contient plus de m éléments de $\mathcal{A}$. On affirme que la formule $H_m := \bigvee_{\bar{\alpha}\in\mathcal{A}} J_{\bar{\alpha}}$ est vraie si et seulement si $|\mathcal{I}| \geqslant m$.

Démonstration. Si H_m est vraie, alors pour un $\bar{\alpha} \in \mathcal{A}$ la formule $J_{\bar{\alpha}}$ est vraie et ceci implique $|\mathcal{I}| \geqslant m$. Pour la réciproque, supposons $|\mathcal{I}| \geqslant m$. Alors $\mathcal{I} = \{x_1, \ldots, x_l\}$ où les x_i sont tous différents, et $m \leqslant l \leqslant n$. Considérons toutes les équations

$$\sum_{i=1}^{m} A_i x_i = \sum_{i=1}^{m-2} A_{\sigma(i)} x_{t(i)} + (A_{\sigma(m-1)} + A_{\sigma(m)}) x_{t(m-1)}$$

où σ parcourt toutes les permutations de $\{1, \ldots, m\}$ et t parcourt toutes les applications de $\{1, \ldots, m-1\}$ dans $\{1, \ldots, l\}$. Ces équations définissent une famille $\mathcal{H}_{\bar{x}}$ d'hyperplans de $\mathbb{R}^m$ en $A_1, \ldots, A_m$ paramétrés par $(x_1, \ldots, x_l)$. Tout d'abord ce sont de *vrais* hyperplans. Remarquons aussi que $|\mathcal{H}_{\bar{x}}| \leqslant N$. Sur chaque hyperplan de $\mathcal{H}_{\bar{x}}$ il existe au plus m éléments de $\mathcal{A}$ puisqu'ils sont en position générale. Comme $|\mathcal{A}| > |\mathcal{H}_{\bar{x}}|m$, il doit y avoir au moins un $\bar{\alpha} \in \mathcal{A}$ qui n'est sur aucun hyperplan de $\mathcal{H}_{\bar{x}}$. Pour un tel $\bar{\alpha}$ la formule $J_{\bar{\alpha}}$ est vraie (prendre $x = \sum_{i=1}^{m} \alpha_i x_i$ pour le premier quantificateur existentiel). Ainsi la formule H_m est vraie. □

Théorème 8.2 $\mathrm{QR}_{\mathbb{Q}vs}(\mathrm{Card}_m, n) \leqslant \lceil \log m \rceil + 1$.

Démonstration. Supposons $|\mathcal{I}| \leqslant n$. Nous devons juste utiliser un quantificateur de moins que dans la méthode précédente. Étendons pour cela la définition de $S_{(\alpha_1,\ldots,\alpha_m)}(x)$ au cas où les α_i sont rationnels. On définit $S_{(1/q)}(x) := I(qx)$, où qx est $x + \ldots + x$ (q fois) – c'est ici qu'on gagne un quantificateur. On définit aussi $S_{(p/q)}(x) := \exists y\ I(qy) \wedge x = py$ si $p \neq 1$. Comme précédemment, on pose $S_{(\alpha_1,\ldots,\alpha_m)}(x) := \exists y\ S_{(\alpha_1,\ldots,\alpha_{\lfloor m/2 \rfloor})}(y) \wedge S_{(\alpha_{\lceil m/2 \rceil},\ldots,\alpha_m)}(x-y)$. Maintenant, soit $\mathcal{A}$ un ensemble de $Nm+1$ éléments de $(1/(\mathbb{N}\setminus\{0\}))^m$ en position générale. Pour $\bar{\alpha} = (\alpha_1, \ldots, \alpha_m) \in \mathcal{A}$, quel est le rang de quantification de $J_{\bar{\alpha}}$? Le rang de quantification de $S_{\bar{\alpha}}(x)$ est $\lceil \log m \rceil$. De plus, on affirme que pour tout $\bar{\beta}$, on peut permuter les β_i de sorte que $\mathrm{qr}(S_{\bar{\beta}}(x)) = \lceil \log m \rceil$. Remarquons que, dans un $\bar{\beta}$, tous les coefficients ont pour numérateur 1, sauf peut-être celui qui est de la forme $\alpha_i + \alpha_j$. Deux cas peuvent se produire. Si $m-1$ est une puissance de 2, alors $\mathrm{qr}(J_{\bar{\beta}}(x)) = \lceil \log(m-1) \rceil + 1$, ce qui est égal à $\lceil \log m \rceil$. Si $m-1$ n'est pas une puissance de 2, alors on peut permuter les β_i de sorte que $\mathrm{qr}(J_{\bar{\beta}}(x)) = \lceil \log(m-1) \rceil$: tout ce que nous avons à faire est de placer le coefficient p/q avec $p \neq 1$ *dans une partie de l'arbre qui n'atteint pas le niveau le plus bas.* Dans tous les cas, $\mathrm{qr}(S_{(\alpha_1,\ldots,\alpha_m)}) = \lceil \log m \rceil + 1$. □

Corollaire 8.1 $\mathrm{QR}_{\mathbb{Q}vs}(\mathrm{Pair}, n) = \mathrm{QR}_{ACF_0}(\mathrm{Pair}, n) = \lceil \log n \rceil + 1$.

Démonstration. Soit n fixé. Par le théorème 8.2, pour tout $m \leqslant n$, il existe une formule F_m exprimant $|\mathcal{I}| \geqslant m$, avec $\mathrm{qr}(F_m) = \lceil \log m \rceil + 1 \leqslant \lceil \log n \rceil + 1$. Bien sûr $F_m \wedge \neg F_{m+1}$ exprime que $|\mathcal{I}| = m$. Maintenant si on sait que $|\mathcal{I}| \leqslant n$, $|\mathcal{I}|$ est pair si et seulement si $\bigvee_{2k\leqslant n} |\mathcal{I}| = 2k$. Remarquons que $|\mathcal{I}| \geqslant n$ est équivalent à $|\mathcal{I}| = n$ puisque $|\mathcal{I}| \leqslant n$. Ainsi notre formule exprimant la parité est $\bigvee_{2k\leqslant n} F_{2k} \wedge \neg F_{2k+1}$ si n est impair, et $\bigvee_{2k<n} (F_{2k} \wedge \neg F_{2k+1}) \vee F_n$ si n est pair. Ceci permet d'obtenir la borne supérieure désirée. La borne inférieure est établie dans le théorème 8.1. □

Remarque 8.1 *Le théorème 8.1 est valable en caractéristique positive, la preuve est inchangée. Le résultat du théorème 8.2 est aussi valable pour un espace vectoriel. Tout espace vectoriel étant un espace vectoriel sur le sous-corps premier de son corps de base, il suffit de le montrer pour un espace vectoriel sur $\mathbb{F}_p$ (le corps à p éléments, p étant premier). La méthode de la proposition 8.2 ne s'applique plus, mais l'espace vectoriel engendré par un nombre fini de points est fini ce qui simplifie en fait la méthode. Par ailleurs, la borne inférieure sur les corps algébriquement clos implique qu'on a cette même borne inférieure pour la théorie des espaces vectoriels sur $\mathbb{F}_p$ de cardinal infini. On a donc montré que*

$$\mathrm{QR}(\mathrm{Pair}, n) = \lceil \log n \rceil + 1$$

sur tout espace vectoriel infini et tout corps algébriquement clos.

8.4 Parité en présence de l'ordre

On rappelle que $\mathcal{O}vs$ est la théorie des $\mathbb{Q}$-espaces vectoriels ordonnés. Montrons pour commencer une borne inférieure. On définit N_p de la manière suivante :

$$\begin{cases} N_0 & = & 1 \\ N_{p+1} & = & (2^p+1)N_p. \end{cases}$$

Définissons une mesure algébrique d_∞ de la manière suivante. Pour $x \leqslant y$, on définit $d_\infty(x,y) = |\{z,\ x < z \leqslant y\}|$. Alors, pour $j \in \mathbb{N}$, on définit $d_j(x,y) = d_\infty(x,y)$ si $d_\infty(x,y) < N_j$, $d_j(x,y) = \infty$ sinon. Enfin, on pose $d_\infty(y,x) = -d_\infty(x,y)$ et $d_j(y,x) = -d_j(x,y)$.

Lemme 8.2 *On considère une version modifiée du jeu de va-et-vient entre deux ensembles totalement ordonnés A et B où il est possible de choisir jusqu'à 2^j éléments à la fois, du même côté, quand il reste j coups à jouer. Si $|A|, |B| \geqslant N_{n+1}+1$, alors le second joueur a une stratégie gagnante pour le nouveau jeu de longueur n entre A et B.*

Démonstration. Soit $|A|, |B| \geqslant N_{n+1}+1$. On montre comment jouer un jeu de longueur n. Avant que le jeu ne commence, définissons un isomorphisme partiel α qui envoie les extrémités de A sur celles de B. Supposons qu'il reste j coups à jouer. On peut supposer que le premier joueur joue dans la structure A ; notons a_i les points choisis par ce joueur (au maximum 2^j éléments). Appelons $D \subset A$ le domaine de définition de α. Par hypothèse de récurrence, on suppose que $d_{j+1}(e,e') = d_{j+1}(\alpha(e), \alpha(e'))$ pour tous $e, e' \in D$. On procède comme dans le cas d'un va-et-vient entre deux ordres finis – voir [EF95, Imm98] – excepté que les joueurs peuvent choisir 2^j éléments. On va traiter d'un seul coup tous les a_i se trouvant dans un intervalle $]c,d[$ avec $c,d \in D$, $]c,d[\cap D = \emptyset$. Considérons un tel intervalle $]c,d[$ et notons sans perte de généralité $a_1 < a_2 < \ldots < a_k$ les a_i de cet intervalle. Premier cas : $d_{j+1}(c,d) < \infty$. Par récurrence, $d_{j+1}(c,d) = d_{j+1}(\alpha(c), \alpha(d))$ et on choisit les $\alpha(a_i)$ de manière évidente. Second cas : $d_{j+1}(c,d) = \infty$. Soit $a_0 = c$ et $a_{k+1} = d$. On choisit successivement $\alpha(a_l)$ pour $l = 1, 2, \ldots, s$ tels que $d_j(a_l, a_{l+1}) = d_j(\alpha(a_l), \alpha(a_{l+1}))$, où s est le plus petit indice vérifiant $d_j(a_s, a_{s+1}) = \infty$. On procède de la même manière pour $l = k, k-1, \ldots, t$ où t est le plus grand indice

tel que $d_j(a_{t-1}, a_t) = \infty$. Si les images de tous les a_i pour $1 \leqslant i \leqslant k$ n'ont pas été définies, alors on choisit successivement les images des a_l pour $l = s+1, \ldots, t-1$: on prend $\alpha(a_l)$ tel que $d_\infty(\alpha(a_{l-1}), \alpha(a_l)) = \min\{N_j, d_j(a_{l-1}, a_l)\}$. Montrons qu'on a assez de points de B dans $]\alpha(a_0), \alpha(a_{k+1})[$. Comme $d_{j+1}(\alpha(a_0), \alpha(a_{k+1})) = \infty$ par récurrence, on a $d_\infty(\alpha(a_0), \alpha(a_{k+1})) \geqslant N_{j+1}$. Prenant en compte que $k \leqslant 2^j$ et $N_{j+1} = (2^j+1)N_j$, il y a en effet assez de points pour procéder de cette manière. □

Théorème 8.3 $\mathrm{QR}_{\mathcal{O}vs}(\mathrm{Pair}, n) = \Omega(\sqrt{\log n})$.

Démonstration. On va faire un jeu de va-et-vient de longueur n entre deux $\mathbb{Q}$-espaces vectoriels ordonnés avec points $(V, \mathcal{M})$ et $(W, \mathcal{N})$. On choisit $\mathcal{M}$ (resp. $\mathcal{N}$) de sorte que c'est une base de V (resp. W). On note $a \ll b$ ou $a = o(b)$ si $\forall n \in \mathbb{N},\ n|a| \leqslant |b|$. En décomposant un point $v \in V$ dans la base $\mathcal{M}$, il s'écrit $v = \sum_{i=1}^{r} \alpha_i a_i$ avec $\alpha_i \in \mathbb{Q}^*$, $a_i \in \mathcal{M}$ et $a_1 \gg \ldots \gg a_r$. On utilise les notations suivantes : $\mathrm{supp}(v) = \{a_1, \ldots, a_r\}$, $\mathrm{supp}(v, l) = \{a_i,\ i \leqslant \min(l, r)\}$, $z(v, j) = a_{\min(2^j, r)}$ et $\mathrm{T}_j(v) = \sum_{i=1}^{\min(2^j, r)} \alpha_i a_i$. Ainsi $z(v, j) = z(\mathrm{T}_j(v), j)$. Remarquons que si $|\mathrm{supp}(\mathrm{T}_j(x))| < 2^j$ alors $x = \mathrm{T}_j(x)$. On note π la projection canonique de $\mathcal{M} \times \mathcal{N}$ sur $\mathcal{M}$. Étant donnée $R \subset \mathcal{M} \times \mathcal{N}$ une fonction injective d'une partie de $\mathcal{M}$ dans $\mathcal{N}$, on note $\mathcal{L}_R$ l'application linéaire définie sur $\mathrm{Vect}(\pi(R))$ et étendant R. On utilisera aussi des notations identiques dans $(W, \mathcal{N})$. Quand il reste j coups à jouer, on aura un isomorphisme φ_{j+1} défini de E_{j+1} sur F_{j+1}. On aura aussi besoin des distances d_∞ et d_j définies précédemment, mais relativisées à l'ensemble $\mathcal{M}$ (ou $\mathcal{N}$). Soit V le $\mathbb{Q}$-espace vectoriel ordonné engendré par $\mathcal{M} = \{\varepsilon_1, \ldots, \varepsilon_{n_v}\}$ avec $0 < \varepsilon_1 \ll \ldots \ll \varepsilon_{n_v}$ et $n_v = N_{n+1} + 1$ – cela revient au même que de considérer $\mathbb{Q}^{n_v}$ muni de l'ordre lexicographique. De même, $W = \mathrm{Vect}(\mathcal{N})$ avec $\mathcal{N} = \{\eta_1, \ldots, \eta_{n_w}\}$ tel que $0 < \eta_1 \ll \ldots \ll \eta_{n_w}$ et $n_w = N_{n+1} + 2$. On pose $E_{n+1} = \mathrm{Vect}(\{\varepsilon_1, \varepsilon_{n_v}\})$, $F_{n+1} = \mathrm{Vect}(\{\eta_1, \eta_{n_w}\})$ et φ_{n+1} est défini par $\varphi(\varepsilon_1) = \eta_1$ et $\varphi(\varepsilon_{n_v}) = \eta_{n_w}$. On pose enfin $R_{n+1} = \{(\varepsilon_1, \eta_1), (\varepsilon_{n_v}, \eta_{n_w})\} \subset \mathcal{M} \times \mathcal{N}$. À chaque étape, on va maintenir la propriété $\mathcal{P}_j$ suivante.

a) Pour tout $x, y \in \pi(R_j)$, $d_j(x, y) = d_j(R_j(x), R_j(y))$.
b) Pour tout $v \in E_j$, on a $\mathrm{T}_j(\varphi_j(v)) = \mathcal{L}_{R_j}(\mathrm{T}_j(v))$. De même, pour tous $w \in F_j$, $\mathrm{T}_j(\varphi_j^{-1}(w)) = \mathcal{L}_{R_j^{-1}}(\mathrm{T}_j(w))$.
c) L'application φ_j est un isomorphisme de $\mathbb{Q}$-espace vectoriel ordonné de E_j sur F_j tel que, pour tout $x \in E_j$, $x \in \mathcal{M}$ si et seulement si $\varphi_j(x) \in \mathcal{N}$.

Remarquons que le point b) signifie entre autres que $\mathcal{L}_{R_j}(\mathrm{T}_j(v))$ a un sens, et implique donc $\mathrm{supp}(\mathrm{T}_j(v)) \subset \pi(R_j)$. Remarquons aussi que, d'après a), R_j est une application strictement croissante de $\pi(R_j) \subset \mathcal{M}$ dans $\mathcal{N}$. Montrons que $\mathcal{P}_{n+1}$ est vérifiée : a) vient de $|\mathcal{M}|, |\mathcal{N}| \geqslant N_{n+1} + 1$, les autres points sont clairs. Supposons maintenant que $n - j$ pas de va-et-vient ont été traités. Il reste $j \geqslant 1$ pas à jouer. L'isomorphisme φ_{j+1} est défini de E_{j+1} sur F_{j+1}. Par symétrie, on peut supposer que le point v est choisi dans $(V, \mathcal{M})$. Sans perte de généralité, on suppose que $v \notin E_{j+1}$. Soit $u \in \mathrm{Vect}(E_{j+1} \cup \{v\}) \setminus E_{j+1}$ *tel que $z(u, j)$ soit minimal* pour l'ordre sur V. Soit $S = \mathrm{supp}(u, 2^j) \setminus \pi(R_{j+1})$. Par le lemme 8.2, on peut définir la relation R_j étendant R_{j+1} et vérifiant $\pi(R_j) = \pi(R_{j+1}) \cup S$.

Soit φ_j l'application linéaire étendant φ_{j+1} et telle que $\varphi_j(u) = \mathcal{L}_{R_j}(\mathrm{T}_j(u))$. Soit $E_j = \mathrm{Vect}(E_{j+1} \cup \{u\})$ et $F_j = \varphi_j(E_j)$. Dans la suite, φ_j sera noté φ, R_j sera noté R et $\mathcal{L}_{R_j}$ parfois noté $\mathcal{L}_j$. Montrons que la propriété $\mathcal{P}_j$ est vérifiée. Remarquons tout d'abord que φ est une application linéaire de E_j sur F_j. Montrons que φ est injective. Soit $w \in E_j$, $\varphi(w) = 0$. On écrit $w = \alpha u + e$ avec $e \in E_{j+1}$ et $\alpha \in \mathbb{Q}$. Si $\alpha = 0$, comme φ_{j+1} est injective, alors $e = 0$. Supposons maintenant $\alpha \neq 0$. Alors $\varphi(e) = \varphi_{j+1}(e) = -\alpha\varphi(u) = -\alpha\mathcal{L}_j\mathrm{T}_j(u)$. Par le point $\mathcal{P}_{j+1}$ $b)$ pour φ_{j+1}^{-1} on obtient $\mathrm{T}_{j+1}(e) = -\alpha\mathcal{L}_{R_{j+1}^{-1}}\mathrm{T}_{j+1}\mathcal{L}_j\mathrm{T}_j u$. Donc $\mathrm{T}_{j+1}(e) = -\alpha\mathcal{L}_{R_{j+1}^{-1}}\mathcal{L}_j\mathrm{T}_j u = -\alpha\mathrm{T}_j(u)$ parce que cette expression a un sens et R_j étend R_{j+1}. Mais $2^j < 2^{j+1}$, donc $e = -\alpha\mathrm{T}_j(u)$. Maintenant si $u \neq \mathrm{T}_j(u)$, cela donne $w = e + \alpha u \notin E_{j+1}$ avec $w = o(z(u, j))$ ce qui est impossible par choix de u. Comme $u = \mathrm{T}_j(u)$, on a $e = -\alpha u$, c'est-à-dire $w = 0$.

Le point a) découle de la construction. Montrons le point b) pour φ. Soit $v \in E_j$. Si $v \in E_{j+1}$, il est clair par $\mathcal{P}_{j+1}$ en tenant compte de $2^{j+1} \geqslant 2^j$, $R_{j+1} \subset R_j$ et du fait que φ étend φ_{j+1}. Supposons donc $v \notin E_{j+1}$. Alors $v = \alpha u + e$ où u est le vecteur choisi ci-dessus, $\alpha \in \mathbb{Q}^*$ et $e \in E_{j+1}$. L'équation suivante est satisfaite :

$$\mathrm{T}_j(v) = \mathrm{T}_j(\alpha\mathrm{T}_j(u) + \mathrm{T}_{j+1}(e)). \tag{8.1}$$

Démonstration. i) Supposons $z(\mathrm{T}_{j+1}(e)) \leqslant z(\mathrm{T}_j(u))$. Alors $\mathrm{T}_{j+1}(e) = e + o(z(\mathrm{T}_j(u)))$ et $v = \alpha u + e = \alpha\mathrm{T}_j(u) + \mathrm{T}_{j+1}(e) + o(z(\mathrm{T}_j(u)))$. Comme $z(\mathrm{T}_j(v)) \geqslant z(\mathrm{T}_j(u))$ par choix de u, on obtient la relation 8.1 en tronquant l'égalité précédente à l'ordre 2^j.
ii) Maintenant supposons $z(\mathrm{T}_{j+1}(e)) \geqslant z(\mathrm{T}_j(u))$.
ii-a) Si $\mathrm{T}_{j+1}(e) = e$, on a en particulier $\mathrm{T}_{j+1}(e) = e + o(z(\mathrm{T}_j(u)))$ et on finit comme au point précédent.
ii-b) Sinon, $|\mathrm{supp}(e, 2^{j+1})| = 2^{j+1}$. De plus $v = e + \alpha u = \mathrm{T}_{j+1}(e) + \alpha\mathrm{T}_j(u) + o(z(\mathrm{T}_{j+1}(e)))$. Comme la somme $\mathrm{T}_{j+1}(e) + \alpha\mathrm{T}_j(u)$ a au moins 2^j termes de $\mathrm{T}_{j+1}(e)$, on obtient l'équation 8.1 en tronquant l'égalité précédente à l'ordre 2^j. La propriété suivante est également vérifiée :

$$\mathrm{T}_j(\varphi(v)) = \mathrm{T}_j(\alpha\mathrm{T}_j(\varphi(u)) + \mathrm{T}_{j+1}(\varphi(e))). \tag{8.2}$$

Démonstration. i) Supposons $|\mathrm{supp}(e, 2^{j+1})| < 2^{j+1}$. Alors $e = \mathrm{T}_{j+1}(e)$. Par $\mathcal{P}_{j+1}$, on obtient $\mathrm{T}_{j+1}(\varphi(e)) = \mathcal{L}_{j+1}(\mathrm{T}_{j+1}(e)) = \mathcal{L}_{j+1}(e)$. Mais $\mathcal{L}_{j+1}(e)$ a strictement moins que 2^{j+1} termes donc $\mathrm{T}_{j+1}(\varphi(e)) = \varphi(e)$. Rappelons que $\varphi(u) = \mathrm{T}_j(\varphi(u))$ par choix de $\varphi(u)$. En substituant ces termes dans $\mathrm{T}_j(\varphi(v)) = \mathrm{T}_j(\alpha\varphi(u) + \varphi(e))$ on obtient l'équation 8.2.
ii) Sinon $|\mathrm{supp}(e, 2^{j+1})| = 2^{j+1}$. Alors $\mathrm{T}_{j+1}(\varphi(e)) = \mathcal{L}_{j+1}(\mathrm{T}_{j+1}(e))$ a 2^{j+1} termes. Mais $\varphi(v) = \alpha\varphi(u) + \varphi(e) = \alpha\mathrm{T}_j(\varphi(u)) + \mathrm{T}_{j+1}(\varphi(e)) + o(z(\mathrm{T}_{j+1}(\varphi(e))))$. Comme $\alpha\mathrm{T}_j(\varphi(u)) + \mathrm{T}_{j+1}(\varphi(e))$ a au moins 2^j termes de $\mathrm{T}_{j+1}(\varphi(e))$, on obtient la relation 8.2 en tronquant l'égalité précédente à l'ordre 2^j.

Montrons maintenant le point P_j $b)$ pour φ. Soit $v \in E_j$. On écrit $v = \alpha u + e$ avec $e \in E_{j+1}$ et $\alpha \in \mathbb{Q}$. Par l'équation 8.2, $\mathrm{T}_j(\varphi(v)) = \mathrm{T}_j(\alpha\mathrm{T}_j(\varphi(u)) + \mathrm{T}_{j+1}(\varphi(e)))$. Mais $\mathrm{T}_j(\varphi(u)) = \varphi(u) = \mathcal{L}_j\mathrm{T}_j(u)$ par choix de $\varphi(u)$. De plus, par $\mathcal{P}_{j+1}$, $\mathrm{T}_{j+1}(\varphi(e)) = \mathcal{L}_{j+1}\mathrm{T}_{j+1}(e)$. Et $\mathcal{L}_{j+1}\mathrm{T}_{j+1}(e) = \mathcal{L}_j\mathrm{T}_{j+1}(e)$ puisque $\mathcal{L}_j$ étend $\mathcal{L}_{j+1}$. Par linéarité de $\mathcal{L}_j$, cela donne $\mathrm{T}_j(\varphi(v)) = \mathrm{T}_j(\mathcal{L}_j(\alpha\mathrm{T}_j(u) + \mathrm{T}_{j+1}(e)))$. Clairement, si $\mathrm{T}_j\mathcal{L}_j(x)$ a un sens pour

$x \in E_j$, alors $\mathcal{L}_j \mathrm{T}_j(x) = \mathrm{T}_j \mathcal{L}_j(x)$. Ainsi $\mathrm{T}_j(\varphi(v)) = \mathcal{L}_j \mathrm{T}_j(\alpha \mathrm{T}_j(u) + \mathrm{T}_{j+1}(e))$. Avec la relation 8.1 on obtient $\mathrm{T}_j(\varphi(v)) = \mathcal{L}_j(\mathrm{T}_j(v))$.

On montre maintenant b) pour φ^{-1}. Soit $w \in F_j$ et $v \in E_j$ tels que $w = \varphi(v)$. On a $\mathrm{T}_j(w) = \mathrm{T}_j(\varphi(v)) = \mathcal{L}_j(\mathrm{T}_j(v))$ par $\mathcal{P}_j$ b) pour φ. De plus, $\mathcal{L}_{R_j}^{-1} = \mathcal{L}_{R_j^{-1}}$; donc $\mathrm{T}_j(\varphi^{-1}(w)) = \mathcal{L}_{R_j^{-1}}(\mathrm{T}_j(w))$. Cela montre le point b) de $\mathcal{P}_j$ pour φ^{-1}. Il reste à montrer c). Si $a \in E_j \cap \mathcal{M}$, alors par $\mathcal{P}_j$ b) on a $\mathrm{T}_j(\varphi(a)) = \mathcal{L}_j \mathrm{T}_j(a) = \mathcal{L}_j(a) = R(a)$. Mais $|\mathrm{supp}(R(a))| = 1 < 2^j$, donc $\varphi(a) = R(a) \in \mathcal{N}$. De même, si $x \in E_j$ est positif, alors $x = \alpha a + o(a)$ avec $a \in \mathcal{M}$ et $\alpha > 0$. Par le point b) de $\mathcal{P}_j$, on a $\varphi(a) = \alpha R_j(a) + o(R_j(a))$. Mais $R_j(a) \in \mathcal{N}$; ainsi $R_j(a) > 0$, ce qui montre $\varphi(x) > 0$. Le même raisonnement s'applique à φ^{-1} ce qui termine de montrer le point c). Ceci termine le jeu de va-et-vient. On a montré $\mathrm{QR}_{\mathcal{O}vs}(\mathrm{Pair}, N_{n+1} + 2) > n$. Comme $N_p = \prod_{i=0}^{p}(2^i + 1) \leqslant 2^{(p+1)(p+2)/2}$, on obtient $\mathrm{QR}_{\mathcal{O}vs}(\mathrm{Pair}, n) = \Omega(\sqrt{\log n})$. □

Un résultat similaire est-il vrai sur les corps réels clos ? Il existe une borne plus faible sur les structures o-minimales qui ont l'élimination des quantificateurs : voir proposition 8.7. Montrons maintenant une borne supérieure.

L'idée est la suivante. Pour exprimer que $|\mathcal{I} \cap]a, b[| \geqslant 2^{p^2}$, il est suffisant de trouver un ensemble S de 2^{2p} éléments de $\mathcal{I}$ tels qu'entre deux éléments consécutifs de S, on ait au moins $2^{(p-1)^2}$ éléments de $\mathcal{I}$ (ce qu'on exprime de manière récursive). L'ensemble S est représenté par la somme de ses éléments, de laquelle il est possible d'extraire les éléments à l'aide d'une formule de rang de quantification p. Cependant, si plusieurs ensembles d'éléments ont même somme, il n'y a pas d'éléments canoniques à extraire de cette somme, et les intervalles considérés dans l'étape de récurrence pourraient se chevaucher. C'est pourquoi on s'assure qu'aucun autre ensemble ne peut donner la même somme en pondérant les coefficients des éléments dans la somme.

Proposition 8.3 $\mathrm{QR}_{\mathcal{O}vs}(\mathrm{Card}_m, n) = O(\sqrt{\log m})$.

Démonstration. On travaille dans $\mathbb{Q}$. Soit n fixé. Dans la suite on suppose que $|\mathcal{I}| \leqslant n$. On définit la suite m_i par $m_0 = 1$ et $m_p = 2^p + (2^p + 1)m_{p-1}$. On définit également une famille de formules. Pour $k \geqslant 1$ et $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k) \in \mathbb{N}^k$, on construit une formule $S_{\bar{\alpha}}(a, b, x)$ de rang de quantification $O(\log k)$ qui est vraie si et seulement si

$$\exists x_1, \ldots, x_k \in \mathcal{I}\ a < x_1 < x_2 < \ldots < x_k < b \wedge x = \sum_{i=1}^{k} \alpha_i x_i.$$

On construit aussi $E_{(\alpha_1,\ldots,\alpha_k),j}(a, b, x, z)$ pour $1 \leqslant i \leqslant k$, également de rang de quantification $O(\log k)$, qui est vraie si et seulement si

$$\exists x_1, \ldots, x_k \in \mathcal{I}\ a < x_1 < x_2 < \ldots < x_k < b \wedge x = \sum_{i=1}^{k} \alpha_i x_i \wedge x_j = z.$$

On ne détaillera pas la construction de ces formules qui est évidente. On va montrer par récurrence sur p que pour tout $m \leqslant m_p$ il existe une formule $F_m(a, b)$ de rang de quantification $O(p)$ exprimant que $|\mathcal{I} \cap]a, b[| \geqslant m$. Ceci est clair pour $p = 0$. On veut

maintenant montrer la propriété pour p en supposant l'avoir montrée pour $p-1$. Soit $m_{p-1} < m \leqslant m_p$; alors $m = (m_{p-1}+1)q + r$ pour un $1 \leqslant q \leqslant 2^p$ et $r \leqslant m_{p-1}$. Soit $\bar{\alpha} = (\alpha_1, \ldots, \alpha_q) \in \mathbb{N}^q$. On définit

$$\begin{aligned} G^m_{\bar{\alpha}} \quad := \quad & \exists x S_{\bar{\alpha}}(a,b,x) \wedge \exists z E_{\bar{\alpha},1}(a,b,x,z) \wedge F_{m_{p-1}}(a,z) \wedge \\ & \bigwedge_{i=1}^{q-1} \exists z_1, z_2 E_{\bar{\alpha},i}(a,b,x,z_1) \wedge E_{\bar{\alpha},i+1}(a,b,x,z_2) \wedge F_{m_{p-1}}(z_1,z_2) \wedge \\ & \exists z E_{\bar{\alpha},q}(a,b,x,z) \wedge F_r(z,b). \end{aligned}$$

Soit $N = n^{2q}$ et $\mathcal{A}$ un ensemble de $Nq+1$ éléments de $\mathbb{N}^q$ en position générale. On affirme que

$$(\mathbb{Q}, \mathcal{I}) \models \bigwedge_{\bar{\alpha} \in \mathcal{A}} G^m_{\bar{\alpha}}(a,b) \iff |\mathcal{I} \cap]a,b[| \geqslant m.$$

De droite à gauche : supposons $|\mathcal{I} \cap]a,b[| \geqslant m$. Soit $x_1 < x_2 < \ldots < x_m$ des éléments de $\mathcal{I} \cap]a,b[$. Soit $\bar{\alpha}$ un élément de $\mathcal{A}$. Alors $G^m_{\bar{\alpha}}$ est vraie – prendre $x = \sum_{i=1}^q \alpha_{i(m_{p-1}+1)} x_{i(m_{p-1}+1)}$ pour le premier quantificateur, et $z = x_i$ quand on demande un élément z tel que $E_{\bar{\alpha},i}(a,b,x,z)$. Ainsi $\bigwedge_{\bar{\alpha}\in\mathcal{A}} G^m_{\bar{\alpha}}$ est vraie.

Réciproquement, supposons que $\bigwedge_{\bar{\alpha}\in\mathcal{A}} G^m_{\bar{\alpha}}$ est vraie. Notons $x_1 < x_2 < \ldots < x_l$ les éléments de $\mathcal{I} \cap]a,b[$, où $0 \leqslant l \leqslant n$ par hypothèse. Définissons une famille d'hyperplans $\mathcal{H}_{\bar{x}}$ paramétrée par $(x_1, \ldots, x_l)$. La famille $\mathcal{H}_{\bar{x}}$ se compose de tous les hyperplans de la forme

$$\sum_{i=1}^q x_{f(i)} A_i = \sum_{i=1}^q x_{g(i)} A_i$$

où f et g parcourent toutes les paires d'applications strictement croissantes de $\{1, \ldots, q\}$ dans $\{1, \ldots, l\}$ avec $f \neq g$. Comme $|\mathcal{H}_{\bar{x}}| \leqslant (l^q)^2 \leqslant N$, on a $|\mathcal{A}| > q|\mathcal{H}_{\bar{x}}|$, donc il doit exister $\bar{\alpha} \in \mathcal{A}$ ne se trouvant sur aucun hyperplan de $\mathcal{H}_{\bar{x}}$. Comme $G^m_{\bar{\alpha}}(a,b)$ est vraie, cela implique que $|\mathcal{I} \cap]a,b[| \geqslant m$. Il existe en effet des éléments $z_1 < \ldots < z_q$ de $\mathcal{I} \cap]a,b[$ tels que $x = \sum_{i=1}^q \alpha_i z_i$ "conviennent" pour le premier quantificateur de $G^m_{\bar{\alpha}}(a,b)$. De plus, toute autre suite $z'_1 < \ldots < z'_q$ vérifie $\sum_{i=1}^q \alpha_i z'_i \neq x$, parce que $\bar{\alpha}$ n'est pas sur $\mathcal{H}_{\bar{x}}$. Cela signifie que, pour tout $1 \leqslant i \leqslant q$, le seul z tel que $E_{\bar{\alpha},i}(a,b,x,z)$ est z_i. Par hypothèse de récurrence, on en déduit que $|\mathcal{I} \cap]z_i, z_{i+1}[| \geqslant m_{p-1}$ pour $1 \leqslant q-1$, $|\mathcal{I} \cap]a, z_1[| \geqslant m_{p-1}$ et enfin $|\mathcal{I} \cap]z_q, b[| \geqslant r$. Ainsi $|\mathcal{I} \cap]a,b[| \geqslant m$. □

Corollaire 8.2 $\mathrm{QR}_{\mathcal{O}vs}(\mathrm{Pair}, n) = \Theta(\sqrt{\log n})$.

Proposition 8.4 *Dans un $\mathbb{Q}$-espace vectoriel ordonné, on peut exprimer que $|\mathcal{I}| \geqslant m$ avec une formule de rang de quantification $O(\sqrt{\log m})$.*

Démonstration. En comparaison à la proposition précédente, on ne dispose plus de borne sur $|\mathcal{I}|$. Cependant on affirme qu'une formule, construite comme dans la proposition précédente, exprimant $|\mathcal{I}| \geqslant m$ sous l'hypothèse $|\mathcal{I}| \leqslant m-1$, est valable même quand on supprime l'hypothèse que $|\mathcal{I}| \leqslant m-1$. En effet, la seule façon pour cette formule de se tromper est de répondre que $|\mathcal{I}| \geqslant m$ alors que ce n'est pas le cas. Mais cela ne peut arriver que si $|\mathcal{I}| \geqslant m$ puisque cette formule répond correctement quand $|\mathcal{I}| \leqslant m-1$. Cette formule se se trompe donc jamais. □

Remarque 8.2 *Peut-on également supprimer l'hypothèse concernant le cardinal de $\mathcal{I}$ dans la proposition 8.2 ?*

8.5 Connexité d'un graphe

Dans cette section, on considère un graphe fini G plongé dans une structure infinie M. On ajoute donc deux prédicats à la signature de M : un prédicat unaire V qui interprète l'ensemble des sommets $\mathcal{V}$ de graphe, et un prédicat binaire E pour les arêtes. On note $d(\cdot,\cdot)$ la distance usuelle sur les sommets du graphe. On s'intéresse à la connexité et à l'atteignabilité. La requête Connexe demande si le graphe G est connexe. La requête Atteint_m à deux variable libres a et b est vraie si $a, b \in \mathcal{V}$ et $d(a,b) \leqslant m$. La requête Atteint est définie de la même manière, excepté qu'il n'y a plus de borne sur la longueur du chemin reliant a et b. Une fois encore on considère des restrictions de ces requêtes au cas où on connaît une borne sur $|\mathcal{V}|$. Ainsi $\text{QR}_M(\text{Connexe}, n)$ est le rang de quantification minimal d'une formule exprimant qu'un graphe plongé dans M est connexe, sachant qu'il a au plus n sommets. Bien sûr le résultat du lemme 8.1 s'applique aussi à ces requêtes : si deux structures M et M' sont élémentairement équivalentes, alors $\text{QR}_M(\text{Connexe}, n) = \text{QR}_{M'}(\text{Connexe}, n)$ et $\text{QR}_M(\text{Atteint}_m, n) = \text{QR}_{M'}(\text{Atteint}_m, n)$. Remarquons que $\text{QR}_M(\text{Atteint}, n) = \text{QR}_M(\text{Atteint}_{n-1}, n)$. Remarquons aussi que $\forall a, b\ V(a) \wedge V(b) \rightarrow \text{Atteint}(a,b)$ exprime la connexité, donc $\text{QR}_M(\text{Connexe}, n) \leqslant \text{QR}_M(\text{Atteint}, n) + 2$. On montre un résultat similaire au théorème 8.3 pour un graphe fini plongé dans un $\mathbb{Q}$-espace vectoriel ordonné.

Corollaire 8.3 $\text{QR}_{\mathcal{O}vs}(\text{Connexe}, n) = \Omega(\sqrt{\log n})$.

Démonstration. On utilise une réduction du premier ordre classique de la parité à la connexité. Soit $\mathcal{I}$ un ensemble composé des éléments $v_1 < v_2 < \ldots < v_n$. On considère le graphe $G_n = (V, E)$ sur $V = \{v_1, \ldots, v_n\}$ où $E(v_i, v_j)$ est vrai si et seulement si $|i-j| = 2$ ou $\{i,j\} = \{1,n\}$. Pour $n \geqslant 2$, G_n est connexe si et seulement $n = |\mathcal{I}|$ est pair. Comme on peut exprimer E avec une formule du premier ordre de rang de quantification 2 (à l'aide du prédicat I interprétant $\mathcal{I}$) dans toute structure ordonnée M, on obtient $\text{QR}_M(\text{Pair}, n) \leqslant \text{QR}_M(\text{Connexe}, n) + 2$. Il reste à appliquer ceci à la théorie $\mathcal{O}vs$. □

En utilisant des techniques similaires aux précédentes, on peut établir une borne inférieure sur les corps algébriquement clos. Par la remarque faite au début de cette section, il est suffisant de montrer le résultat sur $\mathbb{C}$. On reprend la définition de la section 8.4 de N_i pour $i \geqslant 0$. Appelons C_n le cycle de longueur N_{n+1}. Soit G_n le graphe C_n et H_n le graphe composé de deux copies disjointes de C_n. Comme dans la section 8.4 on définit d_j comme étant la troncature de la distance d dans le graphe à l'ordre N_j. Commençons par établir un analogue du lemme 8.2.

Lemme 8.3 *Considérons la variante du jeu de va-et-vient où il est possible de choisir jusqu'à 2^{j-1} éléments en une fois, du même côté, quand il reste j coups à jouer. Alors le second joueur a une stratégie pour gagner ce jeu modifié de longueur n entre G_n et H_n.*

Démonstration. Soit V_G l'ensemble des sommets de G_n, et V_H l'ensemble des sommets de H_n. Quand il reste j coups à jouer, on a un isomorphisme partiel φ_j défini sur $A_j \subset V_G$ sur $B_j \subset V_H$. Pour montrer le lemme, il suffit de maintenir à chaque étape la propriété suivante :

$$\forall x, y \in A_j, d_j(x,y) = d_j(\varphi_j(x), \varphi_j(y)).$$

La méthode est similaire à celle du lemme 8.2. □

Proposition 8.5 $\mathrm{QR}_{ACF_0}(\text{Connexe}, n) = \Omega(\sqrt{\log n})$.

Démonstration. Par la remarque faite au début de la section, il est suffisant de montrer le résultat sur $\mathbb{C}$. Considérons $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ un plongement de G_n dans $\mathbb{C}$ tel que tous les sommets $V_\mathcal{G}$ soient algébriquement indépendants sur $\mathbb{Q}$. De même, soit $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$ un plongement de H_n dans $\mathbb{C}$ tels que les sommets $V_\mathcal{H}$ soient algébriquement indépendants sur $\mathbb{Q}$. Il suffit de montrer que le second joueur a une stratégie pour gagner le jeu de longueur n entre $(\mathbb{C}, \mathcal{G})$ et $(\mathbb{C}, \mathcal{H})$. La conclusion viendra alors du fait 8.1. Le va-et-vient est mené comme dans le théorème 8.1. Quand il reste j coups à jouer, on note E_j le corps où φ est défini et $F_j = \varphi(E_j)$. Soit $E_n = F_n = \overline{\mathbb{Q}}$ et $\varphi = \mathrm{Id}_{\overline{\mathbb{Q}}}$. À chaque étape on maintient la propriété suivante $\mathcal{P}_j$.

a) L'application φ_j est un isomorphisme de corps algébriquement clos de caractéristique nulle de E_j sur F_j.
b) S'il existe $a \in V_\mathcal{G} \setminus E_j$ tel que $a \in \overline{E_j \cup A}$ et $A \subset V_\mathcal{G} \setminus (E_j \cup \{a\})$, alors $|A| \geqslant 2^{j-1}$. Et la propriété correspondante dans $(\mathbb{C}, \mathcal{H})$.
c) $\forall x \in E_j, x \in V_\mathcal{G}$ ssi $\varphi_j(x) \in V_\mathcal{H}$.
d) $\forall x, y \in E_j, d_j(x,y) = d_j(\varphi_j(x), \varphi_j(y))$.

Remarquons qu'une conséquence de d) est que $\forall x, y \in E_j$, $(x,y) \in E_\mathcal{G}$ si et seulement si $(\varphi_j(x), \varphi_j(y)) \in E_\mathcal{H}$. Supposons que $n - j - 1$ coups aient été joués. L'isomorphisme φ est défini sur E_{j+1} et il reste $j+1$ coups à jouer. La propriété $\mathcal{P}_{j+1}$ est vérifiée par hypothèse de récurrence. Par symétrie, on peut supposer que le point choisi se trouve dans $(\mathbb{C}, \mathcal{G})$. Notons v ce point. On peut également supposer $v \notin E_{j+1}$. Deux cas se présentent :
• Premier cas : $v \in \overline{E_{j+1} \cup \{a_1, \ldots, a_r\}}$ avec $a_i \in V_\mathcal{G} \setminus E_{j+1}$ distincts et $r \leqslant 2^{j-1}$. Alors le second joueur choisit des éléments $b_1, \ldots, b_r$ dans $V_\mathcal{H} \setminus F_{j+1}$ *comme suggéré dans le lemme 8.3*, et on définit $\varphi(a_i) = b_i$. Que cela signifie-t-il exactement ? À partir du coup v joué par le premier joueur dans le jeu entre $(\mathbb{C}, \mathcal{G})$ et $(\mathbb{C}, \mathcal{H})$, le second joueur détermine des éléments $a_1, \ldots, a_r$ qui correspondent à un coup de la version modifiée du jeu de va-et-vient définie au lemme 8.3. Il choisit alors des points $b_1, \ldots, b_r$ en accord avec la stratégie gagnante définie dans le lemme 8.3, et joue le coup correspondant dans le jeu entre $(\mathbb{C}, \mathcal{G})$ et $(\mathbb{C}, \mathcal{H})$.
• Second cas : soit $f \notin \overline{F_{j+1} \cup V_\mathcal{H}}$. Soit $\varphi(v) = f$. On pose $E_j = \overline{E_{j+1} \cup \{v\}}$.
Les propriétés a), b) et c) sont maintenues puisque le jeu se déroule comme dans le théorème 8.1. La propriété d) est vérifiée puisque les points sont choisis comme décrit dans le lemme 8.3. □

On établit maintenant une borne supérieure pour l'atteignabilité dans les $\mathbb{Q}$-espaces vectoriels – ce qui montre la même borne concernant les théories $\mathcal{O}vs$ et ACF_0.

Proposition 8.6 $\mathrm{QR}_{\mathbb{Q}vs}(\mathrm{Atteint}_m, n) = O(\sqrt{\log m})$.

Démonstration. Soit n fixé. On définit la suite m_i par $m_0 = 1$ et $m_{p+1} = (2^p+1)m_p$. On montre par récurrence sur p que, pour tout $m_p < m \leqslant m_{p+1}$, il existe une formule des $\mathbb{Q}$-espaces vectoriels de rang de quantification $O(p)$ exprimant "$d(\cdot,\cdot) \leqslant m$" si $|\mathcal{V}| \leqslant n$. Soit $m = m_p q + r$, avec $1 \leqslant q \leqslant 2^p+1$ et $0 \leqslant r < m_p$. Maintenant, pour deux sommets u et v, $d(u,v) \leqslant m$ si et seulement s'il existe des sommets $c_1, c_2, \ldots, c_q$ tels que $d(u,c_1) \leqslant m_p$, $d(c_i, c_{i+1}) \leqslant m_p$ pour tout i, et $d(c_q, v) \leqslant r$. La suite $(c_1, c_2, \ldots, c_q)$, composée d'au plus 2^p+1 éléments, est représentée par une somme pondérée de laquelle il est possible d'extraire sans ambiguïté les c_i par une formule de rang de quantification $O(p)$ par la technique de la proposition 8.2. Les formules exprimant "$d(c_i, c_{i+1}) \leqslant \cdot$" sont obtenues de manière récursive. □

Corollaire 8.4 *Pour* $T \in \{\mathbb{Q}vs, ACF_0, \mathcal{O}vs\}$,
$\left|\begin{array}{l} \mathrm{QR}_T(\mathrm{Atteint}_m, n) = \Theta(\sqrt{\log m}) \\ \mathrm{QR}_T(\mathrm{Atteint}, n) = \Theta(\sqrt{\log n}) \\ \mathrm{QR}_T(\mathrm{Connexe}, n) = \Theta(\sqrt{\log n}). \end{array}\right.$

8.6 Relations avec l'effondrement actif-naturel

Sur une structure M présentant l'effondrement actif-naturel, il existe une relation entre la croissance du rang de quantification lorsqu'on transforme une formule en sémantique naturelle en une formule équivalente en sémantique active, et le rang de quantification nécessaire pour exprimer la parité. Ceci est détaillé dans les deux remarques qui suivent.

Proposition 8.7 *Soit M une structure o-minimale admettant l'élimination des quantificateurs. Alors* $\mathrm{QR}_M(\mathrm{Pair}, n) \geqslant \log\log n + O(1)$.

Démonstration. Soit ψ une formule exprimant la parité de $|\mathcal{I}|$ pour $|\mathcal{I}| \leqslant n$. On considère une structure M', élémentairement équivalente à M et contenant une suite indiscernable $E = \{e_i,\ i \in \mathbb{N}\}$. La même formule ψ exprime encore la parité (jusqu'à n) sur M'. Soit ψ' la formule obtenue en remplaçant $I(t)$ par $\forall z\ (z = t \to I(z))$ dans ψ, où z est une nouvelle variable. Remarquons que $\mathrm{qr}(\psi') \leqslant \mathrm{qr}(\psi) + 1$. On peut maintenant appliquer à ψ' l'algorithme de [BL00] pour obtenir une formule équivalente en sémantique active ψ_{act} (comme mentionné dans l'article, il n'est pas nécessaire que ψ soit sous forme prénexe pour appliquer cet algorithme). On peut vérifier que $\mathrm{qr}(\psi_{act}) \leqslant 2^{qr(\psi')+O(1)}$. Maintenant quand on se restreint au cas où $\mathcal{I} \subset E$, la formule ψ_{act} est équivalente à une formule du premier ordre pur ψ_o avec $\mathrm{qr}(\psi_o) = \mathrm{qr}(\psi_{act})$. En appliquant la borne relative au premier ordre (en présence d'un ordre total uniquement) rappelée en section 8.1, on obtient $\mathrm{qr}(\psi_o) \geqslant \log n + O(1)$. Ainsi $\mathrm{qr}(\psi) \geqslant \log\log n + O(1)$. □

Question 8.1 *On a montré* $\Omega(\log\log n) \leqslant \mathrm{QR}_{\mathbb{R}}(\mathrm{Pair}, n) \leqslant O(\sqrt{\log n})$, *où* $\mathbb{R}$ *signifie* $(\mathbb{R}, +, -, \times, <)$. *Est-il possible d'établir un jeu de va-et-vient sur les corps réels clos pour améliorer cette borne inférieure ?*

Proposition 8.8 *On ne peut éviter une augmentation exponentielle du rang de quantification quand on transforme une formule naturelle sur les $\mathbb{Q}$-espaces vectoriels en une formule active équivalente. Le même résultat est valable pour les corps algébriquement clos.*

Démonstration. Pour une formule du premier ordre ϕ sur $\mathbb{Q}vs$ étendu avec un prédicat unaire I, notons $a(\phi)$ le rang de quantification minimum d'une formule équivalente en sémantique active. Soit $\alpha(r) = \max\{a(\phi),\ \mathrm{qr}(\phi) = r\}$. On veut montrer que $\alpha(p) \geqslant 2^p$. Considérons la formule ϕ_n exprimant que $\mathcal{I}$ a au moins n éléments – voir théorème 8.2. Soit ϕ_n^a une formule équivalente en sémantique active. Quand on se restreint au cas où $\mathcal{I} \subset E$ avec $E = \{e_i,\ i \in \mathbb{N}\}$ un ensemble indiscernable, on obtient $\tilde{\phi}_n^a$ une formule du premier ordre pur exprimant que $\mathcal{I}$ a au moins n éléments. Comme $\mathrm{qr}(\phi_n^a) = \mathrm{qr}(\tilde{\phi}_n^a) \geqslant n$ et $\mathrm{qr}(\phi_n) = \lceil \log n \rceil$, prendre $n = 2^p$ donne le résultat. Cette preuve se traduit directement au cas des corps algébriquement clos (de caractéristique quelconque puisqu'on a uniquement besoin d'une borne inférieure). □

9. Effondrement actif-naturel

On montre dans ce chapitre plusieurs résultats d'effondrement actif-naturel. Ces résultats ne sont pas nouveaux, ils se déduisent du théorème 7.6 de Flum et Ziegler. La méthode est cependant radicalement différente. Alors que le théorème précédent repose sur une propriété de va-et-vient, la méthode exposée dans ce chapitre repose sur une procédure de localisation directement inspirée de l'algorithme de Benedikt et Libkin pour les structures o-minimales éliminant les quantificateurs [BL00]. Cette méthode est à la fois plus élémentaire et plus algorithmique. On s'intéresse ensuite à l'élimination uniforme des quantificateurs, notion intimement liée à celle d'effondrement actif-naturel.

9.1 Condition suffisante pour l'effondrement actif-naturel

On considère une signature relationnelle $\mathcal{B} = \{R_1, R_2, \ldots, R_s\}$ et une L-structure infinie S de domaine M. De plus, on traite ici le cas fini où chaque relation interprète un ensemble fini. On note D le domaine actif, c'est-à-dire l'ensemble des coordonnées des points de la base de donnée. On rappelle que les formules actives n'utilisent que des quantificateurs restreints, de type $\exists x \in D$ et $\forall x \in D$, que nous écrirons $\exists^a x$ et $\forall^a x$.

Pour prouver l'effondrement actif-naturel, il est suffisant de supprimer un quantificateur existentiel devant une formule active avec paramètres. Considérons la formule $\varphi(\bar{x}) := \exists z\ \alpha(\bar{x}, z)$ où $\alpha(\bar{x}, z)$ est une formule active. On peut supposer sans perte de généralité que $\alpha(\bar{x}, z)$ est sous forme prénexe. Ainsi

$$\alpha(\bar{x}, z) = Q_1^a y_1 \ldots Q_m^a y_m \beta(\bar{x}, \bar{y}, z)$$

où $Q_i \in \{\exists, \forall\}$. On peut de plus supposer que

- toute sous-formule atomique de $\alpha(\bar{x}, z)$ est soit une formule de S (c'est-à-dire sur L), soit une formule de $\mathcal{B}$,
- $m > 0$ et $\alpha(\bar{x}, z)$ a au moins une sous-formule atomique de S,
- z n'apparaît dans aucune sous-formule atomique de $\mathcal{B}$.

Ainsi toute sous-formule atomique $\alpha(\bar{x}, z)$ de S est de la forme $\tau(\bar{x}, \bar{y}, z)$ avec $\bar{y} = (y_1, \ldots, y_m)$.

Définition 9.1 *Le domaine actif $D \subset M$, les paramètres $\bar{x}$ et un ensemble $\mathcal{T} = \{\tau_1(\bar{x}, \bar{y}, z), \ldots, \tau_k(\bar{x}, \bar{y}, z)\}$ de formules atomiques étant fixés, on appelle vecteur signe une application de $\mathcal{T} \times D^m$ dans $\{\text{vrai}, \text{faux}\}$. On appelle vecteur signe d'un point $u \in M$ l'application $(\tau, \bar{a}) \mapsto \tau(\bar{x}, \bar{a}, u)$.*

Proposition 9.1 (Condition suffisante pour l'effondrement actif-naturel) *Soit S une structure de domaine M éliminant les quantificateurs et possédant les propriétés suivantes. Pour toute famille finie $\mathcal{T} = \{\tau_1(\bar{x}, \bar{y}, z), \ldots\}$ de formules atomiques de S, il existe $\mathcal{B} \in \mathbb{N}$, un ensemble fini Γ, une famille de formules actives $\mathcal{S}_\gamma[\tau]$ de S pour $(\gamma, \tau) \in \Gamma \times \mathcal{T}$ et un ensemble de formules actives $\mathcal{F}_\gamma$ de S pour $\gamma \in \Gamma$ tels que pour tout $\bar{x} \in M^n$, pour tout domaine actif fini $D \subset M$ et pour tout vecteur signe $\vec{v}$, il existe $z \in M$ de vecteur signe $\vec{v}$ si et seulement s'il existe $(\gamma, \bar{t}) \in \Gamma \times D^{\mathcal{B}}$ tel que*
i) $\vec{v} = (\tau, \bar{a}) \mapsto \mathcal{S}_\gamma[\tau](\bar{x}, \bar{a}, \bar{t})$
ii) $(S, D) \models \mathcal{F}_\gamma(\bar{x}, \bar{t})$.
Alors S a l'effondrement actif-naturel.

Démonstration. On prend les notations du début de la section. Soit $\mathcal{T}$ un ensemble de formules atomiques de S apparaissant dans $\alpha(\bar{x}, z)$. Soit $\alpha_\gamma(\bar{x}, \bar{t})$ une formule $\alpha(\bar{x}, z)$ où $\tau(\bar{x}, \bar{y}, z)$ est remplacé par $\mathcal{S}_\gamma[\tau](\bar{x}, \bar{y}, \bar{t})$. Posons

$$\varphi_{act}(\bar{x}) := \exists^a \bar{t} \in D^{\mathcal{B}} \bigvee_{\gamma \in \Gamma} \mathcal{F}_\gamma(\bar{x}, \bar{t}) \wedge \alpha_\gamma(\bar{x}, \bar{t}).$$

Cette formule est équivalente à $\varphi(\bar{x})$ quand la base de données est non vide. On peut aussi traiter le cas de la base de données vide exactement comme dans [BL00]. Soit $\varphi_{nat}(\bar{x})$ une formule naturelle équivalente à $\varphi(\bar{x})$. Soit $\varphi_\emptyset(\bar{x})$ une formule $\varphi_{nat}(\bar{x})$ où toute sous-formule du type $R(\ldots)$ pour R dans la base de données a été remplacée par *faux*. Soit $\varphi'_\emptyset(\bar{x})$ une formule sans quantificateurs équivalente à $\varphi_\emptyset(\bar{x})$. Pour la formule active on prend $(\exists^a x\ x = x \wedge \varphi_{act}(\bar{x})) \vee (\neg \exists^a x\ x = x \wedge \varphi'_\emptyset(\bar{x}))$. □

Remarque 9.1 *On notera la similitude de la méthode de localisation précédente avec celle des théorèmes de transfert du chapitre 4.*

9.2 Structures fortement minimales

Pour une formule $\phi(\bar{a}, x)$, on note $\phi(\bar{a}, M)$ l'ensemble $\{x \in M,\ S \models \phi(\bar{a}, x)\}$. On rappelle qu'une structure S de domaine M est fortement minimale si pour toute formule $\phi(\bar{y}, x)$, il existe $d \in \mathbb{N}$ tel que pour tout $\bar{a} \in M$, l'un des ensembles $\phi(\bar{a}, M)$ et $\neg\phi(\bar{a}, M)$ contient au plus d points.

Proposition 9.2 *Les structures fortement minimales qui ont l'élimination des quantificateurs (par exemple les corps algébriquement clos) ont l'effondrement actif-naturel.*

Démonstration. Soit S une structure fortement minimale de domaine M qui élimine les quantificateurs. On va utiliser la proposition 9.1. Soit Θ un ensemble fini de formules atomiques de S de la forme $\theta(\bar{x}, \bar{y}, z)$. Soit Ψ l'ensemble composé des éléments de Θ et de leurs négations. Comme S est fortement minimale et Ψ est finie, il existe un entier d de sorte que pour tout $\psi \in \Psi$ et tout $\bar{a}, \bar{b}$ on a

$$\min\{|\psi(\bar{a}, \bar{b}, M)|, |M \setminus \psi(\bar{a}, \bar{b}, M)|\} \leqslant d.$$

Dans la suite, $\bar{x}$ est fixé. Soit $z \in M$. Pour chaque $(\psi, \bar{y}) \in \Psi \times D^m$, la formule $\psi(\bar{x}, \bar{y}, \cdot)$ partitionne M en un ensemble fini d'au plus d points et son complément. Deux cas se présentent alors : soit z se trouve du côté fini pour au moins un élément de $\Psi \times D^m$, soit il est toujours du côté infini. On va construire des *schémas* de vecteurs signe pour les deux cas.

Premier cas : il existe $(\psi_0, \bar{y}_0) \in \Psi \times D^m$ tel que $S \models \psi_0(\bar{x}, \bar{y}_0, M)$ est fini et $\psi_0(\bar{x}, \bar{y}_0, z)$. Dans ce cas le vecteur signe de z est complètement déterminé par une formule du type

$$\mathrm{loc}_{\bar{\psi}}(\bar{x}, \bar{t}, u) := \bigwedge_{i=1}^{d} \psi_i(\bar{x}, \bar{t}_i, u)$$

pour un $(\psi_i, \bar{t}_i) \in \Psi \times D^m$. En effet l'ensemble des points de M satisfaisant $\psi_0(\bar{x}, \bar{y}_0, \cdot)$ est fini, non vide, et contient au plus d points. On pose $E(u) := \psi_0(\bar{x}, \bar{y}_0, u)$. Ensuite on teste tour à tour pour chaque $(\psi, \bar{y}) \in \Psi \times D^m$ si $\psi(\bar{x}, \bar{y}, z)$. Pour chaque couple, ceci donne une nouvelle formule $E'(u) := E(u) \wedge \psi(\bar{x}, \bar{y}, u)$. Si $E'(M) \subsetneq E(M)$, alors on continue avec $E = E'$. Sinon, on laisse E inchangé. À chaque fois qu'on ajoute une formule atomique à E, cela diminue $|E(M)|$ d'au moins un. Cela montre que E est composé d'au plus d formules atomiques à la fin de la procédure. On est maintenant prêt à définir le premier ensemble de formules de vecteurs signe. Soit $\mathcal{B} = md$. Pour $\bar{t} \in D^{\mathcal{B}}$, on note $\bar{t} = \bar{t}_1.\bar{t}_2 \ldots \bar{t}_d$ avec $|\bar{t}_i| = m$. Pour $\bar{\psi} \in \Psi^d$ et $\bar{t} \in D^{\mathcal{B}}$, soit $\mathcal{S}_{\bar{\psi}}[\tau](\bar{x}, \bar{y}, \bar{t})$ une formule sans quantificateurs équivalente à

$$\forall u \in M \ \mathrm{loc}_{\bar{\psi}}(\bar{x}, \bar{t}, u) \to \tau(\bar{x}, \bar{y}, u).$$

De plus, la formule $\mathcal{F}_{\bar{\psi}}$ va exprimer que l'ensemble des points satisfaisant $\mathrm{loc}_{\bar{\psi}}(\bar{x}, \bar{t}, \cdot)$ est non vide et que tous ces points ont le même vecteur signe. Soit $\mathrm{NonVide}_{\bar{\psi}}(\bar{x}, \bar{t})$ une formule sans quantificateurs équivalente à $\exists u \ \mathrm{loc}_{\bar{\psi}}(\bar{x}, \bar{t}, u)$. Soit $\mathrm{meme}(\bar{x}, \bar{r}, u, v)$ la formule $\bigwedge_{\psi \in \Psi} \psi(\bar{x}, \bar{r}, u) \leftrightarrow \psi(\bar{x}, \bar{r}, v)$ et $\theta_{\bar{\psi}}(\bar{x}, \bar{t}, \bar{r})$ une formule sans quantificateurs équivalente à

$$\forall u, v \ (\mathrm{loc}_{\bar{\psi}}(\bar{x}, \bar{t}, u) \wedge \mathrm{loc}_{\bar{\psi}}(\bar{x}, \bar{t}, v)) \to \mathrm{meme}(\bar{x}, \bar{r}, u, v).$$

On définit $\mathrm{LocPrecise}_{\bar{\psi}}(\bar{x}, \bar{t}) := \forall^a \bar{r} \ \theta_{\bar{\psi}}(\bar{x}, \bar{t}, \bar{r})$. Enfin on pose

$$\mathcal{F}_{\bar{\psi}}(\bar{x}, \bar{t}) := \mathrm{NonVide}_{\bar{\psi}}(\bar{x}, \bar{t}) \wedge \mathrm{LocPrecise}_{\bar{\psi}}(\bar{x}, \bar{t}).$$

Second cas : pour tout $(\psi, \bar{y}) \in \Psi \times D^m$, z est dans l'ensemble infini entre $\psi(\bar{x}, \bar{y}, M)$ et son complément. Soit $\mathrm{Infini}[\tau](\bar{x}, \bar{y})$ une formule sans quantificateurs équivalente à

$$\exists u_1, \ldots, u_{d+1} \bigwedge_{1 \leqslant i < j \leqslant d+1} u_i \neq u_j \wedge \bigwedge_{i=1}^{d+1} \tau(\bar{x}, \bar{y}, u_i).$$

Le vecteur signe de z est donc donné par $\mathcal{S}_G[\tau](\bar{x},\bar{y},\bar{t}) := \text{Infini}[\tau](\bar{x},\bar{y})$. Et on pose $\mathcal{F}_G(\bar{x},\bar{t}) := \text{vrai}$.

Pour $\Gamma = \Psi^d \cup \{G\}$, $\mathcal{B} = md$, et les formules associées définies ci-dessus, on va vérifier que les hypothèses de la proposition 9.1 sont satisfaites. Si $\vec{v}$ est une vecteur signe réalisable, il existe un point $z \in M$ qui en témoigne. Si le point z est du côté fini de $\psi_0(\bar{x},\bar{y}_0,\cdot)$ pour un couple $(\psi_0,\bar{y}_0)$, il existe $(\bar{\psi},\bar{t})$ localisant précisément z. Ainsi la formule $\mathcal{F}_{\bar{\psi}}(\bar{x},\bar{t})$ est satisfaite et $\vec{v} = \mathcal{S}_{\bar{\psi}}[\cdot](\bar{x},\cdot,\bar{t})$. Sinon $\vec{v} = \mathcal{S}_G[\cdot](\bar{x},\cdot,\bar{t})$ (pour tout $\bar{t}$). Réciproquement supposons qu'il existe $\gamma \in \Gamma$ et $\bar{t} \in D^{\mathcal{B}}$ vérifiant i) et ii). Si $\gamma = \bar{\psi}$, alors $\mathcal{F}_\gamma(\bar{x},\bar{t})$ prouve qu'il existe un point satisfaisant $\text{loc}_{\bar{\psi}}(\bar{x},\bar{t},\cdot)$ (par NonVide), que les points satisfaisant $\text{loc}_{\bar{\psi}}(\bar{x},\bar{t},\cdot)$ ont tous le même vecteur signe (par LocPrecise), et que les formules $\mathcal{S}_{\bar{\psi}}[\cdot](\bar{x},\cdot,\bar{t})$ définissent ce vecteur signe. Sinon $\gamma = G$; soit

$$A_{gen} := \{x \in M,\ \bigwedge_{\psi\in\Psi}\bigwedge_{\bar{y}\in D^m} |\psi(\bar{x},\bar{y},M)| < \infty \to x \in M \setminus \psi(\bar{x},\bar{y},M)\}.$$

Bien sûr $A_{gen} \neq \emptyset$ puisque c'est l'intersection d'un nombre fini d'ensembles cofinis. De plus il est clair que les points de A_{gen} ont le vecteur signe donné par $\mathcal{S}_G[\cdot](\bar{x},\cdot,\bar{t})$ (pour tout $\bar{t}$). □

9.3 Corps différentiellement clos

On rappelle qu'une dérivation sur le corps K est une application $d : K \to K$ tel que pour tout x, y dans K, $d(x+y) = d(x) + d(y)$ et $d(xy) = xd(y) + yd(x)$. Un corps différentiel K est un corps équipé d'une dérivation. Un polynôme différentiel en les variables $x_1,\ldots,x_k$ est un polynôme en les $d^j(x_i)$ pour $1 \leqslant i \leqslant k$ et $j \in \mathbb{N}$. L'ordre d'un polynôme différentiel $p(x)$ est le plus grand n tel que $x^{(n)}$ apparaisse dans p. On dit que K est un corps différentiellement clos si pour tous polynômes non constants f et g avec l'ordre de g strictement plus petit que l'ordre de f il existe x tel que $f(x) = 0 \wedge g(x) \neq 0$ [Mar96]. La structure S que l'on considère est maintenant un corps différentiellement clos de caractéristique nulle K. Commençons par deux remarques.

Lemme 9.1 *Soit $d, n \in \mathbb{N}$. Alors il existe $B' \in \mathbb{N}$ tel que pour tout $N \in \mathbb{N}$ et tous polynômes différentiels $p_1(x),\ldots,p_N(x)$ d'ordres et de degrés (en chacune des variables $x, x', x'', \ldots$) majorés par n et d respectivement, il existe $i_1,\ldots,i_{B'}$ tels que*

$$\forall x \in K\ \bigwedge_{j=1}^{B'} p_{i_j}(x) = 0 \to \bigwedge_{j=1}^{N} p_j(x) = 0.$$

Démonstration. Il est suffisant de prendre pour B' la dimension du K-espace vectoriel des polynômes d'ordres et de degrés majorés par n et d. On peut donc choisir $B' = (d+1)^{n+1}$. □

Lemme 9.2 *Soit $d, n \in \mathbb{N}$. Alors il existe $B'' \in \mathbb{N}$ tels que pour tous polynômes $p_1,\ldots,p_s$ d'ordres et de degrés majorés par n et d, on a la propriété suivante.. Soit $V = \{x \in$*

K, $p_1(x) = \ldots = p_s(x) = 0\}$. Alors pour tout $N \in \mathbb{N}$ et tous $q_1(x), \ldots, q_N(x)$ de degrés et d'ordres majorés par n et d, il existe $i_1, \ldots, i_{B''}$ tels que

$$\{x \in V, \bigwedge_{i=1}^{N} q_i(x) \neq 0\} = \emptyset \rightarrow \{x \in V, \bigwedge_{j=1}^{B''} q_{i_j}(x) \neq 0\} = \emptyset.$$

Démonstration. Par le lemme 9.1, on peut supposer $0 \leqslant s \leqslant B'$. Maintenant on fixe s. Pour $k \in \{1, \ldots, N\}$, soit $\bar{a}_k$ le uple composé des coefficients des polynômes $p_1, \ldots, p_s$ suivis des coefficients de q_k. Soit $\theta(\bar{a}_k, x) := \bigwedge_{i=1}^{s} p_i(x) = 0 \wedge q_k(x) \neq 0$. La NFCP (No Finite Cover Property), qui est vérifiée sur les corps différentiellement clos – voir [Mar96] section 2, nous donne une borne $B''(s)$. Ainsi la borne $B'' = \max\{B''(s),\ 0 \leqslant s \leqslant B'\}$ convient. □

Proposition 9.3 *Les corps différentiellement clos de caractéristique nulle ont l'effondrement actif-naturel.*

Démonstration. On va utiliser la proposition 9.1. Soit $\mathcal{T}$ un ensemble fini de formules atomiques de S : on peut supposer qu'elles sont de la forme $p(\bar{x}, \bar{y}, z) = 0$. Soit $\mathcal{P}$ l'ensemble fini des polynômes différentiels $p(\bar{x}, \bar{y}, z)$ apparaissant dans $\mathcal{T}$. Soit n et d bornant l'ordre et les degrés (en chacune des variables $x, x', x'', \ldots$) de ces polynômes. Soit B' et B'' les entiers donnés par les lemmes 9.1 et 9.2. Soit $\Gamma = \emptyset \cup \mathcal{P} \cup \ldots \cup \mathcal{P}^{B'}$ et $\mathcal{B} = mB'$. On va noter $\bar{t} \in D^{\mathcal{B}}$ le uple $\bar{t}_1.\bar{t}_2 \ldots \bar{t}_{B'}$, chaque $\bar{t}_i$ étant de longueur m. Pour $\bar{p} \in \Gamma$ et $\bar{t} \in D^{\mathcal{B}}$, définissons $\mathrm{loc}_{\bar{p}}(\bar{x}, \bar{t}, u) := \bigwedge_{i=1}^{|\bar{p}|} p_i(\bar{x}, \bar{t}_i, u) = 0$ (remarquons que ceci est vrai en particulier quand $|\bar{p}| = 0$). Pour $q \in \mathcal{P}$, soit $\mathcal{S}_{\bar{p}}[q = 0](\bar{x}, \bar{y}, \bar{t})$ une formule sans quantificateurs équivalente à $\forall u\ \mathrm{loc}_{\bar{p}}(\bar{x}, \bar{t}, u) \rightarrow q(\bar{x}, \bar{y}, u) = 0$. Soit $\mathrm{PartielNonVide}_{\bar{p},\bar{q}}(\bar{x}, \bar{t}, \bar{r})$ – pour $|r| = mB''$ et $|q| = B''$ – une formule sans quantificateurs équivalente à

$$\exists u \bigwedge_{i=1}^{|\bar{p}|} p_i(\bar{x}, \bar{t}_i, u) = 0 \wedge \bigwedge_{j=1}^{B''} (\mathcal{S}_{\bar{p}}[q_j = 0](\bar{x}, \bar{r}_j, \bar{t}) \vee q_j(\bar{x}, \bar{r}_j, u) \neq 0).$$

Posons

$$\mathcal{F}_{\bar{p}}(\bar{x}, \bar{t}) := \forall^a \bar{r} \bigwedge_{\bar{q} \in \mathcal{P}^{B''}} \mathrm{PartielNonVide}_{\bar{p},\bar{q}}(\bar{x}, \bar{t}, \bar{r}).$$

Montrons que Γ et les formules associées $\mathcal{F}_\gamma$ et $\mathcal{S}_\gamma$ définies ci-dessus vérifient la proposition 9.1. Considérons $\vec{v}$ un vecteur signe réalisé par un point $z \in M$. Le lemme 9.1 appliqué à l'ensemble des polynômes $q(\bar{x}, \bar{y}_0, \cdot)$ pour tout $(q, \bar{y}_0) \in \mathcal{P} \times D^m$ tel que $q(\bar{x}, \bar{y}_0, z) = 0$ nous donne s couples $(p_i, \bar{t}_i) \in \mathcal{P} \times D^m$ avec $0 \leqslant s \leqslant B'$. Montrons que $\gamma = \bar{p}$ et $\bar{t} = \bar{t}_1 \ldots . \bar{t}_s$ satisfont i) et ii). D'une part $\mathcal{S}_{\bar{p}}[\cdot](\bar{x}, \cdot, \bar{t}) = \vec{v}$: en effet $\models \mathcal{S}_{\bar{p}}[q = 0](\bar{x}, \bar{r}, \bar{t})$ si et seulement si $q(\bar{x}, \bar{r}, K) = 0$ contient $\bigcap_i p_i(\bar{x}, \bar{t}_i, K) = 0$, ce qui est vrai si et seulement si $q(\bar{x}, \bar{r}, z) = 0$. D'autre part $\models \mathcal{F}_{\bar{p}}(\bar{x}, \bar{t})$ puisque z en témoigne. Réciproquement, soit $\bar{p}, \bar{t}$ tels que $\models \mathcal{F}_{\bar{p}}(\bar{x}, \bar{t})$. Soit $\vec{v} = \mathcal{S}_{\bar{p}}[\cdot](\bar{x}, \cdot, \bar{t})$. Comme on a $\mathcal{F}_{\bar{p}}(\bar{x}, \bar{t})$, le lemme 9.2 nous dit que l'ensemble des points de K possédant ce vecteur signe est non vide. □

9.4 Élimination uniforme des quantificateurs

Nous nous intéressons maintenant à une notion introduite par Basu : l'élimination uniforme des quantificateurs [Bas99]. Une famille de formules $(\phi_n)_{n\in\mathbb{N}}$ sur la structure M est dite uniforme si elle est de la forme

$$\phi_n(\bar{x},\bar{y}) = Q^1_{1\leqslant i_1\leqslant n}\dots Q^m_{1\leqslant i_m\leqslant n}\phi(\bar{x},y_{i_1},\dots,y_{i_m})$$

où $Q_i \in \{\bigvee,\bigwedge\}$, et où ϕ est une formule du premier ordre sur M. On dit qu'une structure M a l'élimination uniforme des quantificateurs si étant donnée une famille uniforme $(\phi_n(z,\bar{x},\bar{y}))_{n\in\mathbb{N}}$ sur M, il existe une famille uniforme $(\psi_n(\bar{x},\bar{y}))_{n\in\mathbb{N}}$ sur M telle que pour tout n, $\psi_n(\bar{x},\bar{y})$ soit une formule sans quantificateurs équivalente à $\exists z\phi_n(z,\bar{x},\bar{y})$. Ceci implique bien sûr l'élimination des quantificateurs usuelle.

Proposition 9.4 *Une structure a l'élimination uniforme des quantificateurs si et seulement si elle a l'effondrement actif-naturel pour une unique relation unaire.*

Démonstration. Tout d'abord, remarquons que dans le cas où on a une relation unaire, toute formule active peut s'écrire sans utiliser le symbole I : il suffit pour cela de remplacer $I(t(\bar{w}))$ par $\exists^a v\ v = t(\bar{w})$. C'est ce que nous ferons dans la suite. Associons à chaque formule active une famille uniforme de formules, et réciproquement. Nous prendrons des formules sous forme prénexe uniquement, mais ce n'est pas impératif. Soit $\Phi(\bar{x},\bar{y})$ la famille uniforme donnée par la suite des

$$\phi_n(\bar{x},\bar{y}) = Q^1_{1\leqslant i_1\leqslant n}\dots Q^m_{1\leqslant i_m\leqslant n}\phi(\bar{x},y_{i_1},\dots,y_{i_m})$$

où $Q_i \in \{\bigvee,\bigwedge\}$. À cette famille nous faisons correspondre la formule active suivante

$$\psi(\bar{x}) := \mathcal{Q}^1 t_1\dots\mathcal{Q}^m t_m\ \phi(\bar{x},t_1,\dots,t_m)$$

avec $\mathcal{Q}^i = \exists^a$ (resp. $\forall^a$) si $Q^i = \bigvee$ (resp. $\bigwedge$). Bien sûr cette correspondance se définit aussi dans l'autre sens.

De plus $\Phi(\bar{x},\bar{y})$ et $\psi(\bar{x})$ sont reliés de la manière suivante : $(M,\mathcal{I}) \models \psi(\bar{x})$ si et seulement si $M \models \phi_n(\bar{x},\bar{y}_\mathcal{I})$ où $n = |\mathcal{I}|$ et $\bar{y}_\mathcal{I}$ liste les éléments de $\mathcal{I}$. Une structure a l'élimination uniforme des quantificateurs si et seulement si pour toute famille uniforme $\Phi(\bar{x},\bar{y},z)$, il existe une famille uniforme $\tilde{\Phi}(\bar{x},\bar{y})$ telle que : pour tout $\bar{x} \in M$ et tout $n \in \mathbb{N}$, $\exists z \in M\ \phi_n(\bar{x},\bar{y},z) \leftrightarrow \tilde{\phi}_n(\bar{x},\bar{y})$. De même, une structure a l'effondrement actif-naturel pour une relation unaire si et seulement si pour toute formule active $\psi(\bar{x},z)$, il existe une formule active $\tilde{\psi}(\bar{x})$ telle que $\forall\bar{x} \in M(\exists z \in M\ \psi(\bar{x},z) \leftrightarrow \tilde{\psi}(\bar{x}))$. Il est donc clair que ces deux notions sont équivalentes. □

Dans [Bas99] la question de savoir si les théories ACF_p (p premier ou nul) et DCF_0 (la théorie des corps différentiellement clos de caractéristique nulle) ont l'élimination uniforme des quantificateurs, est soulevée. La réponse à cette question est positive. Il est en effet montré dans [FZ99] (et rappelé au chapitre 7) que la "No Finite Cover Property" (NFCP) est une condition suffisante pour qu'une structure éliminant les quantificateurs

admette l'effondrement actif-naturel, et les deux théories mentionnées ci-dessus ont la NFCP – voir [Mar96] pour les corps différentiellement clos. Une méthode élémentaire (exposée précédemment dans ce chapitre) montre également que les théories ACF_p et DCF_0 ont l'effondrement actif-naturel. En particulier, ils ont l'effondrement actif-naturel dans le cas où la signature relationnelle $\mathcal{B}$ se compose d'une seule relation unaire, ce qui est équivalent à l'élimination uniforme des quantificateurs par la proposition 9.4. L'effondrement actif-naturel implique l'élimination uniforme des quantificateurs, qui à son tour implique l'élimination des quantificateurs. De plus, il existe une structure qui élimine les quantificateurs mais pas uniformément : la structure ternaire aléatoire [BL00]. Existe-t-il une structure qui admet l'élimination uniforme des quantificateurs mais pas l'effondrement actif-naturel ? Remarquons qu'aucune structure stable ne peut satisfaire cela – c'est une conséquence de [FZ99]. On peut également se poser toutes les questions intermédiaires : si $\bar{\alpha} \in (\mathbb{N} \setminus \{0\})^\infty$ est plus petit que $\bar{\beta} \in (\mathbb{N} \setminus \{0\})^\infty$ dans l'ordre lexicographique, alors une structure qui a l'effondrement actif-naturel pour les signatures d'arité $\bar{\alpha}$ l'a aussi pour les signatures d'arité $\bar{\beta}$, mais la réciproque est-elle vraie ? Si ce n'est pas le cas, il pourrait y avoir une hiérarchie infinie entre l'élimination des quantificateurs et l'effondrement actif-naturel.

Conclusion

Nous avons apporté certaines réponses concernant des questions de complexité et d'expressibilité sur les réels. Terminons avec quelques problèmes intéressants.

◦ Nous avons montré l'inclusion $\mathrm{NP}_{\mathbb{R}_{ovs}} \subset \mathrm{P}_{\mathbb{R}_{ovs}}(\mathrm{NP})$. Cette inclusion est probablement stricte puisque $\mathrm{P}_{\mathbb{R}_{ovs}}(\mathrm{NP})$ contient co-NP. Est-il possible de donner une caractérisation (ou une borne supérieure plus précise) de $\mathrm{NP}_{\mathbb{R}_{ovs}}$ où on aurait séparé comme ci-dessus l'aspect combinatoire de l'aspect algébrique ?

◦ Existe-t-il une famille d'arbres de calcul de profondeur polynomiale sur le corps ordonné des réels décidant un problème $\mathrm{NP}_{\mathbb{R}}$-complet, par exemple $4\mathrm{FEAS}_{\mathbb{R}}$? Rappelons que ceci permettrait d'obtenir un théorème de transfert pour $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$ [Koi00a]. Une question plus simple, mais déjà intéressante, est de trouver une famille d'arbres de calcul de profondeur polynomiale pour la programmation linéaire sur $\mathbb{R}$.

◦ Existe-t-il un algorithme parallèle rapide de localisation dans un arrangement d'hyperplans ? Plus précisément, étant donné un ensemble de $2^{(\log n)^{O(1)}}$ hyperplans de $\mathbb{R}^n$, essayer de construire un arbre de décision linéaire parallèle (avec un nombre polynomial de tests en chaque nœud) de profondeur $(\log n)^{O(1)}$ décidant l'union de ces hyperplans – une borne inférieure en $(\log n)^{O(1)}$ provenant de [Yao82].

◦ Nous avons montré l'encadrement $\Omega(\log\log n) \leqslant \mathrm{QR}(\mathrm{Pair}, n) \leqslant O(\sqrt{\log n})$ pour le rang de quantification nécessaire pour exprimer la parité restreinte sur un corps réels clos. Il serait intéressant de trouver de meilleures bornes.

Index

Bibliographie

[AHV95] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[AS00] P. Agarwal and M. Sharir. Arrangements and their applications. In *Handbook of Computational Geometry*, pages 49–119. North Holand, Amsterdam, 2000.

[Bas99] S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *Journal of the ACM*, 46(4) :537–555, july 1999.

[BB00] J. T. Baldwin and M. Benedikt. Stability theory, permutations of indiscernibles, and embedded finite models. *Trans. Amer. Math. Soc.*, 352(11) :4937–4969, 2000.

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Springer, 1997.

[BCSS96] L. Blum, F. Cucker, M. Shub, and S. Smale. Algebraic settings for the problem "P≠NP ?". In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 125–144. American Mathematical Society, 1996.

[BCSS98] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[BDG90] J.L. Balcázar, J. Diáz, and J. Gabarró. *Structural Complexity II*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1990.

[BDG95] J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, second edition, 1995.

[BDMM00] S. Ben-David, K. Meer, and C. Michaux. A note on non-complete problems in $\mathrm{NP}_{\mathbb{R}}$. *Journal of Complexity*, 16(1) :324–332, 2000.

[BEHW89] A. Blumer, A Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4) :929–965, 1989.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM J. Comput.*, 4(4) :431–442, 1975.

[BL96] M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. In *Proc. 11th IEEE Symposium on Logic in Computer Science*, pages 25–34, 1996.

[BL97] M. Benedikt and L. Libkin. Languages for relational databases over interpreted structures. In *Proc. 16th ACM Symposium on Principles of Database Systems*, pages 87–98, 1997.

[BL00] M. Benedikt and L. Libkin. Relational queries over interpreted structures. *Journal of the ACM*, 47(4) :644–680, 2000.

[BO83] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proc. 15th Ann. ACM Symp. Theory of Computing*, pages 80–86, 1983.

[Bou01] M. Bourgade. Séparations et transferts dans la hiérarchie polynomiale des groupes abéliens infinis. *Mathematical Logic Quaterly*, 47(4) :493–502, 2001.

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers : NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1) :1–46, July 1989.

[BST99] O. V. Belegradek, A. Stolboushkin, and M. A. Taitslin. Extended order-generic queries. *Annals of Pure and Applied Logic*, 97 :85–125, 1999.

[Bür00] P. Bürgisser. *Completeness and reduction in algebraic complexity theory.* Number 7 in Algorithms and Computation in Mathematics. Springer-Verlag, 2000.

[CG97] F. Cucker and D. Grigoriev. On the power of real Turing machines with binary inputs. *SIAM Journal on Computing*, 26(1) :243–254, 1997.

[CG01] F. Cucker and D. Grigoriev. There are no NP_W-hard sets. In *Proc MFCS'01*, volume 2136 of *Lecture Notes in Computer Science*, pages 285–291. Springer, 2001.

[Cha93] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete and Computational Geometry*, 9 :145–158, 1993.

[CK90] C. C. Chang and H. J. Keisler. *Model Theory.* North Holland, third edition, 1990.

[CK95] F. Cucker and P. Koiran. Computing over the reals with addition and order : Higher complexity classes. *Journal of Complexity*, 11 :358–376, 1995.

[CK99] O. Chapuis and P. Koiran. Saturation and stability in the theory of computation over the reals. *Annals of Pure and Applied Logic*, 99(1-3) :1–49, 1999.

[CKM97] F. Cucker, P. Koiran, and M. Matamala. Complexity and dimension. *Information Processing Letters*, 62 :209–212, 1997.

[Cla87] K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2(2) :195–222, 1987.

[CO97] J. Cai and M. Ogihara. Sparse sets versus complexity classes. In L. Hemaspaandra and A. Selman, editors, *Complexity theory retrospective II*. Springer, 1997.

[DL78] D. Dobkin and R. J. Lipton. A lower bound of $(1/2)n^2$ on linear search programs for the knapsack problem. *Journal of Computer and System Sciences*, 16 :413–417, 1978.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[FK98] H. Fournier and P. Koiran. Are lower bounds easier over the reals ? In *Proc. 30th ACM Symposium on Theory of Computing*, pages 507–513, 1998.

[FK00] H. Fournier and P. Koiran. Lower bounds are not easier over the reals : Inside PH. In *Proc 27th International Colloquim on Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 832–843. Springer, 2000.

[Fou98] H. Fournier. Transferts sur les réels avec addition. Mémoire de DEA, École Normale Supérieure de Lyon, Juillet 1998.

[Fou01a] H. Fournier. Quantifier rank for parity of embedded finite models. In *Proc MFCS'01*, volume 2136 of *Lecture Notes in Computer Science*, pages 375–386. Springer, 2001.

[Fou01b] H. Fournier. Sparse NP-complete problems over the reals with addition. *Theoretical Computer Science*, 255(1-2) :607–610, 2001.

[FW91] D. Flath and S. Wagon. How to pick out the integers in the rationals : An application of number theory to logic. *American Mathematical Monthly*, 98 :812–823, 1991.

[FZ99] J. Flum and M. Ziegler. Pseudo-finite homogeneity and saturation. *Journal of Symbolic Logic*, 64(4) :1689–1699, 1999.

[Gaß97] C. Gaßner. On NP-completeness of linear machines. *J. Complexity*, 13(2) :259–271, 1997.

[GG98] E. Grädel and Y. Gurevich. Metafinite model theory. *Information and Computation*, 140(1) :26–81, 1998.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freedman and Company, New York, 1979.

[GK97] D. Grigoriev and M. Karpinski. Randomized $\Omega(n^2)$ for knapsack. In *29th Ann. ACM Symp. Theory of Computing*, pages 76,85, 1997.

[Goo94] J. B. Goode. Accessible telephone directories. *Journal of Symbolic Logic*, 59(1) :92–105, 1994.

[Gri97] D. Grigoriev. Nearly sharp complexity bounds for multiprocessor algebraic computations. *Journal of Complexity*, 13(1) :50–64, March 1997.

[GRS90] R. L. Graham, B. L. Rothschild, and J. H. Spencer. *Ramsey Theory*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2nd edition, 1990.

[GS97] S. Grumbach and J. Su. Queries with arithmetical constraints. *Theoretical Computer Science*, 173 :151–181, 1997.

[Hem01] A. Hemmerling. On P versus NP for parameter-free programs over algebraic structures. *Mathematical Logic Quaterly*, 47(1) :67–92, 2001.

[Hod97] W. Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.

[HW87] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2 :127–151, 1987.

[Imm98] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, 1998.

[Joh90] D. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2. Elsevier, 1990.

[Juk01] S. Jukna. *Extremal Combinatorics - With Application in Computer Science*. EATCS Texts in Theoretical Computer Science. Springer, 2001.

[KKR95] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer System Sciences*, 51 :26–52, 1995. Extended abstract in *PODS'90*.

[KL82] R. Karp and R. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28 :191–209, 1982.

[KLP00] G. Kuper, L. Libkin, and J. Paredaens, editors. *Constraint Databases*. Springer-Verlag, 2000.

[Koi94] P. Koiran. Computing over the reals with addition and order. *Theoretical Computer Science*, 133(1) :35–48, 1994.

[Koi97] P. Koiran. A weak version of the Blum, Shub & Smale model. *Journal of Computer and System Sciences*, 54 :177–189, 1997.

[Koi99a] P. Koiran. Elimination of parameters in the polynomial hierarchy. *Theoretical Computer Science*, 215(1-2) :289–304, 1999.

[Koi99b] P. Koiran. The real dimension problem is $\mathrm{NP}_{\mathbb{R}}$-complete. *Journal of Complexity*, 15 :227–238, 1999.

[Koi00a] P. Koiran. Circuits versus trees in algebraic complexity. In *STACS*, volume 1770 of *Lecture notes in computer science*, pages 35–52. Sprinegr, 2000.

[Koi00b] P. Koiran. Transfer theorems via sign conditions. LIP Research Report 2000-13, 2000.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[KV00] B. Korte and J. Vygen. *Combinatorial Optimization. Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, 2000.

[MadH84] F. Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *Journal of the ACM*, 31(3) :668–676, 1984.

[MadH88] F. Meyer auf der Heide. Fast algorithms for n-dimensional restrictions of hard problems. *Journal of the ACM*, 35(3) :740–747, 1988.

[Mah82] S. Mahaney. Sparse complete sets for NP : Solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25 :130–143, 1982.

[Mar96] D Marker. Model theory of differential fields. In *Model theory of fields*, number 5 in Lecture notes in logic. Springer, 1996.

[Mat] J. Matoušek. Lectures on discrete geometry. Available on the web at http://www.ms.mff.cuni.cz/acad/kam/matousek/dg.html.

[Mat98] J. Matoušek. Geometric set systems. In *2nd. European Math. Congress*, volume 169 of *Progr. Math.*, pages 1–27. Birkäuser, 1998.

[Mee92] K. Meer. A note on a $P \neq NP$ result for a restricted class of real machines. *Journal of Complexity*, 8 :451–453, 1992.

[Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2) :286–303, 1993.

[Mic94] C. Michaux. P $\neq$ NP over the nonstandard reals implies P $\neq$ NP over $\mathbb{R}$. *Theoretical Computer Science*, 133 :95–104, 1994.

[MM99] M. Matamala and K. Meer. On the computational structure of the connected components of difficult sets. *Information Processing Letters*, 72(3-4) :83–90, 1999.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[OT91] P. Orlik and H. Terao. *Arrangements of hyperplanes*. Springer, Berlin, 1991.

[OVdB96] M. Otto and J. Van den Bussche. First-order queries on databases embedded in an infinite structure. *Information Processing Letters*, 60(1) :37–41, 1996.

[Poi95] B. Poizat. *Les Petits Cailloux*. Nur Al-Mantiq Wal-Ma'rifah **3**. Aléas, Lyon, 1995.

[Poi00] B. Poizat. *A course in model theory*. Universitext. Springer, 2000.

[Por99] N. Portier. Stabilité polynomiale des corps différentiels. *Journal of Symbolic Logic*, 64(2) :803–816, 1999.

[Rob49] J. Robinson. Definability and decision problems in arithmetic. *Journal of Symbolic Logic*, 14 :98–114, 1949.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New-York, 1986.

[SS96] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "P=NP". *Duke Mathematical Journal*, 81(1) :47–54, 1996.

[Sto85] L. Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4) :849–861, 1985.

[Str90] V. Strassen. Algebraic complexity theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 11. Elsevier, 1990.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity.* EATCS Texts in Theoretical Computer Science. Springer, 1999.

[Weg87] I. Wegener. *The Complexity of Boolean Functions.* B. G. Teubner, and John Wiley & Sons, 1987.

[Yao82] A. Yao. On parallel computation for the knapsack problem. *Journal of the ACM*, 29(3) :898–903, 1982.