Numéro d'ordre : 240-98 Année 1998

THÈSE
présentée devant l'UNIVERSITÉ CLAUDE BERNARD - LYON I
pour obtenir le titre de DOCTEURE EN SCIENCES
dans la spécialité MATHÉMATIQUES

par

Natacha PORTIER

Complexité algébrique
Stabilité polynomiale des corps différentiels
&
Résolutions universelles pour des problèmes NP-complets

soutenue le mercredi 9 décembre 1998

Jury :
Vincent BLONDEL
chargé de cours à l'Université de Liège, examinateur
Fokko du CLOUX
professeur à l'Université Claude Bernard de Lyon, examinateur
Françoise DELON
directrice de recherches au CNRS à Paris VII, présidente du jury
Christian MICHAUX
chef de travaux à l'Université de Mons, rapporteur
Bruno POIZAT
professeur à l'Université Claude Bernard de Lyon, directeur de thèse

Rapporteurs :
Felipe CUCKER, professeur à l'Universitat Pompeu Fabra de Barcelone,
Christian MICHAUX, chef de travaux à l'Université de Mons.

A Lucien e Andrée
Sabé, Independènci

Remerciements

Je souhaite remercier toutes les personnes ayant contribué de près ou de loin à cette thèse. Tout d'abord, ma plus sincère gratitude va à Bruno Poizat, qui a encadré mon travail depuis le DEA, ainsi qu'à l'Institut Girard Desargues et à l'équipe du thème logique qui m'ont chaleureusement accueillie pendant ces trois années de thèse. En particulier, Olivier Chapuis et Christine Charretton ont toujours su me soutenir et me conseiller sagement. Le groupe de travail hebdomadaire « complexité algébrique » de l'IGD et du LIP de l'ENS Lyon a été un rendez-vous régulier qui m'apporta beaucoup et où je fis mes premiers exposés dans une atmosphère détendue et bienveillante. J'en remercie les organisateurs, dont Pascal Koiran, ainsi que les participants.

Je remercie également les membres de mon jury, et en particulier ceux qui ont su trouver un train en ce mois de décembre où ils se firent rares. Merci à Françoise Delon pour avoir accepté de diriger mon jury de thèse, à Felipe Cucker et Christian Michaux pour m'avoir encouragée au début de ma thèse et pour avoir lu attentivement mon travail et l'avoir rapporté, ainsi qu'à Vincent Blondel et Fokko du Cloux pour avoir bien voulu faire partie de mon jury. Merci également à tous ceux qui m'ont fait l'honneur d'assister à ma soutenance.

Mes collègues de bureau et les autres membres du laboratoire, et en particulier Vincent Borrelli, Nik Lygeros et Michel Mizony, m'ont aidée par leur expérience et leurs conseils, et m'ont appris tout ce qu'une doctorante, une chercheuse, une enseignante et une utilisatrice d'unix ne doit pas ignorer...

Pour finir, je désire exprimer ma reconnaissance envers les membres de ma famille, pour le soutien qu'ils m'ont apporté tout au long de la réalisation de ce travail, et le jour de ma soutenance, malgré route et aéroport embrumé.

Table des matières

Chapitre 1

Introduction

La complexité classique étudie des algorithmes agissant sur des mots booléens, c'est-à-dire sur des suites finies de 0 et de 1. Un problème est alors un sous-ensemble de $\{0;1\}^*$. Les algorithmes ainsi que les classes de complexité, qui sont des ensembles de problèmes, sont définis à partir des machines de Turing. On pourra par exemple trouver les définitions dans le livre de Balcázar, Díaz et Gabarró ([3]). Suivant Poizat([27]), nous appellerons ce modèle le modèle standard. Par la suite, nous utiliserons les algorithmes standards de façon peu formelle, et nous nous contenterons de décrire les opérations à effectuer sans les traduire en termes de machines de Turing.

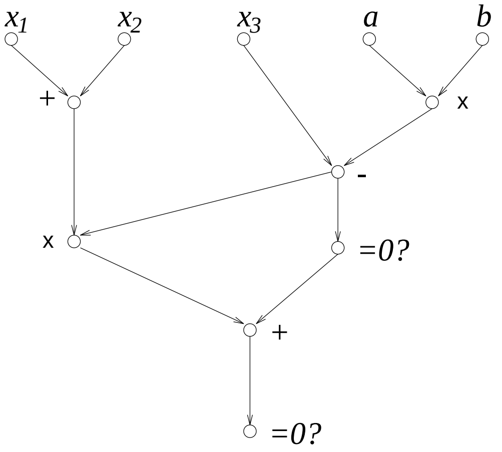
Une autre approche est celle de la complexité algébrique : nous comptons le nombre d'opérations nécessaires à un calcul. En 1989, L. Blum, M. Shub et S. Smale ont introduit dans [7] des machines opérant sur un anneau R (éventuellement, R est un corps ordonné), en supposant le coût d'une opération polynomiale constant. Dans [27], B.Poizat a défini des machines de Turing pour un modèle d'une théorie quelconque. Elles généralisent les machines de Turing standards et les machines sur un anneau de Blum, Shub et Smale. Il montre que la complexité dans une structure quelconque peut être définie avec des circuits. Nous nous limiterons ici à ce point de vue.

Les circuits permettent de définir des ensembles. Les ensembles définissables par des circuits sont exactement les ensembles définissables par des formules sans quantificateurs. L'intérêt des circuits est de se présenter sous une forme plus concise que les formules. Dans la pratique, quand nous voulons évaluer une formule, nous commençons par calculer tous les termes qui y apparaissent. Quand un terme se trouve à différents endroits, nous ne le calculons qu'une fois. Les circuits formalisent cette technique naturelle de calcul. Dans le circuit qui correspond à la formule, le terme peut n'apparaître qu'une fois, mais être utilisé plusieurs fois.

Nous allons définir les circuits dans le cas particulier des anneaux et des anneaux différentiels. Un anneau différentiel est un anneau muni d'une dérivée. Nous définirons plus précisément ce qu'est une dérivée et un anneau différentiel au paragraphe 2.2.1.

Un **circuit** est un graphe fini, orienté, et sans cycle orienté. Les sommets sont appelés **portes** , et les arêtes sont appelées **flèches**. Les portes qui ne reçoivent pas de flèches sont les **entrées** , et les portes qui n'en émettent pas sont des **sorties** . Les portes sont étiquetées. Un **circuit arithmétique** sur l'anneau K est un circuit dont les entrées sont étiquetées par un élément de K, appelé **paramètre** , ou par une variable, et dont les autres portes reçoivent deux flèches et sont étiquetées par une des opérations de l'anneau $\times$, $+$ ou -.

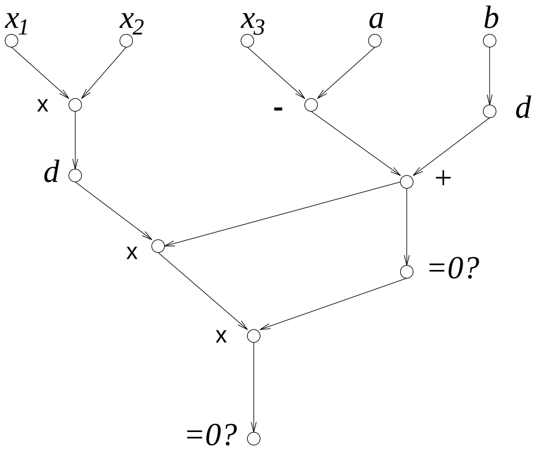
Un **circuit au sens des anneaux sur** K, ou circuit sur l'anneau K, est un circuit arithmétique sur K qui, en outre, peut avoir des portes qui ne reçoivent qu'une seule flèche et sont étiquetées par le test « $= 0?$ ».
Exemple de circuit sur l'anneau K :



Un **circuit arithmétique différentiel** sur un anneau différentiel K est un circuit arithmétique sur K dont certaines portes ne reçoivent qu'une flèche et sont étiquetées par d. Cela revient à calculer un polynôme différentiel des entrées du circuit. Grâce au lemme 5 (de réduction), nous pouvons le faire de manière algorithmique et polynomiale : nous remplaçons le circuit arithmétique différentiel par un circuit arithmétique équivalent dont les entrées sont les dérivées successives des entrées du précédent. Cette façon de voir est importante pour la démonstration du théorème principal du chapitre 2 : l'élimination des paramètres dans les corps différentiels.

Un **circuit au sens des anneaux différentiels sur** K , ou circuit sur l'anneau différentiel K, est un circuit arithmétique différentiel sur K qui peut de plus avoir des portes de test.

Exemple de circuit sur l'anneau différentiel K :



Un circuit est **sans paramètre** si ses seuls paramètres sont 0 et 1. Remarquons qu'un circuit sans paramètre peut se coder par un mot booléen. Une suite de circuits peut donc être produite par une machine de Turing classique. Nous dirons qu'une suite de circuits $(C_n)_n$ est produite par un algorithme standard travaillant en temps polynomial s'il existe une machine de Turing T et un polynôme q tels que sur l'entrée n, T calcule C_n en temps au plus $q(n)$.

Si un circuit a n entrées étiquetées par les variables $\bar{x} = (x_1, ..., x_n)$, le circuit est noté $C(\bar{x})$. Le **calcul sur un n-uple** d'éléments de K s'effectue alors de manière naturelle, de proche en proche. Aux variables d'entrée sont affectés les éléments de K. Une porte étiquetée par une opération effectue cette opération. Une porte étiquetée par une test émet 1 si la valeur qu'elle reçoit est nulle, et 0 sinon. Nous pouvons nous demander dans quel ordre effectuer les calculs. En fait, les flèches induisent un ordre partiel sur l'ensemble des portes. Il est facile de compléter cet ordre en ordre total. Nous pouvons le faire avec un algorithme polynomial, ou nous pouvons supposer que les circuits sont donnés avec un ordre total sur leurs portes. Par la suite, nous ne nous préoccuperons pas de l'ordre dans lequel les calculs sont effectués.

Dans le cas général défini par Poizat ([15, 26]), si nous ne nous restreignons pas comme ici à des structures d'anneau ou d'anneau différentiel, nous avons besoin de portes de sélection. Ce sont des portes qui reçoivent trois flèches et qui permettent un calcul par cas : si x est nul le calcul donne y, si x vaut 1 le calcul donne z, et sinon n'importe quoi de calculable. Si nous avons à notre disposition les opérations arithmétiques, une telle porte peut être simulée par le polynôme $(1-x)y+xz$. C'est pour cela qu'ici, nous n'utilisons pas de porte de sélection.

La **taille** d'un circuit est son nombre de portes. Elle est aussi appelée **temps de calcul** du circuit. Certains définissent le temps de calcul en excluant les entrées, mais cela ne change rien au niveau polynomial. La **profondeur** d'un circuit est la longueur du plus long chemin qui relie une entrée à une sortie. Deux circuits sont par définition **équivalents** si, pour toute entrée, les deux calculs sont égaux.

Un **circuit à paramètres** $C(\bar{x},\bar{a})$, où $\bar{a}$ est un uple d'éléments de K, est un circuit $C(\bar{x},\bar{y})$ pour lequel les variables $\bar{y}$ sont fixées à $\bar{a}$. Ainsi, si $C(x_1,...,x_n,\bar{a})$ est un circuit à paramètres sur K, il décide d'un sous-ensemble E de K^n : l'ensemble des entrées $\bar{x}$ acceptées. Pour décider d'un problème X de K, c'est-à-dire d'un ensemble de mots de longueur finie d'éléments de K, il nous faut donc un circuit par taille d'entrée. Nous dirons qu'une suite de circuits $C_n(x_1,...,x_n,\bar{a}_n)$ décide de X si pour tout entier n, $(x_1,...,x_n)$ appartient à X si et seulement si $C_n(x_1,...,x_n,\bar{a}_n)$ calcule 1. Cela signifie que pour tout n, C_n décide de $X\cap K^n$. Pour définir les classes de complexité P et NP, nous utilisons des suites de circuits pour lesquelles les paramètres $\bar{a}_n$ ne dépendent pas de n. Cela revient à fixer un nombre fini de paramètres pour tout l'algorithme.

Les sorties d'un circuit arithmétique sur un anneau calculent des polynômes dont les variables sont les variables d'entrée du circuit, et dont les coefficients sont des polynômes à coefficients entiers des paramètres d'entrée. Dans le livre de Blum, Cucker, Shub et Smale ([6]), si P est un polynôme à n variables et à coefficients dans $\mathbf{Z}$, un calcul de longueur m pour P est une suite $(u_0, ..., u_{n+m})$ de polynômes tels que $u_0 = 1$, $u_1, ..., u_n$ soient les variables $t_1, ..., t_n$, et pour tout i compris entre $n+1$ et $n+m$, il existe j et l strictement inférieurs à i tels que soit $u_i = u_j - u_l$, soit $u_i = u_j + u_l$, soit $u_i = u_j \times u_l$. Le temps de calcul du polynôme est alors le plus petit m tel qu'une telle suite existe. Cela revient à compter les portes d'un circuit arithmétique sans paramètre calculant P sans tenir compte des entrées.

Soit C un circuit arithmétique, à une seule variable d'entrée x, sur un anneau K infini. Soit $P(x)$ le polynôme calculé par une de ses sorties. Ce polynôme est-il identiquement nul ? Nous pouvons le savoir facilement si nous connaissons son développement : le polynôme est identiquement nul si et seulement si tous les coefficients de son développement sont nuls. Or nous ne connaissons du polynôme qu'un circuit qui le calcule. Nous pouvons certes en déduire la forme développée. Mais alors que le circuit est de taille τ, la forme développée peut faire apparaître un nombre exponentiel en τ de monômes. C'est le cas par exemple pour $(1+x)^{2^n}$, qui peut se calculer par un circuit de taille $(n+4)$, mais qui compte (2^n+1) monômes dans sa forme développée. Or, en général, nous voulons garder des objets de taille minimale, et donc si possible polynomiale. Développer les polynômes n'est donc pas une bonne méthode. Une façon rapide de faire nous sera donnée par le théorème des témoins de [6]. Cette question est très importante pour l'élimination des paramètres. Considérons un élément a d'un corps K et un polynôme $P(x, a)$ à coefficients dans un sous-corps k de K. Le polynôme est donné par un circuit arithmétique $C(x, a)$. Peut-on savoir si $P(x, a) = 0$ pour un élément x de k sans calculer dans K, en n'autorisant que des calculs dans k, donc en particulier sans calculer $P(x, a)$? Et ceci de manière algorithmique et polynomiale ? C'est la question de l'élimination des paramètres. Nous verrons que c'est possible dans le cas des corps (paragraphe 2.1) et des corps différentiels (paragraphe 2.2) de caractéristique nulle.

Les fonctions calculées par les portes d'un circuit au sens des anneaux sont des polynômes conditionnés par des tests : ce sont des fonctions définies par cas, chaque cas étant déterminé par une formule libre, c'est-à-dire par une combinaison booléenne de tests sur des polynômes, et telles que la fonction soit un polynôme dans chaque cas. Une telle fonction $y = f(\bar{x})$ est définie par une suite finie de formules sans quantificateurs $\phi_n(\bar{x})$ et une suite de polynômes $f_n(\bar{x})$ telles que $y = f_i(\bar{x})$ si et seulement si $\phi_i(\bar{x})$.

Un **circuit de décision** est un circuit qui n'a qu'une sortie, laquelle est étiquetée par un test. S'il a n entrées, il décide un ensemble E de n-uples s'il calcule 1 pour un n-uple de cet ensemble et 0 sinon.

Nous pouvons associer à tout circuit $C(\bar{x})$ de décision une formule du premier ordre sans quantificateur $\phi(\bar{x})$, qui est en quelque sorte la forme développée du circuit : $C(\bar{x})$ calcule 1 sur l'entrée $\bar{a}$ si et seulement si $K \vDash \phi(\bar{a})$. Par la suite, nous écrirons directement $K \vDash C(\bar{a})$.

Un ensemble de suites finies d'éléments de K est appelé problème sur K. Nous pouvons maintenant définir des classes de complexité, qui sont des ensembles de problèmes particuliers :

Définition 1 *Si K est un corps et A un sous-ensemble de K, $D_K(A)$ est l'ensemble des problèmes Π de K définissables à paramètres dans A , c'est-à-dire tels qu'il existe une suite de formules $(\phi_n(x_1, ..., x_n, \bar{c}_n))_{n\geq 1}$ du premier ordre du langage des corps, où les $\bar{c}_n$ sont des paramètres de A qui dépendent de n, telles que pour tout uple $\bar{x} = (x_1, ..., x_n)$ de K, $\bar{x}$ appartient à Π si et seulement si K satisfait $\phi_n(\bar{x}, \bar{c}_n)$.*

Définitions 2 *Si K est un anneau différentiel, un problème Π est $\mathbb{P}_K$ (resp. $\mathbb{P}_K^d$) s'il existe une suite $(C_n(x_1, ..., x_n, y_1, ..., y_m))_{n\geq 0}$ de circuits de décision au sens des anneaux (resp. au sens des anneaux différentiels) sans paramètre , dont la taille est bornée polynomialement en n, et des éléments $\bar{c} = (c_1, ..., c_m)$ de K tels que pour toute suite $\bar{a} = (a_1, ..., a_n)$ d'éléments de K, $\bar{a}$ appartienne à Π si et seulement si $C_n(\bar{a}, \bar{c})$ calcule 1. Le problème Π est P_K (resp. P_K^d) si, de plus, la suite de circuits sans paramètres est calculable par un algorithme standard en temps polynomial en n.*

Un problème Π est $N\mathbb{P}_K$ (resp. $N\mathbb{P}_K^d$, NP_K , NP_K^d) s'il existe un problème Σ qui est $\mathbb{P}_K$ (resp. $\mathbb{P}_K^d$, P_K, P_K^d), et un polynôme q, tels qu'une suite $\bar{x} = (x_1, ..., x_n)$ d'éléments de K soit dans Π si et seulement s'il existe une suite $\bar{y} = (y_1, ..., y_{q(n)})$ d'éléments de K telle que la suite $(\bar{x}, \bar{y})$ appartienne à Σ.

Si C est une classe de complexité, $\mathbb{C}$ est sa version non-uniforme. Si nous astreignons les paramètres des circuits à être dans un sous-ensemble A de K, nous obtenons la classe $C(A)$ des problèmes C à paramètres dans A.

Nous voyons ici que le problème $P = NP$? est un problème d'élimination rapide des quantificateurs. En particulier, le problème n'a aucun sens dans une structure qui n'élimine pas les quantificateurs.

Définition 3 *Une fonction f est calculable en temps polynomial au sens de l'anneau K (resp. de l'anneau différentiel K) s'il existe une suite de circuits $(C_n(x_1, ..., x_n, y_1, ..., y_k))_n$ au sens de l'anneau K (resp. de l'anneau différentiel K), produite par une machine de Turing standard travaillant en temps polynomial en n, et un k-uple $(a_1, ..., a_k)$ d'éléments de M tels que $C_n(\bar{x}, \bar{a})$ calcule $f(x_1, ..., x_n)$.*

Définitions 4 *Un problème Π est NP_K-dur (resp. NP_K^d-dur) si pour tout problème Y qui est NP_K (resp. NP_K^d), il existe une fonction f de K^* dans K^* calculable en temps polynomial au sens de l'anneau K (resp. de l'anneau différentiel K) et telle que pour tout mot de longueur finie u, $u \in Y$ si et seulement si $f(u) \in \Pi$. Pour les classes uniformes, nous exigeons moins de la fonction de réduction . Il suffit qu'elle soit produite par une suite de circuits de tailles polynomiales et avec un nombre fini de paramètres, et il n'est pas nécessaire qu'elle puisse être produite par un algorithme standard. Un problème Π est $N\mathbb{P}_K$-dur (resp. $N\mathbb{P}_K^d$-dur) si pour tout problème Y qui est $N\mathbb{P}_K$ (resp.$N\mathbb{P}_K^d$), il existe une fonction f de K^* dans K^*, calculée par une suite de circuits $(C_n(x_1, ..., x_n, a_1, ..., a_k))_n$ au sens de l'anneau K (resp. de l'anneau différentiel K), dont les tailles sont bornées par un polynôme $q(n)$, et où les $(a_1, ..., a_k)$ sont des éléments de K, et telle que pour tout mot de longueur finie u, $u \in Y$ si et seulement si $f(u) \in \Pi$.*

Un problème est NP_K-complet s'il est à la fois NP_K et NP_K-dur. Nous définissons de la même façon les classes $N\mathbb{P}_K$-complet, NP_K^d-complet et $N\mathbb{P}_K^d$-complet.

Définition 5 *Soit C une classe de complexité sur un anneau K (resp. un anneau différentiel K). La classe $C/poly$ est l'ensemble des problèmes Π sur l'ensemble K tels qu'il existe un problème X dans la classe C, un polynôme p et une fonction f de $\mathbf{N}$ dans K^∞ tel que pour tout entier n, $f(n)$ soit de longueur au plus $p(n)$ et $(x_1, ..., x_n)$ appartienne à Π si et seulement si la concaténation $(x_1, ..., x_n)f(n)$ appartient à X.*

Nous avons une définition équivalente pour la classe $P/poly$:

Lemme 1 *Un problème* Π *est* $P/poly$ *au sens des anneaux (resp. des anneaux différentiels) s'il existe un polynôme* p*, une suite de circuits de décision* $(C_n(x_1, ..., x_n, \bar{c}_n))_{n\geq 1}$ *au sens des anneaux (resp. des anneaux différentiels) , tels que pour tout* n *entier* $\bar{c}_n$ *soit un uple d'éléments de* K*,* C_n *un circuit de taille au plus* $p(n)$*, et* $(x_1, ..., x_n)$ *appartienne à* Π *si et seulement si* $C_n(x_1, ..., x_n, \bar{c}_n)$ *calcule 1.*

Remarquons que dans le cas standard, les classes $\mathbb{P}$ et $P/poly$ sont confondues. En effet, se donner un nombre polynomial de paramètres booléens revient à se donner la suite de circuits. Nous pourrions également définir la classe $\mathbb{P}/poly$, mais celle-ci est toujours confondue avec la classe $P/poly$. En effet, à un nombre polynomial de paramètres utilisés par les circuits, il suffit d'ajouter un nombre polynomial de paramètres booléens qui codent les circuits.

C. Michaux introduit ([25]) la classe $P/const$, intermédiaire entre les classes P et $P/poly$. Dans le cas des corps et dans celui des corps différentiels, et grâce à l'élimination des paramètres, cette classe est confondue avec la classe P. Nous n'en parlons donc pas ici.

Dans le chapitre 2, nous étudions la complexité dans les corps et les corps différentiels de caractéristique nulle. La majorité des résultats qui y sont présentés va être l'objet d'une publication dans *the Journal of Symbolic Logic* ([28]). Les paragraphes 2.1.3, 2.2.4, 2.2.5 et 2.2.6 sont des ajouts.

Dans le premier paragraphe, nous étudions la complexité dans les corps, et nous redémontrons le théorème d'élimination des paramètres en utilisant les circuits. Dans le deuxième paragraphe, nous généralisons ce théorème au cas des corps différentiels. Nous étudions ensuite quelques conséquences de ce théorème.

Dans le troisième chapitre, le point de vue est plus général, et nous nous intéressons à la complexité sur un modèle d'une théorie quelconque, en généralisant la notion de circuit. Nous étudions alors une notion plus forte que la NP-complétude : les résolutions universelles. Celles-ci sont définies dans le cas standard par Manindra Agrawal et Somenath Biswas, qui montrent un théorème pour les caractériser. Nous généralisons leur définition et le théorème aux calculs effectués dans une structure arbitraire, en particulier aux calculs sur les réels définis par Blum, Shub et Smale. Nous étudions ensuite des exemples de telles résolutions, dans des structures réelles, et faisant intervenir par exemple des réseaux neuronaux. Ce chapitre est paru sous forme d'article dans *Theoretical Computer Science* ([29]).

Chapitre 2

Stabilité polynomiale des corps différentiels

Tous les corps considérés sont de caractéristique nulle.

Si k est une sous-structure de K, si Π est un problème P sur K, nous pouvons nous demander si sa restriction Π_k aux mots sur k est P sur k. Si cela est vrai pour tous les problèmes et pour tous les modèles k et K d'une théorie T, cette théorie est appelée P-stable par Chapuis et Koiran ([9]). Pour que le problème Π_k soit P, la suite de circuits qui le définit doit utiliser des paramètres uniquement de k. Si nous nous plaçons dans une théorie ω-stable, comme la théorie des corps algébriquement clos ou la théorie des corps différentiellement clos, nous savons qu'une telle suite utilisant uniquement des paramètres de k existe. Il nous faut encore trouver une manière polynomiale d'obtenir cette suite à partir de celle qui définit Π. C'est une version algorithmique et même polynomiale de la définissabilité des types sans quantificateurs. Pour les corps, la P-stabilité est montrée par Blum, Cucker, Shub et Smale dans [6]. C'est l'objet du premier paragraphe. Dans le deuxième paragraphe, nous le montrons pour les corps différentiels. Cela nous conduit à comparer la question $P = NP$ dans les corps algébriquement clos et dans les corps différentiellement clos. Puis, nous étudions le problème des grandes racines et celui des grandes puissances, et nous montrons que nous ne pouvons pas les résoudre rapidement, même à l'aide de la dérivée.

2.1 L'élimination des paramètres dans les corps

2.1.1 Le théorème des témoins

Si Π est un problème P sur le corps K, alors il existe une suite (A_n) de circuits de tailles polynomiales à paramètres dans K qui résout Π. Si Π_k est la restriction de Π à un sous-corps k de K, est-ce que Π_k est P_k ? La réponse est oui. Pour le montrer, il nous faut trouver une suite de circuits (B_n) de tailles polynomiales et à paramètres dans k qui résout Π_k. Nous pouvons obtenir (B_n) en transformant (A_n). Pour cela, il nous faut éliminer les paramètres de (A_n) qui ne sont pas dans k. C'est l'objet du théorème des témoins de Blum, Cucker, Shub et Smale ([6]). Commençons par l'énoncer, généralisé pour un corps K quelconque de caractéristique nulle.

Théorème 1 (théorème des témoins)

Soit $F(\bar{x},\bar{t}) = F(x_1,...,x_r,t_1,...,t_s)$ un polynôme produit par un circuit arithmétique d'entrées $1, x_1, ..., x_r, t_1, ..., t_s$ et de taille τ. Soit $N \geq 2^{4(r+s)\tau^2+4}$. Pour tout uple $\bar{a} = (a_1, ..., a_r)$ d'éléments du corps K de caractéristique nulle, nous définissons $F_{\bar{a}}(\bar{t}) = F(\bar{a},\bar{t})$. Alors $F_{\bar{a}}$ est le polynôme nul si et seulement si pour tout w appartenant à $\{2, a_1, ..., a_r\}$, $F_{\bar{a}}(w^N, w^{N^2}, ..., w^{N^s}) = 0$.

Démonstration du théorème 1 : Ce théorème est un ensemble d'énoncés universels du premier ordre de la théorie des corps. Le cas où K est la clôture algébrique de $\mathbf{Q}$ est traité dans [6]. Puisque la théorie des corps algébriquement clos de caractéristique nulle est complète, ce théorème est vrai pour tout corps algébriquement clos. De plus, tout corps possède une extension algébriquement close. Donc le théorème est vrai pour tout corps de caractéristique nulle. $\square$

2.1.2 L'élimination des paramètres dans les corps pour la classe P

Nous pouvons maintenant énoncer le théorème d'élimination dans les corps de Blum, Cucker, Shub et Smale ([6]) :

Théorème 2 **(P-stabilité de la théorie des corps de caractéristique nulle)**

Soit K un corps, et k un sous-corps de K. Soit Π un problème P_K. Alors la restriction de Π à k est P_k. Le théorème est également vrai pour le cas non-uniforme.

Dans [6], les classes de complexité sont définies à l'aide d'une généralisation des machines de Turing, alors qu'ici elles le sont à l'aide de circuits. Nous adaptons donc la démonstration aux circuits.

Lemme 2 (élimination des paramètres dans un circuit)

Soit K un corps de caractéristique nulle, k un sous-corps de K et soit $(a_1, ..., a_m)$ un uple de paramètres de K. Alors il existe des paramètres de k notés $(b_1, ..., b_r)$ tels que si $A(x_1, ..., x_n, y_1, ..., y_m)$ est un circuit de décision sans paramètre au sens des corps, il existe un circuit au sens des corps $B(x_1, ..., x_n, z_1, ..., z_r)$ sans paramètre tel que $A(\bar{x}, \bar{a})$ et $B(\bar{x}, \bar{b})$ soient équivalents pour des entrées de k. Si nous supposons que les paramètres $\bar{a}$ sont fixés et que nous connaissons les relations qui les lient (les coefficients des éventuels polynômes minimaux), alors B peut-être obtenu à partir de A par un algorithme en temps polynomialement borné par la taille de A.

Démonstration :
Il suffit de considérer $K = k(\bar{a}) = k(a_1)...(a_m)$. De plus, si pour $m = 1$ et $K = k(a)$, et indépendamment de A, nous pouvons éliminer en temps polynomial les paramètres de $k(a)$ en ajoutant des paramètres de k, alors, par récurrence, nous aurons prouvé le théorème pour tout m.

En effet, supposons que a n'appartienne pas à k. Si e est un paramètre de $k(a)$, alors e est une fraction rationnelle de a à coefficients dans k. Nous remplaçons donc l'entrée e par la sortie d'un circuit qui calcule cette fraction, et qui a pour entrées a ainsi que des paramètres de k. Nous obtenons une suite de circuits qui n'a plus e comme paramètre, et qui utilise la division.

Remarquons que tout circuit de décision sur le corps K, mais qui comporte en outre des portes de division, est équivalent à un circuit sur l'anneau K de taille comparable, et qui bien sur n'utilise pas la division . En effet, toutes les portes d'un circuit arithmétique avec division calculent des fractions rationnelles des entrées. Nous pouvons dédoubler chaque porte p en un couple de portes (p_1, p_2) qui calculera respectivement le numérateur et le dénominateur, et simuler les opérations arithmétiques sans plus utiliser la division. Par exemple, si la porte r est étiquetée par une division et reçoit des flèches des portes p et q alors, dans le nouveau circuit, la porte r_1 est étiquetée par une multiplication et reçoit des flèches des portes p_1 et q_2, et la porte r_2 est étiquetée par une multiplication et reçoit des flèches des portes p_2 et q_1. Pour les portes de test, il suffit de tester si le numérateur est nul. Cette technique reste valable dans le cas des circuits sur le corps différentiel K. Le circuit obtenu a au plus six fois plus de portes que le circuit de départ. Par la suite, il n'y aura plus de porte de division dans les circuits, et nous écrirons indifféremment circuit sur un corps ou sur un anneau.

Nous pouvons donc remplacer tous les paramètres de $k(a)$ par a et des paramètres de K, en multipliant la taille du circuit par six. Il ne nous reste maintenant plus qu'à éliminer a.

Lorsque nous voulons connaître la valeur de la sortie de $A(\bar{x}, a)$ sur une entrée donnée, nous devons tester si certains polynômes calculés au fur et à mesure par le circuit sont nuls en $(\bar{x}, a)$. Ces polynômes peuvent être considérés comme des polynômes en a dont les coefficients sont des polynômes à coefficients entiers de $\bar{x}$.

Si a est transcendant sur k, un polynôme P à coefficients dans k est nul en a si et seulement s'il est identiquement nul. Le théorème des témoins nous donne alors une méthode pour tester la nullité de ce polynôme sans utiliser a. Nous l'appliquons avec $s = 1$, τ est la taille de A, $x_1 ... x_r$ sont les entrées de A autres que y et $N = 2^{4(r+1)\tau^2+4}$. Nous commençons par former le circuit qui calcule 2^N : c'est un circuit arithmétique, qui a une entrée étiquetée par le paramètre 2, et $(4(r+1)\tau^2 + 4)$ portes étiquetées par des multiplications. La première reçoit ses deux flèches de l'entrée, dont elle calcule le carré. La deuxième reçoit ses deux flèches de la première, et ainsi de suite jusqu'à la dernière, qui calcule donc 2^N. Nous formons de la même manière les circuits qui calculent $\{x_1^N, ..., x_r^N\}$. Nous obtenons un circuit total de taille $(r+1)(4(r+1)\tau^2+5)$, donc polynomiale en τ. Or P est identiquement nul si et seulement s'il s'annule sur $\{2^N, x_1^N, ..., x_r^N\}$. Nous remplaçons dans le circuit A l'entrée y par la sortie du circuit qui calcule 2^N. Nous obtenons un circuit qui teste si P s'annule en 2^N. Avec r copies de A, nous testons de même si P s'annule en $\{x_1^N, ..., x_r^N\}$. Avec r portes supplémentaires, nous multiplions ces $(r+1)$ tests, et nous obtenons un test équivalent au test $P(a)=0$ sans utiliser le paramètre a dans le circuit. Nous remplaçons ensuite les flèches issues des portes qui testent la nullité de P en a dans les $(r+1)$ copies de A par des flèches issues de la porte qui calcule le produit. Nous faisons cela pour toutes les portes de test du circuit A. Nous obtenons finalement de manière algorithmique et en temps polynomial un circuit équivalent et qui n'utilise que des paramètres de k. Sa taille est bornée par un polynôme en τ^4.

Si a est algébrique sur k, un polynôme P est nul en a si et seulement s'il est multiple du polynôme minimal M de a, ou encore si et seulement si le reste de la division euclidienne de P par M est le polynôme identiquement nul. Si ce reste est donné comme la suite de ses coefficients, qui sont des polynômes à coefficients entiers dont les variables sont les entrées autres que y, tester la nullité de P en a revient à tester si tous ces coefficients sont nuls. De plus, le nombre de ces coefficients est borné par le degré t de M. Dans le circuit A, nous allons remplacer chaque porte qui n'est pas un test, et qui donc calcule un polynôme Q, par un t-uple de portes, qui calcule les coefficients du reste de la division euclidienne de Q par M. Les additions deviennent des additions terme à terme des t-uples, et pour la multiplication, nous n'avons besoin que de connaître les t-uples qui représentent $y^t ... y^{2t-2}$, et donc de nouveaux paramètres de k. Les tests sont remplacés par t tests suivis de la multiplication des résultats. La taille du nouveau circuit est bornée linéairement par τ, et il est clair que nous avons bien décrit ici un algorithme polynomial.

$\square$

Démonstration du théorème 2 (P-stabilité de la théorie des corps de caractéristique nulle) : Si Π est un problème $\mathbb{P}_K$ (resp. P_K), il existe une suite de circuits de décision au sens des corps $(A_n(x_1, ..., x_n, y_1, ..., y_m))_{n\geq 1}$ sans paramètre, de tailles polynomiales en n (resp. produits par un algorithme travaillant en temps polynomial en n) et des éléments $\bar{a} = (a_1, ..., a_m)$ de K tels que pour toute suite $\bar{x} = (x_1, ..., x_n)$ d'éléments de K, $\bar{x}$ appartienne à Π si et seulement si $A_n(\bar{x}, \bar{a})$ calcule 1.

En appliquant le lemme précédent à chaque circuit A_n, nous obtenons de manière polynomiale une suite de circuits équivalents sur k et n'utilisant que des paramètres de k, et nous avons donc montré le résultat.

□

Une conséquence de ce théorème est la suivante : si k est un sous-corps algébriquement clos du corps algébriquement clos K, si Π est un problème P_K définissable à paramètres dans k, alors Π est P_K à paramètres dans k, i.e. $P_K(k)$.

En effet, soit p un polynôme, $(A_n(x_1, ..., x_n, \bar{a}))_{n\geq 1}$ une suite de circuits au sens du corps K de taille bornée par p, et $(B_n(x_1, ..., x_n, \bar{b_n}))_{n\geq 1}$ une suite de circuits au sens du corps K vérifiant : le uple $\bar{x}$ de longueur n appartient à Π si et seulement si $A_n(\bar{x}, \bar{a})$ calcule 1 et si et seulement si $B_n(\bar{x}, \bar{b_n})$ calcule 1, et $K \vDash \forall \bar{x}[A_n(\bar{x}, \bar{a}) \leftrightarrow B_n(\bar{x}, \bar{b_n})]$. La restriction Π_k de Π à k est l'ensemble des uples de k acceptés par les B_n. D'après le théorème précédent, Π_k est aussi P_k : grâce à l'élimination des paramètres dans les circuits A_n, nous obtenons une suite $C_n(\bar{x}, \bar{c})$ de circuits polynomiaux à paramètres dans k qui décide de Π_k. Nous avons : $k \vDash \forall \bar{x}[C_n(\bar{x}, \bar{c}) \leftrightarrow B_n(\bar{x}, \bar{b_n})]$ et donc $K \vDash \forall \bar{x}[C_n(\bar{x}, \bar{c}) \leftrightarrow B_n(\bar{x}, \bar{b}_n)]$ et $K \vDash \forall \bar{x}[A_n(\bar{x}, \bar{a}) \leftrightarrow C_n(\bar{x}, \bar{b}_n)]$. Le problème Π est donc P_K à paramètres dans k grâce aux circuits C_n.

Koiran ([18]) généralise ce résultat pour un sous-corps k quelconque, non nécessairement algébriquement clos : soit K un corps algébriquement clos de caractéristique nulle et k un sous-corps de K. Si Π est un problème P_K définissable à paramètres dans k, alors Π est P_K à paramètres dans k. En particulier, ceci implique que si HN_K est P_K, alors il l'est sans paramètre. Le problème HN_K est le Nullstellensatz de Hilbert dans le corps K : un système fini d'équations et d'inéquations polynomiales à coefficients dans K a-t-il une solution dans K ? Ce problème est NP_K-complet, c'est-à-dire qu'il est NP_K et qu'il est au moins aussi difficile à résoudre que tous les autres problèmes NP_K. Si un problème NP_K-complet est P_K, alors $P_K = NP_K$. Nous connaissons peu de problèmes NP-complets en dehors du cas standard. Par exemple, le problème de la majoration de la dimension d'une variété algébrique de $\mathbb{C}$ est $NP_{\mathbb{C}}$-complet ([20]).

Une autre conséquence du lemme d'élimination est la proposition suivante ([6]) :

Proposition 3 *La question $P = NP$? a la même réponse dans tous les corps algébriquement clos de caractéristique nulle. Il en est de même pour la question $\mathbb{P} = \mathbb{NP}$?.*

Démonstration de la proposition 7 : Comme deux corps quelconques ont une extension commune, il suffit de montrer que la réponse est la même pour deux corps algébriquement clos tels que : $k \subset K$.

Montrons tout d'abord que si $P_K = NP_K$, alors $P_k = NP_k$. Soit Π un problème NP_k. il existe un polynôme p, un entier q, des éléments $\bar{c} = (c_1, ..., c_q)$ de k, une suite de circuits $(C_n(y_1, ..., y_{p(n)}, x_1, ..., x_n, z_1...z_q))_{n\geq 1}$ au sens des corps sans paramètre, obtenue en temps polynomial par un algorithme standard, tels que, pour tout uple $\bar{t} = (t_1, ..., t_n)$ d'éléments de k, $\bar{t}$ appartienne à Π si et seulement s'il existe un uple $\bar{b} = (b_1, ..., b_{p(n)})$ d'éléments de k, tels que k satisfasse $C_n(\bar{b}, \bar{t}, \bar{d})$. Soit Π_K l'ensemble des uples $\bar{t} = (t_1, ..., t_n)$ d'éléments de K tels qu'il existe un uple $\bar{b} = (b_1, ..., b_{p(n)})$ d'éléments de K, et K satisfait $C_n(\bar{b}, \bar{t}, \bar{c})$. Par définition, Π_K est NP_K, donc P_K. Or Π est la restriction de Π_K à k. Donc, d'après le lemme de réduction, nous pouvons éliminer les paramètres de K et Π est P_k. Nous avons montré que $P_k = NP_k$.

Montrons la réciproque. Pour cela, faisons une remarque : si Π est un problème $D_K(k)$, c'est-à-dire définissable à paramètre dans k, et si la restriction Π_k de Π à k est P_k, alors Π est $P_K(k)$, donc en particulier P_K. En effet, soit $(\phi_n(\bar{x}, \bar{a_n}))$ une suite de formules à paramètres dans k qui décide de Π. Alors, cette suite décide aussi de Π_k dans k. Soit $(B_n(\bar{x}, \bar{b}))$ une suite de circuits au sens du corps k, de taille polynomiale et qui décide de Π_k. Pour tout entier n, le corps k vérifie la formule $\forall \bar{x}[\phi_n(\bar{x}, \bar{a}_n) \leftrightarrow B_n(\bar{x}, \bar{b})]$. Comme le corps K est une extension élémentaire de k, il vérifie aussi cette formule du langage des corps, et la suite $(B_n(\bar{x}, \bar{b}))$ décide de Π. Donc Π est $P_K(k)$.

Soit SAT_K le problème de la satisfaisabilité des formules à paramètres dans K. C'est l'ensemble des mots $\bar{m}\bar{a}$ sur K tels que $\bar{m}$ soit un mot booléen codant une formule du langage des corps $\phi(\bar{x}, \bar{y})$ et K satisfasse $\exists \bar{x} \phi(\bar{x}, \bar{a})$. Comme il est NP-complet, $P = NP$ dans un modèle donné si et seulement si SAT est P (cf. [27]). Supposons que $P_k = NP_k$ et que SAT_k soit P_k^d. Comme SAT est définissable sans paramètre, et que la restriction de SAT_K à k est SAT_k, nous en déduisons que SAT_K est P_K, et donc que $P_K = NP_K$. Comme le problème SAT est également $\mathbb{NP}$-complet, la démonstration est la même dans le cas non-uniforme. $\square$

2.1.3 L'élimination des paramètres dans les corps pour la classe $P/poly$

Nous pouvons montrer l'élimination pour les problèmes $P/poly$ sans utiliser le lemme des témoins :

Théorème 4 *Soit K un corps, et k un sous-corps algébriquement clos de K. Soit Π un problème $P_K/poly$. Alors, sa restriction Π_k à k est $P_k/poly$.*

Nous commençons par supposer que le sous-corps k est ω-saturé, ce qui implique en particulier qu'il est algébriquement clos.

Lemme 3 *Soit K un corps, et k un sous-corps ω-saturé de K. Soit Π un problème $P_K/poly$. Alors, sa restriction Π_k à k est $P_k/poly$.*

Démonstration : Soit p_1 et p_2 des polynômes, $(A_n(x_1, ..., x_n, y_1, ..., y_{p_2(n)}))$ une suite de circuits au sens des corps de taille bornée par le polynôme p_1, et $(\bar{a}_n)$ une suite de uples d'éléments de K, avec pour tout n : $\bar{a}_n$ est de longueur $p_2(n)$ et $\bar{x} = (x_1, ..., x_n)$ appartient à Π si et seulement si K satisfait $A_n(\bar{x}, \bar{a}_n)$.

Fixons n. Posons $\bar{a}_n = (a_1, ..., a_{p_2(n)})$. Si tous les a_i sont algébriques sur k, alors ce sont des paramètres de k car k est algébriquement clos, et le circuit $A_n(\bar{x}, \bar{a}_n)$ décide de Π_k pour les entrées de longueur n. Sinon, supposons que a_1 soit transcendant sur k.

Considérons l'ensemble des polynômes à n variables et à coefficients dans k qui annulent $\bar{a}_n$. Soient $\bar{\alpha}$ les coefficients d'un système fini de polynômes qui l'engendre, et $k_0 = \mathbf{Q}(\bar{\alpha})$. Comme k est ω-saturé, il existe un uple $\bar{b}_1$ de k de longueur $p_2(n)$ qui réalise le type de $\bar{a}_n$ sur k_0, c'est à dire qui satisfait les mêmes formules du langage des corps à paramètres dans k_0. Nous définissons ainsi par récurrence des uples $\bar{b}_2, ..., \bar{b}_{2n+1}$ de k de longueur $p_2(n)$ tels que pour tout i compris entre 1 et $2n$, le type de $\bar{b}_{i+1}$ et celui de $\bar{a}_n$ sur $(\bar{\alpha}, \bar{b}_1, ..., \bar{b}_i)$ soient égaux. En particulier, $\bar{b}_{i+1}$ est transcendant sur $\mathbf{Q}(\bar{\alpha}, \bar{b}_1, ..., \bar{b}_i)$ et le degré de transcendance de $\mathbf{Q}(\bar{\alpha}, \bar{b}_1, ..., \bar{b}_{2n+1})$ est au moins $2n+1$.

Comme le degré de transcendance de $\mathbf{Q}(\bar{x})$ est au plus n, il existe au moins $n+1$ indices i compris entre 1 et $2n+1$ tels que les $\bar{b}_i$ soient de degré de transcendance au moins 1 sur $\mathbf{Q}(\bar{x})$. Pour ces indices, le type de $\bar{b}_i$ et celui de $\bar{a}_n$ sur $\mathbf{Q}(\bar{x})$ sont égaux, et donc $A_n(\bar{x}, \bar{a}_n)$ est équivalent à $A_n(\bar{x}, \bar{b}_i)$. Donc, $A_n(\bar{x}, \bar{a}_n)$ calcule 1 si et seulement si au moins $n+1$ des circuits $A_n(\bar{x}, \bar{b}_i)$ calculent 1.

A partir de A_n, nous formons un circuit équivalent pour les entrées $\bar{x}$ de k dont les paramètres sont $\bar{b}_1, ..., \bar{b}_{2n+1}$. Pour cela, nous commençons par former les circuits $A_n(\bar{x}, \bar{b}_1), ..., A_n(\bar{x}, \bar{b}_{2n+1})$. Puis à l'aide d'un circuit booléen, nous comptons le nombre de réponse positives, et s'il est au moins égal à $n+1$ nous acceptons. □

Démonstration du théorème 4 : Il nous reste à traiter le cas où k n'est pas ω-saturé. Reprenons les notations de la démonstration précédente. Soit k_1 une extension ω-saturée de k telle que $k_1 \cap K = k$. L'idéal annulateur de $\bar{a}_n$ a un système de générateurs identique sur k et sur k_1. En appliquant le lemme précédent, nous montrons qu'il existe des uples $\bar{b}_1, ..., \bar{b}_{2n+1}$ de k_1 de longueur $p_2(n)$ tels que pour une entrée $\bar{x}$ de k_1, $A_n(\bar{x}, \bar{a}_n)$ calcule 1 si et seulement si au moins $n+1$ des circuits $A_n(\bar{x}, \bar{b}_i)$ calculent 1. Or le type de $\bar{a}_n$ est définissable à paramètres dans k_0. Donc il existe une formule $\phi(\bar{x}, \bar{c})$ à paramètres $\bar{c}$ dans k_0 telle que k_1 satisfasse $\phi(\bar{x}, \bar{c}) \leftrightarrow A_n(\bar{x}, \bar{a}_n)$ pour tout $\bar{x}$, et donc satisfasse également la formule :

$$\exists y_1 ... \exists y_{2n+1} \forall \bar{x} [\phi(\bar{x}, \bar{c}) \leftrightarrow \bigvee_{1 \leq i_0 < i_1 < ... < i_{n+1} \leq 2n+1} (A_n(\bar{x}, \bar{y}_{i_0}) \wedge ... \wedge A_n(\bar{x}, \bar{y}_{i_{n+1}}))]$$

Comme cette formule est à paramètres dans k_0 et comme k est un corps algébriquement clos, il vérifie aussi cette formule, et il existe des uples $\bar{e}_1, ..., \bar{e}_{2n+1}$ d'éléments de k tels qu'un uple $\bar{x}$ de k appartienne à Π_k si et seulement si

$A_n(\bar{x}, \bar{e}_i)$ calcule 1 pour au moins $n+1$ indices i compris entre 1 et $2n+1$. Il suffit de faire les $2n+1$ calculs, puis de compter le nombre de réponses positives. □

Remarque : comme le problème SAT est $NP/poly$-complet, une démonstration similaire à celle du paragraphe précédent nous montre que la question $P/poly = NP/poly$? a la même réponse dans tous les corps algébriquement clos de caractéristique nulle.

2.2 L'élimination des paramètres dans les corps différentiels

Dans le premier paragraphe, nous donnons les propriétés des corps différentiels qui nous sont utiles. Puis, dans le deuxième, nous énonçons et nous montrons un lemme qui sera central dans la démonstration de l'élimination des paramètres : le lemme de réduction. Il nous permet de transformer un circuit qui a des portes de dérivation en un circuit dont les seules dérivations sont les dérivations successives des entrées, et ceci de manière algorithmique et polynomiale. Dans les troisième et quatrième paragraphes, nous montrons l'élimination des paramètres. Puis, dans le paragraphe suivant, nous en tirons quelques conséquences, dont la comparaison de la question $P = NP$? pour les corps algébriquement clos de caractéristique nulle et les corps différentiellement clos de caractéristique nulle . Enfin, dans le dernier paragraphe, nous étudions deux problèmes particuliers, dont nous montrons qu'ils ne sont pas accélérés par la dérivée.

2.2.1 Les corps différentiels

Les définitions et propriétés des corps différentiels peuvent être trouvés dans les livres de Kolchin ([21]) et Poizat ([26]), et dans l'article de Wood ([31]).

Un **anneau différentiel** est un anneau commutatif $(K, 0, 1, +, -, \times)$ muni d'une dérivation d, c'est-à-dire d'une opération interne de K dans K qui vérifie, pour tout couple (x, y) d'éléments de K : $d(x+y) = dx + dy$ et $d(xy) = xdy + ydx$. Si K est un corps, K est appelé **corps différentiel** . Dans la suite, nous considérons uniquement des corps et des corps différentiels de **caractéristique nulle**. Les éléments de dérivée nulle de K sont appelées **constantes** . Ils forment un sous-corps de K. C'est un ensemble définissable au sens de la théorie des corps différentiels. Remarquons que tout corps peut être muni d'une structure de corps différentiel : il suffit de prendre la dérivée nulle partout. Pour noter les dérivées successives, nous écrirons selon les cas $d^n x$ ou $x^{(n)}$.

Définissons l'anneau différentiel des **polynômes différentiels** en une variable $K[X]_d$. C'est l'anneau $K[X, X^{(1)}, ..., X^{(n)}, ...]$ des polynômes à coefficients dans K et à une infinité dénombrable de variables $X^{(n)}$. La dérivée prolonge celle de K. Elle est définie de manière unique par les relations : $dX^{(n)} = X^{(n+1)}$. Nous définissons de manière analogue le corps différentiel $K(X)_d$ des **fractions rationnelles différentielles** et ceux des polynômes différentiels en plusieurs variables. A chaque polynôme différentiel $P(X)$ est associée naturellement une fonction polynôme différentielle. L'image d'un élément a de K par la fonction associée à $X^{(n)}$ est $a^{(n)}$.

Un sous-ensemble I d'un anneau différentiel est appelé **idéal différentiel premier** si c'est un idéal premier stable par dérivation. Si c est un élément du sur-corps différentiel K du corps différentiel k, le sous-ensemble I_c de $k[X]_d$ constitué des polynômes différentiels qui annulent c est un idéal différentiel premier de $k[X]_d$. Soit P un polynôme différentiel en une variable X. Nous appelons **idéal différentiel engendré par** P, noté $(P)_d$, l'idéal de l'anneau des polynômes différentiels $K[X]_d$ engendré par P et ses dérivées successives.

L'**ordre** d'un polynôme différentiel non nul $P(X)$ est le plus grand indice i tel que $X^{(i)}$ apparaisse avec un degré au moins un dans $P(X)$. Si r est l'ordre de $P(X)$, son **degré** est son degré partiel en $X^{(r)}$. Si son ordre est r et son degré est v, son **niveau** est le couple (r, v). Les niveaux sont ordonnés par l'ordre lexicographique sur les couples d'entiers. Par convention, le polynôme nul est de niveau strictement inférieur à tous les autres polynômes différentiels.

Si K est un corps différentiel de caractéristique nulle, si k est un sous corps différentiel de K, un élément c de K sera dit **différentiellement transcendant**, ou d-transcendant sur k s'il n'annule aucun polynôme différentiel non nul à coefficients dans k. Dans le cas contraire, c est appelé **différentiel-**

lement algébrique , ou d-algébrique sur k. Un polynôme différentiel M de niveau minimal et irréductible qui annule c est alors appelé **polynôme minimal** de c. Si M est d'ordre r, alors $c, c^{(1)}, \ldots, c^{(r-1)}$ sont algébriquement indépendants sur k.

Si M est un polynôme différentiel non nul d'ordre r et de degré v, la **séparante S de M** est le polynôme différentiel $\partial M / \partial X^{(r)}$, de niveau strictement inférieur au niveau de M. Le **coefficient initial** Δ de M est le coefficient dominant de M vu comme un polynôme en $X^{(r)}$. C'est un polynôme différentiel d'ordre strictement inférieur à r. Nous avons la propriété suivante :

Proposition 5 *Soit M un polynôme différentiel de niveau (r, v), de séparante S et de coefficient initial Δ. Pour tout polynôme différentiel P, il existe des entiers i et j, des polynômes différentiels Q d'ordre au plus r et R de niveau strictement inférieur à (r, v) tels que $S^i P$ soit congru à Q modulo $(M)_d$ et $\Delta^j Q$ soit congru à R modulo (M).*

Démonstration : Nous pouvons facilement montrer par récurrence que pour tout entier n strictement positif, il existe un polynôme différentiel M_{p+n-1} d'ordre au plus $(r+n-1)$ tel que :

$$M^{(n)}(X) = X^{(r+n)} S(X) + M_{r+n-1}(X)$$

Le terme de plus grand ordre de $M^{(n)}$ est exactement $X^{(r+n)} S(X)$.

Supposons P d'ordre $(r+n)$, où n est un entier strictement positif. Soit I le coefficient initial de P :

$$P(X) = X^{(r+n)} I(X) + P_{r+n-1}(X)$$

où P_{r+n-1} est un polynôme différentiel d'ordre au plus $(r+n-1)$. Nous avons donc :

$$S(X)P(X) = I(X)M^{(n)}(X) + S(X)P_{r+n-1}(X) - I(X)M_{r+n-1}(X)$$

ce qui entraîne que :

$$S(X)P(X) \equiv S(X)P_{r+n-1}(X) - I(X)M_{r+n-1}(X) \mod (M)_d$$

Le niveau de $S(X)P_{r+n-1}(X) - I(X)M_{r+n-1}(X)$ est au plus $(r+n-1)$. En itérant le procédé au plus n fois, nous montrons qu'il existe un entier i

(inférieur ou égal à n) et un polynôme différentiel Q d'ordre au plus r tel que $S^i P$ soit congru à Q modulo $(M)_d$.

Supposons maintenant que Q soit de niveau $(r, v+n)$, où n est un entier positif, et que son coefficient dominant soit $C(X)$. Nous avons :

$$M(X) = \Delta(X) X^{(r)^v} + m_{v-1}(X)$$

et

$$Q(X) = C(X) X^{(r)^{v+n}} + Q_{v+n-1}(X)$$

où m_{v-1} est de niveau au plus $(r, v-1)$ et Q_{v+n-1} de niveau au plus $(r, v+n-1)$. Donc :

$$\Delta(X) Q(X) = C(X) X^{(v)^n} M(X) + \Delta(X) Q_{v+n-1}(X) - C(X) X^{(v)^n} m_{v-1}(X)$$

et

$$\Delta(X) Q(X) \equiv \Delta(X) Q_{v+n-1}(X) - C(X) X^{(v)^n} m_{v-1}(X) \quad \bmod (M)$$

Le polynôme différentiel $\Delta(X) Q_{v+n-1}(X) - C(X) X^{(v)^n} m_{v-1}(X)$ est de niveau au plus $(r, v+n-1)$. Nous pouvons recommencer si $n-1$ est positif ou nul, et nous avons donc montré qu'il existe un entier j et un polynôme différentiel R de niveau strictement inférieur à (r, v) tels que $\Delta^j Q$ soit congru à R modulo (M). $\square$

En particulier, il existe un polynôme différentiel S_i de niveau strictement inférieur à (r, v) et un entier l tels que $\Delta^l S^i$ soit congru à S_i modulo (M). Cette propriété nous permet de caractériser I_c. Soit c un élément du corps différentiel K d-algébrique sur le sous-corps différentiel k, M un polynôme minimal de c sur k, de séparante S et de coefficient initial Δ. La proposition précédente nous montre que l'ensemble I_c des polynômes de $k[X]_d$ qui annulent c est égal à l'ensemble des polynômes P de $k[X]_d$ pour lesquels il existe des entiers i et j tels que $S^i \Delta^j P$ appartienne à $(M)_d$. Si $S^i \Delta^j P$ est congru modulo $(M)_d$ au polynôme différentiel R de niveau strictement inférieur à M, pour tester si P est nul en c, il suffit de tester si R identiquement nul, i.e. si les v coefficients de R sont identiquement nuls. C'est pourquoi nous utilisons la représentation d'un polynôme différentiel P quelconque modulo I_c par $2(v+1)$ polynômes différentiels d'ordre au plus $(r-1)$: Δ^j, Δ^l, les v

coefficients de R et les v coefficients de S_i vus comme des polynômes en $X^{(r)}$ de degré au plus $(v-1)$. Ainsi, si nous connaissons les polynômes différentiels qui représentent ${X^{(r)}}^v \ldots {X^{(r)}}^{2v-2}$ modulo $(M)_d$, nous pouvons sommer et multiplier des polynômes différentiels P_1 et P_2 modulo I_c en effectuant un nombre fixe de multiplications et d'additions des polynômes différentiels qui représentent P_1, P_2 et ${X^{(r)}}^v \ldots {X^{(r)}}^{2v-2}$. D'autre part, la dérivation de M nous permet de trouver les représentants de $X^{(r+1)}$ modulo $(M)_d$. En dérivant plusieurs fois M, nous calculons par récurrence les dérivées successives de $X^{(r)}$. Le nombre d'opérations que nous devons effectuer sur les représentants dépend polynomialement du nombre de dérivées que nous voulons obtenir.

Il existe un analogue de la clôture algébrique pour les corps : c'est la clôture différentielle pour les corps différentiels. Robinson ([30]) a défini la théorie des corps différentiellement clos de caractéristique 0 comme la modèle-complétion de la théorie des corps différentiels de caractéristique 0. Dans [4], Lenore Blum en donne une axiomatisation. Un corps K est **différentiellement clos** si pour tous polynômes différentiels P et Q tels que Q soit d'ordre strictement inférieur à P, il existe un élément x de K qui annule P mais pas Q. De plus, tout corps K possède une clôture différentielle, i.e. un sur-corps $\bar{K}$ différentiellement clos tel que pour tout sur-corps différentiellement clos L de K, $\bar{K}$ soit isomorphe à un sous-corps de L.

2.2.2 Le lemme de réduction

Un circuit sur le corps différentiel K effectue, outre les opérations possibles dans les corps, des dérivations. Dans un premier temps, nous allons transformer le circuit de manière à ce que les dérivations soient les toutes premières opérations effectuées sur les entrées, les opérations suivantes étant celles autorisées dans les corps. Un circuit mis sous cette forme est appelé un **circuit réduit** : c'est un circuit tel que toute porte précédent une dérivation soit ou bien une entrée, ou bien une autre porte de dérivation. Nous pourrons alors dans un deuxième temps éliminer les paramètres de K, en nous ramenant au cas des corps.

Pour dériver plusieurs fois, nous utiliserons les puissances symboliques. Pour cela, nous avons besoin des coefficients binômiaux.

Lemme 4 (calcul du triangle de Pascal) *Il existe un circuit T_m arithmétique de taille au plus $[2+m(m-1)/2]$, ayant pour seule entrée le paramètre 1, calculable par un algorithme standard en temps polynomial, dont les portes calculent les C_j^i pour j compris entre 1 et m, et i compris entre 0 et j.*

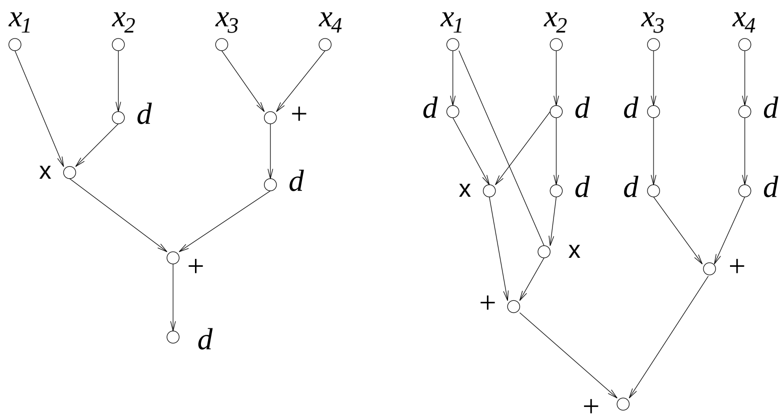
démonstration du lemme 4 :
Le circuit cherché reproduit les calculs du triangle de Pascal, grâce à l'égalité : $C_{i+1}^{j+1} = C_i^j + C_i^{j+1}$. Le circuit T_1 est constitué uniquement de deux entrées, étiquetées 1, qui sont aussi des sorties. A partir du circuit T_m de taille $(2+m(m-1)/2)$, nous construisons le circuit $T_{(m+1)}$ en ajoutant m portes : la première somme 1 et la première sortie de T_m, la deuxième somme la première et la deuxième sortie de T_m, et ainsi de suite jusqu'à la dernière qui somme la dernière sortie de T_m et 1. Le circuit $T_{(m+1)}$ est de taille $[2+m(m-1)/2+m]$, i.e. $[2+m(m+1)/2]$.

□

Remarque : comme le triangle de Pascal est symétrique, il suffit de calculer les C_{2n}^i pour i compris entre 1 et n, et les C_{2n+1}^i pour i compris entre 1 et $n+1$. Nous aurions donc pu trouver un circuit un peu plus petit.

Lemme 5 (réduction d'un circuit) *Tout circuit sur le corps différentiel K, $A(\bar{x})$ de taille τ et de profondeur h, est équivalent à un circuit $B(\bar{x})$ sur le corps différentiel K, réduit, de taille au plus $4\tau^3$ et de profondeur au plus $2h\log_2 h$ (pour τ et h assez grands). De plus, $B(\bar{x})$ peut être obtenu à partir de $A(\bar{x})$ par un algorithme standard en temps polynomial en τ.*

Exemple de réduction d'un circuit simple (arbre) :



Démonstration du lemme 5 :
La **hauteur** d'une porte p d'un circuit C est le nombre de portes du plus long chemin allant de p à une sortie du circuit. Sa **d-hauteur** $\delta(p)$ est le nombre maximal de portes de dérivations sur un chemin allant de p exclue à une sortie s du circuit incluse. La **profondeur** du circuit est le maximum des hauteurs des portes, et sa **d-profondeur** δ est le maximum des d-hauteurs des portes. Si p est une porte d'un circuit C, **le circuit au dessus de** p est le sous circuit de C qui permet de faire le calcul en p, i.e. c'est le sous-circuit de C qui contient p et tel que si la porte q en fasse partie, alors toutes les portes qui émettent une flèche vers q en fassent partie également.

Nous ajoutons sous chaque porte p du circuit $\delta(p)$ portes de dérivations successives. Ce faisant, nous ajoutons de nouvelles sorties. La taille du circuit est au plus multipliée par $(\delta + 1)$, et le nouveau nombre de dérivées est au plus $\delta\tau$. Nous allons transformer ce circuit pas à pas, par le bas, en supprimant les portes de dérivation de d-hauteur nulle qui ne sont pas situées directement sous les entrées, et en ajoutant des portes d'addition et de multiplication, mais en laissant inchangés les circuits au dessus des autres portes de dérivations, ce qui nous permettra de garder un nombre de portes polynomial. Nous commençons par adjoindre le circuit T_δ du lemme 4, qui a $[2 + \delta(\delta - 1)/2]$ portes et est de hauteur $(\delta - 1)$.

Si toutes les portes de dérivation sont précédées soit d'une entrée, soit d'une autre porte de dérivation, alors le circuit est réduit. Sinon, il existe une suite de portes $p_0, ..., p_m$ reliées par des flèches telle que p_0 ne soit ni une porte de dérivation ni une entrée, $p_1, ..., p_m$ soient des portes de dérivations successives de p_0 et m soit au plus égal à $\delta(p_0)$. Montrons qu'alors il existe une telle suite telle que les seules portes de dérivations situées sous une porte de la suite soient les portes de la suite elle-même. Si ce n'est pas le cas, alors il existe un indice i compris entre 1 et m, une porte de dérivation q et un chemin de p_i à q. S'il n'y a que des portes de dérivations entre p_i et q, nous pouvons les confondre avec $p_{i+1}, ..., p_m$, quitte à allonger la suite avec de nouvelles portes de dérivation. Sinon, il existe une porte q_0 située sous p_i qui n'est pas une dérivation, un entier μ et des portes de dérivation $q_1, ..., q_\mu$ tels que les portes de la suite $q_0, ..., q_\mu$ soient reliées par des flèches. Si cette nouvelle suite ne convient pas, nous recommençons le processus. Celui-ci à une fin car la hauteur de q_0 est un entier naturel strictement inférieur à la hauteur de p_0.

Considérons une suite $p_0, ..., p_m$ qui vérifient les conditions précédentes. Si p_0 est un test, alors le résultat est soit 0 soit 1, donc des constantes, et nous remplaçons toutes ses dérivées par le paramètre 0. Si p_0 est une addition, une soustraction ou une multiplication, alors elle reçoit des flèches de deux portes q et r, sous lesquelles il y a respectivement $\delta(q)$ et $\delta(r)$ portes de dérivations successives : $q_1, ..., q_{\delta(q)}$ et $r_1, ..., r_{\delta(r)}$. Si p_0 est une addition (resp. une soustraction), alors pour i compris entre 1 et m, nous remplaçons p_i par une porte d'addition (resp. de soustraction) qui somme q_i et r_i. La profondeur du circuit n'augmente pas.

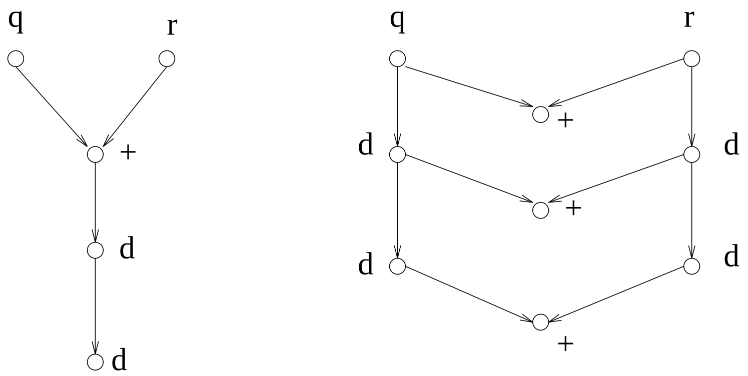


figure : calcul de $d^2(q+r) = d^2q + d^2r$

Si p_0 est une multiplication, alors pour i compris entre 1 et m, nous devons former le circuit qui calcule

$$d^i(p_0) = d^i(qr) = \sum_{j=0}^{i} C_i^j \; d^j(q) \; d^{i-j}(r) = \sum_{j=0}^{i} C_i^j \; q_i \; r_{i-j}$$

Ce circuit a pour entrées les q_i, les r_j et des portes du circuit qui calcule le triangle de Pascal. Il a en outre $2i$ portes de multiplication et i portes d'addition. La profondeur du circuit augmente d'au plus $(\ln(i+1)+1)$.

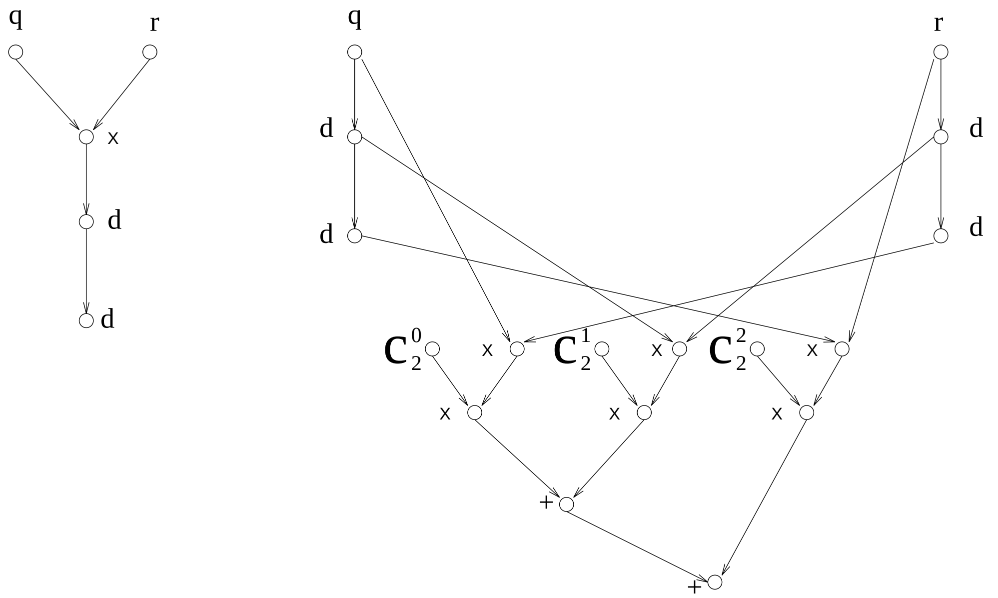


figure : calcul de $d^2(qr) = rd^2q + 2dqdr + qd^2r$

Dans tous les cas, nous pouvons supprimer les portes $p_1, ..., p_m$, et donc diminuer le nombre total de portes de dérivation du circuit. Les d-hauteurs des portes sont invariantes. Pour chaque porte de dérivation du circuit d'origine, et pour chaque porte de calcul que nous devons «remonter», nous créons au plus 3δ portes et la profondeur augmente d'au plus $(1 + \ln(\delta + 1))$. Or δ est majoré à la fois par τ et par h. Donc le circuit obtenu est de taille au plus $[2 + \tau(\tau - 1)/2 + \tau(\tau + 1) + 3\tau^3]$ et de profondeur au plus $2h \log_2 h$.

De cette manière, nous obtenons le circuit $B(\bar{x}, \bar{y})$ par un algorithme standard en temps polynomial en τ.

$\square$

2.2.3 L'élimination des paramètres dans les corps différentiels pour la classe P

A partir de maintenant, K est un corps différentiel, et k est un sous-corps différentiel de K.

Théorème 6 (**P-stabilité de la théorie des corps différentiels de caractéristique nulle**)

La restriction à k d'un problème P_K^d est P_k^d. Le résultat vaut également pour la classe non-uniforme $\mathbb{P}$.

Si Π est un problème P_K^d, il existe une suite de circuits de décision sur le corps différentiel K qui résout le problème Π. Cette suite de circuits utilise des paramètres de K. Ce sont ces paramètres que nous voulons éliminer, de manière à obtenir une suite de circuits sur le corps différentiel k qui décide de la restriction de Π à k.

Maintenant que nous savons mettre un circuit sous forme réduite, nous éliminons les paramètres.

Lemme 6 (élimination des paramètres d'un circuit sur le corps différentiel K)

Si $(a_1, ..., a_m)$ sont des paramètres de K, il existe des paramètres $(b_1, ..., b_r)$ de k tels que tout circuit de décision $A(\bar{x}, \bar{a})$ sur le corps différentiel K soit équivalent pour les entrées de k à un circuit de décision $B(\bar{x}, \bar{b})$ sur le corps différentiel K. De plus, B peut-être obtenu par un algorithme en temps polynomialement borné par la taille de A.

Démonstration :
Nous pouvons considérer le cas où K est le corps différentiel engendré par k et $\bar{a}$, i.e. $K = k(\bar{a})_d$. De même que dans le cas des corps, nous allons montrer que si $K = k(a)_d$, nous pouvons éliminer les paramètres de $k(a)_d$ en ajoutant des paramètres de k, indépendamment du circuit considéré. Nous supposons donc que $\bar{a} = (a, a_1, ..., a_m)$, où $a_1, ..., a_m$ sont des éléments de $k(a)_d$.

Appliquons le lemme 5 de réduction au circuit A : nous obtenons un circuit réduit $C(x_1, ..., x_n, y, y_1, ..., y_m)$, calculable par un algorithme standard en temps polynomial et équivalent à A. Nous supposons que sous chaque entrée de B, il y a au plus δ dérivations successives.

Si a_m n'appartient pas à k mais appartient à $k(a)_d$, alors a_m est égal à une fraction rationnelle différentielle de a à coefficients dans k. Nous remplaçons l'entrée y_m par un circuit à paramètres dans k qui calcule a_m et qui utilise la division. Nous faisons de même pour $a_{m-1}, ..., a_1$. Puis nous dédoublons les portes pour ne plus utiliser la division. Nous obtenons le nouveau circuit par un algorithme standard linéaire, mais le circuit obtenu n'est plus réduit : il faut lui appliquer le lemme de réduction à nouveau avant d'éliminer a.

Si a est d-transcendant sur k, nous pouvons l'éliminer sans ajouter de paramètre de k, en utilisant l'élimination d'un paramètre transcendant pour un corps. En supprimant les flèches entre les entrées et leurs dérivées successives, nous créons de nouvelles entrées. Nous obtenons un circuit au sens des corps, $C(x_1, ..., x_{n(\delta_n+1)}, y_0, ..., y_{\delta_n})$, calculable par un algorithme standard en temps polynomial, et tel que tout uple $\bar{x}$ de K appartienne à Π si et seulement si $C(\bar{x}, \bar{x}^{(1)}, ..., \bar{x}^{(\delta)}, a, a^{(1)}, ..., a^{(\delta)})$ calcule 1. Les paramètres $a, a^{(1)}, ..., a^{(\delta)}$ sont algébriquement indépendants sur k. Nous pouvons donc appliquer l'élimination des paramètres transcendants dans le cas des corps et les éliminer un à un. Ensuite, il suffit de restaurer les flèches de dérivations successives entre les entrées.

Si a est d-algébrique sur k, soit $M(X)$ son polynôme différentiel minimal sur k. Il est de niveau (r, v). Ses coefficients sont des paramètres de k. A priori, il y a δ portes de dérivations successives sous l'entrée y. Nous ne laissons que les $(r-1)$ premières, et nous supprimons les autres. Nous allons calculer modulo I_a dans le circuit.

Pour cela, nous remplaçons toutes les portes p de C par un $2(v+1)$-uple de portes qui calculent les $2(v+1)$ polynômes différentiels de niveau strictement inférieur à (r, v) qui représentent le polynôme différentiel calculé en p modulo I_a. Nous ajoutons les circuits qui calculent $y^{(r)^v} ... y^{(r)^{2v-2}}$ ainsi que $y^{(r+1)}$ modulo I_a. Puis nous effectuons les calculs comme ceci est expliqué dans le paragraphe sur les corps différentiels. Nous commençons par calculer les dérivées de y jusqu'à l'ordre δ, puis nous simulons la suite des calculs. Comment faire lorsque nous avons une porte de test ? Imaginons que nous voulons tester si le polynôme $P(a)$ calculé par la porte p de l'ancien circuit est nul. Nous savons que la réponse est oui si et seulement si R est le polynôme nul. Or parmi les $2(v+1)$ portes qui remplacent p, v portes calculent les coefficients de R, vu comme un polynôme en $y^{(r)}$, qui sont des polynômes différentiels d'ordre au plus $(r-1)$. Il suffit donc, pour tester si R est nul, de tester si les v coefficients sont nuls, et de multiplier les résultats.

D'autre part, comme M est d'ordre r, les éléments $(a, a^{(1)}, ..., a^{(r-1)})$ sont algébriquement indépendants sur k. Pour éliminer a, il suffit de procéder comme dans le cas où a est d-transcendant sur k.

□

Démonstration du théorème 6 :

Soit Π un problème $\mathbb{P}_K^d$ (resp. P_K^d). Alors il existe une suite de circuits au sens des corps différentiels sans paramètre $(A_n(x_1, ..., x_n, y_1, ..., y_m))_{n \geq 1}$, de tailles polynomiales (resp. que nous pouvons obtenir par un algorithme standard polynomial), et un uple $\bar{a} = (a_1, ..., a_m)$ d'éléments de K, tels que $\bar{x} = (x_1, ..., x_n)$ appartienne à Π si et seulement si $A_n(\bar{x}, \bar{a})$ calcule 1. Nous appliquons le lemme précédent à la suite de circuits, et nous obtenons par un algorithme polynomial une suite de circuits de décision sur le corps différentiel k équivalents.

□

2.2.4 L'élimination des paramètres dans les corps différentiels pour la classe $P/poly$

Elle est une conséquence directe de l'élimination des paramètres dans les corps pour la classe $P/poly$.

Corollaire 1 *Soit K un corps différentiel et k un sous-corps différentiel algébriquement clos de K. Soit Π un problème $P_K^d/poly$. Alors la restriction Π_k de Π à k est $P_k^d/poly$.*

Démonstration : Il existe des polynômes p_1 et p_2, une suite de circuits $(A_n(x_1, ..., x_n, y_1, ..., y_{p_2(n)}))$ au sens des corps différentiels de taille bornée par le polynôme p_1, et $(\bar{a}_n)$ une suite de uple d'éléments de K, telle que pour tout n : a_n soit de longueur $p_2(n)$ et $\bar{x} = (x_1, ..., x_n)$ appartienne à Π si et seulement si K satisfait $A_n(\bar{x}, \bar{a}_n)$. Nous pouvons supposer que les circuits A_n soient sous forme réduite. En supprimant les portes de dérivation pour créer de nouvelles entrées, nous obtenons une suite de circuits au sens des corps $(B_n(\bar{x}, \bar{x}_1, ..., \bar{x}_{p_1(n)}, \bar{a}_n, ..., \bar{a}_n^{(p_1(n))}))$ telle que pour tout n-uple $\bar{b}$ de K, $B_n(\bar{b}, ..., \bar{b}^{(p_1(n))}, \bar{a}_n, ..., \bar{a}_n^{(p_1(n))})$ calcule 1 si et seulement si $A_n(\bar{b}, \bar{a}_n)$ calcule 1. Nous pouvons appliquer l'élimination pour la classe $P/poly$ dans le cas des corps à la suite (B_n). Nous obtenons une suite de circuits au sens des corps à paramètres dans k, et il suffit de restaurer les dérivations des entrées $\bar{x}$ pour obtenir une suite de circuits qui nous montre que Π_k est $P_k^d/poly$. $\square$

2.2.5 Conséquences

D'après [6], la question $P = NP$? a la même réponse dans tous les corps algébriquement clos de caractéristique nulle. Le théorème est également vrai pour les corps différentiellement clos de caractéristique nulle.

Proposition 7 *La question $P = NP$? a la même réponse dans tous les corps différentiellement clos de caractéristique nulle. Il en est de même pour la question $\mathbb{P} = \mathbb{NP}$?, et pour la question $P/poly = NP/poly$.*

La démonstration de la proposition est identique à celle de la proposition 3 : Comme deux corps différentiels quelconques ont une extension commune, il suffit de montrer que la réponse est la même pour deux corps différentiellement clos tels que : $k \subset K$.

Le lemme d'élimination des paramètres pour les corps différentiels nous permet de montrer que si $P_K^d = NP_K^d$, alors $P_k^d = NP_k^d$. Pour la réciproque, nous remarquons tout d'abord que si Π est un problème $D_K^d(k)$, c'est-à-dire définissable à paramètre dans k, et si la restriction Π_k de Π à k est P_k^d, alors Π est $P_K^d(k)$, donc en particulier P_K^d. D'autre part, soit SAT_K^d le problème de la satisfaisabilité des formules du langage des corps différentiels à paramètres dans K. Comme il est NP_K^d-complet, $\mathbb{NP}_K^d$-complet et $NP_K^d/poly$-complet cela nous permet de conclure. $\square$

Proposition 8 *Si $P = NP$ pour les corps différentiellement clos de caractéristique nulle K, alors $P = NP$ pour les corps algébriquement clos de caractéristique nulle. Ceci est encore vrai pour la relation entre classes non-uniformes $\mathbb{P} = \mathbb{NP}$, et pour $P/poly = NP/poly$.*

Démonstration de la proposition 8 : Soit k un corps algébriquement clos de caractéristique nulle, et soit Π un problème NP_k. Il existe un polynôme p, un entier q, des éléments $\bar{c} = (c_1, ..., c_q)$ de k, une suite de circuits

$(C_n(y_1, ..., y_{p(n)}, x_1, ..., x_n, z_1, ..., z_q))_{n\geq 1}$ au sens des corps sans paramètre, obtenue en temps polynomial par un algorithme standard, tels que, pour tout uple $\bar{t} = (t_1, ..., t_n)$ d'éléments de k, $\bar{t}$ appartienne à Π si et seulement s'il existe un uple $\bar{b} = (b_1, ..., b_{p(n)})$ d'éléments de k, tels que k satisfasse $C_n(\bar{b}, \bar{t}, \bar{c})$.

Nous munissons k de la structure de corps différentiel en prenant une dérivée nulle sur k. Soit K la clôture différentielle de k ; k est le corps des constantes de K. Soit Π_K l'ensemble des uples $\bar{t} = (t_1, ..., t_n)$ d'éléments de K tels qu'il existe un uple $\bar{b} = (b_1, ..., b_{p(n)})$ d'éléments de K, et K satisfait $C_n(\bar{b}, \bar{t}, \bar{c})$. Par définition, Π_K est NP_K, donc NP^d_K. Nous en déduisons, d'après l'hypothèse, que Π_K est P^d_K. Le problème Π est la restriction de Π_K à k ; en effet, dans le langage des corps, K est une extension élémentaire de k. Donc, d'après le théorème 6, nous pouvons éliminer les paramètres de K. De plus, comme tous les éléments de k sont de dérivée nulle, nous pouvons remplacer toutes les portes de dérivation du circuit par 0. Donc Π est P_k. Nous avons montré que $P_k = NP_k$, et donc que $P = NP$ pour tous les corps algébriquement clos de caractéristique nulle. Les démonstrations des cas non-uniforme et $P/poly$ sont exactement les mêmes. $\square$

Intéressons nous maintenant aux problèmes booléens, c'est-à-dire aux ensembles de suites finies de 0 et de 1. Les opérations sur les corps et les corps différentiels permettent elles de résoudre plus vite ces problèmes?

Définition 6 *Si $\mathcal{C}$ est une classe de complexité sur la structure M, l'ensemble des problèmes booléens de $\mathcal{C}$, noté $BP(\mathcal{C})$, est l'ensemble des problèmes de $\mathcal{C}$ constitués uniquement de mots booléens, i.e. de suites finies de 0 et de 1.*

Nous savons ([27]) que les structures localement finies ont les mêmes problèmes booléens P et $\mathbb{P}$ que la structure standard. De même pour la structure $(\mathbf{R}, 0, 1, +, -, =)$. Les problèmes booléens P au sens de $(\mathbf{R}, 0, 1, +, -, =, <)$, qui sont exactement les problèmes booléens $\mathbb{P}$ au sens de $(\mathbf{R}, 0, 1, +, -, =, <)$ sont les $\mathbb{P}$ au sens standard ([17]).
Les problèmes booléens $\mathbb{P}$ au sens de $(\mathbf{K}, 0, 1, +, \times, =)$, où K est un corps de caractéristique nulle sont $\mathbb{P}$ au sens standard ([16]). En effet, l'élimination des paramètres dans les corps de caractéristique nulle permet de montrer qu'un

problème booléen P_K, ou $\mathbb{P}_K$, l'est sans paramètre. Il suffit pour cela d'éliminer les paramètres dans le corps $\mathbf{Q}$. Les paramètres rationnels peuvent alors être remplacés par des circuits sans paramètre qui les calculent, quitte à dédoubler les portes du circuit pour simuler la division. Ensuite, nous utilisons la proposition 5.6 de [27] :

Proposition 9 *Soit p un entier fixé. Le problème de déterminer si un circuit arithmétique sans paramètres, en p variables $x_1, ..., x_p$ calcule le polynôme nul est un problème $\mathbb{P}$ standard.*

Nous pouvons utiliser le même raisonnement dans le cas différentiel. Si K est un corps différentiel de caractéristique nulle, les problèmes booléens P_K^d et $\mathbb{P}_K^d$ le sont sans paramètre. Toute dérivée est nulle sur $\mathbf{Q}$, et donc les problèmes booléens P_K^d et $\mathbb{P}_K^d$ sont respectivement P_K et $\mathbb{P}_K$ sans paramètre. Les problèmes booléens $\mathbb{P}_K^d$ sont donc les $\mathbb{P}$ au sens standard.

Les booléens P_K (et aussi les P_K^d) sont-ils P standards ? La question, posée dans [27], est ouverte.

2.2.6 Le problème des grandes racines et celui des grandes puissances

L'utilisation de la dérivée permet-elle de résoudre plus vite un problème ? Considérons un corps différentiel K. Il est clair qu'il existe des problèmes P_K^d qui ne sont même pas définissables sans la dérivée. Par exemple, décider sur une entrée $(x_1, ..., x_n)$ si x_1 est une constante, c'est-à-dire si $dx_1 = 0$. En effet, si $C(\bar{x}, \bar{c})$ est un circuit au sens des corps différentiels, si $(x_2, ..., x_n)$ et $\bar{c}$ sont fixés, C calcule 1 pour un nombre fini ou cofini de x_1. Et le corps k des constantes de K n'est ni l'un ni l'autre.

Existe-t-il un problème qui soit P_K^d, définissable au sens des corps par une suite de circuits sur le corps K, mais qui ne soit pas P_K ? La question est ouverte.
Nous pouvons répondre par la négative si la définition au sens du corps n'utilise que des paramètres constants, et si K est algébriquement clos :

Proposition 10 *Si K est un corps différentiel algébriquement clos, si k est* *un sous-corps algébriquement clos de son corps des constantes, alors :* $P_K^d \cap D_K(k) = P_K \cap D_K(k)$

Démonstration de la proposition : Soit Π un problème P_K^d, défini par la suite de circuits $(A_n(\bar{x}, \bar{a}))_{n\geq 1}$. Supposons que Π soit défini par la suite de formules du premier ordre du langage des corps $(\phi_n(x_1, ..., x_n, \bar{c}_n))_{n\geq 1}$ où les $\bar{c}_n$ sont des paramètres de k. Si Π_k est la restriction de Π à k, en éliminant les paramètres non constants des circuits A_n on obtient la suite de circuits $(B_n(\bar{x}, \bar{b}))_{n\geq 1}$ qui nous prouve que Π_k est P_k. Le corps k vérifie pour tout entier n la formule du langage des corps : $\forall \bar{x}(\phi_n(\bar{x}, \bar{c}_n) \leftrightarrow B_n(\bar{x}, \bar{b}))$. Comme k et K sont des corps algébriquement clos, K satisfait aussi ces formules, et donc Π est P_K grâce aux circuits $(B_n(\bar{x}, \bar{b}))_{n\geq 1}$. $\square$

Si nous ne connaissons pas de problème accéléré par la dérivée, nous en connaissons par contre dont nous savons qu'ils ne sont pas accélérés : ils ne sont pas décidables en temps polynomial avec la dérivée, même si nous autorisons un nombre polynomial de paramètres.

Commençons par définir les problèmes.

Soit f une fonction de l'ensemble des entiers dans lui-même, calculable en temps polynomial par une suite de circuits sans paramètre qui utilise la multiplication, l'addition, la soustraction, le test d'égalité à zéro. Il existe donc un polynôme p tel que pour tout entier naturel n, $f(n) \leq 2^{2^{p(n)}}$. Nous demandons également à ce que la fonction croisse assez rapidement. Une fonction croît doucement si elle vérifie les conditions du lemme suivant :

Lemme 7 *Il existe un polynôme q tel que la fonction f soit majorée en $+\infty$ par 2^q si et seulement s'il existe un polynôme p et une suite de circuits $(A_n(x))_{n\geq 1}$ au sens des corps sans paramètre tels que $A_n(x)$ soit de taille au plus $p(n)$ et calcule $x^{f(n)}$.*

Démonstration du lemme : Si une telle suite $(A_n(x))$ existe, alors il est clair par récurrence que pour tout n, $f(n) \leq 2^{p(n)}$. Réciproquement, s'il existe un polynôme q tel que 2^q majore f en $+\infty$, alors pour n fixé suffisamment grand nous pouvons décomposer $f(n)$ en base deux de cette manière : $f(n) = \sum_{i=0}^{q(n)-1} f_i 2^i$, où les f_i sont des booléens. Avec un circuit de taille $q(n)$, nous pouvons calculer les x^{2^i} pour i compris entre 0 et $q(n) - 1$. Et avec au plus $q(n) - 1$ portes de multiplication, nous calculons $x^{f(n)} = \prod_{i=0}^{q(n)-1} x^{f_i 2^i}$. $\square$

Supposons que f soit une fonction ne vérifiant pas les hypothèses du lemme. Nous pourrions par exemple choisir $f(n) = 2^{2^n}$, ce qui est beaucoup plus fort car alors f majore en $+\infty$ toutes les fonctions 2^q, où q est un polynôme. Soit a un élément d'un corps K.

Définition 7 *Le problème des grandes racines de l'élément a est l'ensemble $GR_{f,a}$ des uples de la forme $(x_1, ..., x_n)$, où n est un entier strictement positif, ${x_1}^{f(n)} = a$ et $x_2, ..., x_n$ sont des éléments de K. Le problème des grandes puissances de l'élément a est l'ensemble $GP_{f,a}$ des uples de la forme $(a^{f(n)}, x_2, ..., x_n)$, où n est un entier strictement positif et $x_2, ..., x_n$ des éléments de K .*

Lemme 8 *Si C est un circuit de décision de taille τ sur le corps K avec une seule variable d'entrée x, alors l'ensemble des éléments acceptés par C ou son complémentaire est fini de cardinal au plus $\tau 2^\tau$.*

Démonstration du lemme : Le circuit C utilise des paramètres $\bar{a}$ de K. Soit α un élément transcendant sur K situé dans une extension $K(\alpha)$ de K. Supposons que α appartienne au sous-ensemble S de $K(\alpha)$ accepté par C. Sur l'entrée α, le circuit C effectue des tests de la forme $P(\alpha) = 0$?, et répond non, sauf si P est le polynôme nul. Le circuit effectue au plus τ tests, et les polynômes sont de degrés inférieurs à 2^τ. Si tous les polynômes calculés sont identiquement nuls, alors pour toute entrée x les calculs sont les mêmes, et l'ensemble S est $K(\alpha)$ tout entier. Dans le cas contraire, si un élément x de $K(\alpha)$ n'appartient pas à S, alors il annule au moins un de ces polynômes non nuls. Le complémentaire de S est donc de cardinal au plus $\tau 2^\tau$. Le raisonnement est le même si nous supposons que C refuse α. □

Lemme 9 *Si K est un corps différentiel algébriquement clos, $GR_{f,a}$ est EXP_K mais pas P_K^d ni même $P_K^d/poly$.*

Démonstration du lemme : Il est clair que $GR_{f,a}$ est décidable en temps exponentiel dans K, sans dérivation et avec a comme seul paramètre. En effet, il suffit de calculer x_1 à la puissance $f(n)$, puis de comparer le résultat à a.

Si $\bar{x}$ est un uple de $GR_{f,a}$, nous avons $dx_1 = \dfrac{1}{f(n)}\dfrac{da}{a}x_1$. Soit X l'ensemble des éléments de K qui satisfont cette relation. Il n'est pas vide car il contient les solutions de $x^{f(n)} = a$, et il est infini car stable par multiplication par une constante. De plus, pour tout entier n, il existe un paramètre b_n de K tel que $x^{(n)} = b_n x$ pour tout élément x de X.

Supposons que $GR_{f,a}$ soit $P_K^d/poly$. Il existe un polynôme p et une suite de circuits de décision au sens des corps différentiels $(C_n(x_1, ..., x_n, \bar{c}_n))_{n\geq 1}$, où les $\bar{c}_n$ sont des éléments de K, tel que C_n soit de taille au plus $p(n)$ et $(x_1, ..., x_n)$ appartienne à $GR_{f,a}$ si et seulement si $C_n(x_1, ..., x_n, \bar{c}_n)$ calcule 1. En fixant $(x_2, ..., x_n)$ égal par exemple à $(0, ..., 0)$, nous obtenons une suite de circuits à une seule variable. Nous pouvons appliquer le lemme de réduction à ces circuits. Nous obtenons un polynôme q et une suite de circuits au sens des corps différentiels $(A_n(x, \bar{c}_n))_{n\geq 1}$ réduits, avec A_n de taille au plus $q(n)$. Nous ajoutons aux paramètres $\bar{c}_n$ toutes leurs dérivées jusqu'à l'ordre $q(n)$, ainsi que les paramètres $\bar{b}_n = (b_1, ..., b_{q(n)})$, et nous remplaçons dans le circuit A_n l'éventuelle porte qui calcule $x_1^{(n)}$ par une porte qui calcule $b_n x_1$. Nous obtenons un polynôme r et une suite de circuits au sens des corps $(E_n(x, \bar{e}_n))_{n\geq 1}$ avec E_n de taille au plus $r(n)$, et telle que pour tout élément x de X, $x^{f(n)} = a$ si et seulement si $E_n(x, \bar{e}_n)$ calcule 1. Pour tout entier n, l'ensemble des x_1 de X qui vérifient $x_1^{f(n)} = a$ est de cardinal $f(n)$. Or d'après le lemme 8, l'ensemble des x_1 qui satisfont E_n est soit cofini, ce qui implique ici infini, soit de cardinal au plus $r(n)2^{r(n)}$. Et pour n assez grand, $f(n)$ est strictement supérieur à $r(n)2^{r(n)}$. Donc Π n'est pas $P_K^d/poly$. $\square$

La démonstration est encore valable si le corps n'est pas algébriquement clos, mais contient un bon nombre de racines de a : si $g(n)$ est le nombre de solutions de l'équation $x^{f(n)} = a$ dans K, il faut que g ne soit majorée par aucun polynôme. Sinon, le problème est $P_K/poly$ et donc $P_K^d/poly$.

Notons que si k est un corps algébriquement clos de caractéristique nulle, alors $NP_k \subseteq EXP_k$. Ceci est dû à l'élimination rapide des quantificateurs dans les corps algébriquement clos([13]). Comme nous ne connaissons pas d'élimination rapide des quantificateurs dans les corps différentiellement clos, nous ne savons pas si $NP_K^d \subseteq EXP_K^d$ pour un corps différentiellement clos K de caractéristique nulle. Nous pouvons juste affirmer que $NP_K^d \subseteq A_K^d$, où A_K^d est la classe des problèmes algorithmiques sur le corps différentiel K, c'est-à-dire qui sont résolus par une suite de circuits utilisant un nombre fini de paramètres.

Étudions maintenant le problème des grandes puissances. C'est un problème $P_K/poly$: il est immédiat de tester si $x = a^{f(n)}$ si la constante $a^{f(n)}$ est fournie au circuit.

Proposition 11 *Si a est 0 ou une racine de l'unité, alors $GP_{f,a}$ est P_K, donc P_K^d. Sinon, $GP_{f,a}$ n'est pas $\mathbb{P}_K$, ni même $\mathbb{P}_K^d$.*

Étudions tout d'abord la complexité au sens des corps.

Lemme 10 *Si a est une racine de l'unité, alors $GP_{f,a}$ est P_K.*

Pour cela, nous commençons par montrer un lemme sur le calcul modulo un entier :

Lemme 11 *Soit A un circuit au sens des corps dont les seules entrées sont des paramètres entiers, et qui a une seule sortie. Il est de taille τ, et calcule un entier m. Soit r un entier supérieur ou égal à 2. Alors, il existe un circuit B au sens des corps dont les seules entrées sont des paramètres entiers, dont la seule sortie est le reste de la division euclidienne de m par r, et qui est de taille linéaire en τ (le coefficient est en r^2). De plus, il existe un algorithme polynomial standard qui à A associe B.*

Démonstration du lemme 11 : Commençons par former un circuit de paramètre 1 qui calcule tous les entiers entre 1 et r : nous pouvons le prendre de taille r. Nous ajoutons $r-3$ portes de multiplication pour obtenir les entiers ir pour i compris entre 2 et $r-2$. Nous allons ensuite simuler les calculs de A modulo r : nous dédoublons toutes les portes pour remplacer chaque calcul par son reste et son quotient modulo r. Les paramètres du nouveau circuit sont les restes et les quotients des divisions euclidiennes des paramètres de A par r. Voyons maintenant comment simuler les portes de calcul.

Soient a et c le reste et le quotient de la division euclidienne de l'entier a_1 par r. Tester si a_1 est nul est équivalent à tester si a et c sont nuls, et à multiplier les résultats. Les portes de test ne posant pas de problèmes, étudions maintenant le cas des portes d'addition, de soustraction et de multiplication.

Soient respectivement a et b les restes de la division euclidienne de respectivement a_1 et b_1 par r, et c et d les quotients correspondant. Le reste de la division euclidienne de $a_1 + b_1$ par r est $a + b$ si $a + b \leq r - 1$ et $a + b - r$ sinon. Pour tester si $a + b \leq r - 1$, il suffit de tester si $a + b - i = 0$ pour tous les entiers i compris entre 0 et $r - 1$. Il nous faut donc : une porte pour additionner a et b, $r - 1$ portes pour calculer les $a + b - i$, r portes pour tester s'ils sont nuls, encore r portes pour faire la négation des portes précédentes (la négation du booléen x est 1-x) et $r - 1$ portes pour multiplier les résultats. Ce produit p vaut 1 si et seulement si pour tout entier i compris entre 0 et $r - 1$, $a + b - i$ est non nul, et donc si $a + b \geq 0$. Le reste cherché est donc $p(a + b - r) + (1 - p)(a + b)$, et le quotient est $p(c + d + 1) + (1 - p)(c + d)$, ce qui nous demande 10 portes supplémentaires de calcul. Le calcul du reste et du quotient de la division euclidienne de $a_1 + b_1$ par r à partir des restes a et b peut donc être effectué par un circuit de $4r + 9$ portes.

Le reste de la division euclidienne de $a_1 - b_1$ par r est $a - b$ si $a - b \geq 0$ et $a - b + r$ sinon. Le calcul se fait donc avec le même nombre de portes que précédemment.

Pour la multiplication, le produit ab est compris entre 0 et $(r - 1)^2 = (r - 2)r + 1$. Il nous faut donc effectuer tous les tests $a + b - i = 0$ pour tous les entiers i compris entre 0 et $(r - 2)r - 1$ pour savoir dans quel intervalle se situe le produit ab. Nous effectuons ensuite les produits des négations des tests pour obtenir des booléens $p_0 ... p_{r-3}$ tels que $p_i = 0$ si et seulement si $ab \in [ir, (i + 1)r - 1]$. Le reste de la division euclidienne de $a_1 b_1$ par r est alors :

$$\begin{aligned}&(1 - p_0)ab + p_0[(1 - p_1)(ab - r) + \\ &p_1[...[(1 - p_{r-3})(ab - r(r - 3)) + p_{r-3}(ab - r(r - 2))]...]]\end{aligned}$$

et le quotient est

$$\begin{aligned}&(1 - p_0)(ad + cb + cdr) + p_0[(1 - p_1)(ad + cb + cdr + 1) + p_1[... \\ &[(1 - p_{r-3})(ad + cb + cdr + (r - 3)) + p_{r-3}(ad + cb + cdr + (r - 2))]...]]\end{aligned}$$

En conclusion, le calcul du reste et du quotient de la division euclidienne de $a_1 b_1$ par r à partir des restes a et b peut donc être effectué par un circuit de $8r^2 - r - 17$ portes. $\square$

Nous pouvons en conclure qu'il existe un algorithme polynomial qui a un entier n associe le reste de la division euclidienne de $f(n)$ par r, que nous noterons r_n.

Démonstration du lemme 10 : Nous formons le circuit $A_n(x_1, ..., x_n, a)$ qui calcule a^{r_n} à partir du paramètre a, et qui teste si $x_1 = a^{r_n}$. Ce circuit est de taille au plus $r + 2$ si nous sous-entendons les entrées $x_2, ..., x_n$ qui n'interviennent pas dans le calcul, et de taille $r + n + 1$ sinon. La suite $(A_n(x_1, ..., x_n, a))_{n \geq 1}$ décide du problème $GP_{f,a}$. □

Étudions maintenant le cas où a est algébrique sur $\mathbf{Q}$ sans être ni 0 ni une racine de l'unité :

Proposition 12 *Si K est l'ensemble des nombres algébriques $\bar{\mathbf{Q}}$ et si a n'est ni 0 ni une racine de l'unité, alors $GP_{f,a}$ n'est pas P_K, ni même $\mathbb{P}_K$.*

Pour le montrer, nous avons besoin d'une fonction de K dans $\mathbf{R}$, appelée hauteur, utilisée dans [6] pour montrer le lemme des témoins, et qui vérifie les propriétés (P) suivantes :

- $H(1) = H(0) = 1$, $H(2) = 2$, et pour tout w, $H(w) \geq 1$ et $H(-w) = H(w) = H(1/w)$
- pour tous v et w, $H(v + w) \leq 2H(v)H(w)$, et par récurrence pour tout uple $(v_0, ..., v_n)$, $H\left(\sum_{i=0}^{n} v_i\right) \leq 2^n \prod_{i=0}^{n} H(v_i)$
- pour tout w et tout entier n, $H(w^n) = H(w)^n$, et pour tous v et w, $H(vw) \leq H(v)H(w)$
- pour tous v et w, $H(v + w) \geq \frac{1}{2}\frac{H(v)}{H(w)}$
- pour tout v et pour tout w non nul, $H(vw) \geq \frac{H(v)}{H(w)}$

Nous noterons par H la hauteur classique sur $\bar{\mathbf{Q}}$. La définition peut être trouvée dans le livre de S. Lang ([22]). D'après [6], H satisfait les propriétés (P), et nous avons les :

Définitions 8 *Si $\bar{u} = (u_1, ..., u_n)$ est un uple d'éléments de $\bar{\mathbf{Q}}$, alors sa hauteur $H(\bar{u})$ est le maximum des hauteurs $H(u_i)$, pour i compris entre 1 et n.*

Si P est un polynôme sur $\bar{\mathbf{Q}}$, en une ou plusieurs variables, alors sa hauteur $H(P)$ est le produit des hauteurs de ses coefficients.

Le lemme suivant est le Corollaire 1 d'une proposition du livre [6], et se montre facilement en utilisant seulement les propriétés (P) de H :

Lemme 12 *Soit P un polynôme à une variable et à coefficients dans $\bar{\mathbf{Q}}$, de degré d. Soit x un élément de $\bar{\mathbf{Q}}$ qui vérifie : $H(x) > 2^d H(P)$. Alors x est racine de P si et seulement si P est le polynôme nul.*

Commençons par donner une majoration des entiers :

Lemme 13 *Si N est un entier dont la valeur absolue est strictement inférieure à 2^{n+1}, où n est un entier naturel, alors $H(N) \leq 2^n(2^{n+1}-1) \leq 2^{2n+1}$*

Démonstration du lemme 13 : Comme $H(N) = H(-N)$, il suffit de le montrer pour N positif. Écrivons N en base deux. Il existe des booléens $N_0, ..., N_n$ tels que $N = \sum_{i=0}^{n} N_i 2^i$. D'après les propriétés (P) nous avons :

$$H(N) \leq 2^n \prod_{i=0}^{n} H(N_i)H(2^i) = 2^n \prod_{i=0}^{n} 2^i = 2^n(2^{n+1}-1)$$

□

Remarque : Si nous prenons $n = [\log_2 N]$, nous avons :

$$H(N) \leq 2^{2[\log_2 N]+1} \leq 2^{2\log_2 N+1} = 2N^2$$

Nous pouvons maintenant énoncer un résultat de majoration de la hauteur d'un polynôme :

Lemme 14 *Soit P un polynôme en une variable x produit par un circuit au sens des corps de taille τ ayant pour entrée x ainsi que les paramètres $\bar{b} = (b_1, ..., b_r)$ de $\bar{\mathbf{Q}}$. Alors :*

$$H(P) \leq 2^{2^\tau (2^\tau+1)^{r+1}(2+\log_2 H(\bar{b}))}$$

Démonstration du lemme 14 : Le polynôme P est de degré au plus 2^τ, et les coefficients de P peuvent s'écrire sous la forme :

$$c = \sum_{0 \leq \alpha_1, \dots, \alpha_r \leq 2^\tau} n_{\alpha_1, \dots, \alpha_r} b_1^{\alpha_1} \dots b_r^{\alpha_r}$$

où les $n_{\alpha_1, \dots, \alpha_r}$ sont des entiers strictement inférieurs en valeur absolue à 2^{2^τ}, et $\alpha_1 + \dots + \alpha_r \leq 2^\tau$. Nous avons alors :

$$H(c) \leq 2^{(2^\tau+1)^r - 1} \prod_{0 \leq \alpha_1, \dots, \alpha_r \leq 2^\tau} H(n_{\alpha_1, \dots, \alpha_r}) H(b_1)^{\alpha_1} \dots H(b_r)^{\alpha_r}$$

et donc

$$H(c) \leq 2^{(2^\tau+1)^r} (2^{2^\tau} H(\bar{b})^{2^\tau})^{(2^\tau+1)^r} \leq 2^{2^\tau (2^\tau+1)^r (1 + \log_2 H(\bar{b}))}$$

et

$$H(P) \leq \left(2^{2^\tau (2^\tau+1)^r (1 + \log_2 H(\bar{b}))}\right)^{2^\tau+1} \leq 2^{2^\tau (2^\tau+1)^{r+1} (1 + \log_2 H(\bar{b}))}$$

□

Pour démontrer la proposition 12, nous avons encore besoin d'une proposition (Proposition 1.1 de [22]) qui affirme que a est de hauteur strictement plus grande que 1 :

Proposition 13 *Si x est un élément de $\bar{\mathbf{Q}}$, alors $H(x) = 1$ si et seulement si x est 0 ou une racine de l'unité.*

Démonstration de la proposition 12 : Supposons que le problème $GP_{f,a}$ soit $\mathbb{P}_K$. Alors, il existe des paramètres $\bar{b} = (b_1, \dots, b_r)$, un polynôme p et une suite de circuits au sens des corps $(C_n(x_1, \dots, x_n, y_1, \dots, y_r))_{n \geq 1}$ tels que pour tout entier strictement positif n, le circuit $C_n(\bar{x}, \bar{b})$ soit de taille au plus $p(n)$ et calcule 1 si et seulement si $x_1 = a^{f(n)}$. Si nous fixons les entrées $x_2, \dots, x_n$ du circuit C_n, nous obtenons un circuit de même taille et qui reconnaît le même ensemble. Nous supposons donc que les entrées du circuit C_n sont x_1 et $\bar{b}$.

Lors du calcul sur l'entrée $a^{f(n)}$, le circuit effectue au moins un test de la forme $P_n(x_1, \bar{b}) = 0$?, où P_n est un polynôme non nul en x_1 et à coefficients entiers, et $P_n(a^{f(n)}, \bar{b}) = 0$. En effet, si ce n'était pas le cas, le circuit reconnaîtrait un ensemble infini. Le polynôme P_n est de degré total au plus $d_n = 2^{p(n)}$. Notons $Q_n(x_1)$ le polynôme P_n pour lequel les dernières variables sont fixées à $\bar{b}$. Par hypothèse, Q_n n'est pas identiquement nul. D'après la proposition 14, nous avons :

$$H(Q_n) \leq 2^{2^{p(n)}(2^{p(n)}+1)^{r+1}(2+\log_2 H(\bar{b}))}$$

Or la hauteur de $a^{f(n)}$ est $H(a)^{f(n)}$, c'est-à-dire $2^{f(n)\log_2(H(a))}$. Et pour un entier n suffisamment grand :

$$2^{f(n)\log_2(H(a))} > 2^{2^{p(n)}(2^{p(n)}+1)^{r+1}(2+\log_2 H(\bar{b}))}$$

et donc, alors que $a^{f(n)}$ est racine du polynôme non nul Q_n :

$$H(a^{f(n)}) > 2^{d_n} H(Q_n)$$

ce qui contredit le lemme 12. □

Lemme 15 *Si a est transcendant sur $\mathbf{Q}$, alors $GP_{f,a}$ n'est ni P_K ni $\mathbb{P}_K$.*

Démonstration du lemme : Si $GP_{a,f}$ est $\mathbb{P}_K$, alors il est $\mathbb{P}_{\mathbf{Q}(a)}$ (théorème d'élimination dans les corps). Soit p un polynôme et $(A_n(x_1, ..., x_n, a)_{n\geq 1})$ une suite de circuits au sens des corps de taille bornée par p qui décide de $GP_{a,f}$. Fixons les variables $(x_2, ..., x_n)$.

Le circuit A_n effectue des tests de la forme $P(x_1, a) = 0$?. Le polynôme P est à coefficients dans $\mathbf{Q}$. Il est de degré total au plus $2^{p(n)}$. S'il est de degré r supérieur ou égal à 2 en x_1, alors le terme dominant de $P(a^{f(n)}, a)$ vu comme un polynôme en a est de degré au moins $rf(n)$, et le terme suivant de degré au plus $(r-1)f(n) + 2^{p(n)}$. Et pour n suffisamment grand, $f(n) > 2^{p(n)}$ et donc $rf(n) > (r-1)f(n) + 2^{p(n)}$. Comme a est transcendant sur $\mathbf{Q}$, l'expression $P(a^{f(n)}, a)$ est nulle si et seulement si le polynôme $P(x_1, a)$ est identiquement nul, et donc s'annule pour toute entrée x_1, ce qui signifie que le test peut être remplacé par la constante 1. Si le circuit n'effectue aucun test, l'ensemble reconnu est soit vide, soit $\mathbf{Q}(a)$ tout entier.

Le circuit A_n effectue donc sur l'entrée $a^{f(n)}$ des tests polynomiaux de la forme $P(x_1, a) = 0$?, avec $P(a^{f(n)}, a) \neq 0$ (au moins 1). Donc, d'après ce qui précède, le polynôme $P(x_1, a)$ n'est pas identiquement nul en x_1. Et donc, pour la même raison que précédemment, $P(2a^{f(n)}, a) \neq 0$. Les calculs de A_n sur les entrées $a^{f(n)}$ et $2a^{f(n)}$ sont rigoureusement identiques, ce qui est impossible. $\square$

Si K est un corps différentiel, nous pouvons maintenant nous demander si $X_{f,a}$ est $\mathbb{P}_K^d$. Pour cela, il faut nécessairement que sa restriction à son corps des constantes k soit P_k. Or la restriction de $X_{f,a}$ à k est lui-même si a est une constante, et l'ensemble vide sinon. Donc :

Corollaire 2 *Si a est 0 ou une racine de l'unité, le problème $GP_{f,a}$ est P_K et donc P_K^d. Si a n'est ni 0 ni une racine de l'unité mais appartient à k, alors $GP_{f,a}$ n'est ni $\mathbb{P}_K$ ni $\mathbb{P}_K^d$.*

Il nous reste à traiter le cas où a n'est pas une constante.

Lemme 16 *Si a est d-transcendant ou d-algébrique sur $\mathbf{Q}$ et non constant, $GP_{f,a}$ n'est pas $\mathbb{P}_K^d$.*

Démonstration du lemme :
Si $GP_{f,a}$ est $\mathbb{P}_K^d$, alors il est $\mathbb{P}_{\mathbf{Q}(a)_d}^d$. Il existe un polynôme p et une suite de circuits au sens des corps différentiels $(C_n(x_1, ..., x_n, a))_{n \geq 1}$ tels que pour tout entier strictement positif n, le circuit $C_n(\bar{x}, a)$ soit de taille au plus $p(n)$ et calcule 1 si et seulement si $x_1 = a^{f(n)}$. Nous fixons $x_2, ..., x_n$: les entrées du circuit C_n sont x_1 et a.

Si $x = a^{f(n)}$, alors $dx = f(n)\frac{da}{a}x$, et pour tout entier r, il existe un polynôme différentiel P_r à coefficients entiers tel que $d^r x = P_r(\frac{da}{a})x$. Le polynôme différentiel P_{r+1} est donné par la relation de récurrence : $P_{r+1}(X) = \frac{da}{a}P_r(X) + dP_r(X)$ et $P_1(X) = f(n)\frac{da}{a}X$.

Nous commençons par réduire les circuits. Puis nous remplaçons le calcul des dérivées successives de x_1 par celui des P_r. Nous réduisons à nouveau. Nous obtenons des circuits $A_n(x_1, a)$ de tailles polynomiales et bornées par le polynôme q dans lesquels les seules dérivations sont les dérivations successives du paramètre a, et qui décide de $x_1 = a^{f(n)}$ pour les entrées x_1 de l'ensemble infini $X = \{x/dx = f(n)\frac{da}{a}x\}$.

Supposons que a soit d-transcendant sur $\mathbf{Q}$. Le circuit A_n effectue des tests de la forme $P(x_1, a, da, ..., d^{q(n)}a) = 0?$, où P est un polynôme à coefficients dans $\mathbf{Q}$ de degré total au plus $2^{q(n)}$. Pour les mêmes raisons que dans la démonstration du lemme 15, P s'annule en $a^{f(n)}$ si et seulement s'il est identiquement nul en x_1. La conclusion est la même.

Supposons maintenant que a soit d-algébrique sur $\mathbf{Q}$. Soit r l'ordre de son polynôme différentiel minimal, qui est à coefficients dans $\mathbf{Q}$. Il existe une suite de circuits au sens des corps $(B_n(x_1, a, ..., a^{(r)}))$ de taille polynomiale et équivalente à la suite $(A_n(x_1, a))$ pour les entrées x_1 appartenant à X. En effet, formons le circuit différentiel qui calcule $a^{(r+1)}$ avec comme entrées a, ..., $a^{(r)}$. Puis mettons $q(n)$ dérivées successives sous $a^{(r+1)}$. Réduisons le circuit obtenu. Le calcul de $a^{(r+2)}$ n'utilise que les dérivées d'ordre inférieur à $r + 1$. Nous remplaçons donc celles des dérivées successives des entrées qui calculent $a^{(r+2)}$ par la porte qui le calcule en utilisant seulement les $r+1$ premières dérivées, c'est à dire que les flèches qui partaient de celles-là partent maintenant de celle-ci. Par récurrence, nous supprimons toutes les portes de dérivées. En combinant ce circuit au circuit A_n, nous obtenons un circuit B_n qui n'a pas de porte de dérivation. Or nous avons $\mathbf{Q}(a)_d = \mathbf{Q}(a, ..., a^{(r)})$. En utilisant l'élimination des paramètres dans les corps, nous obtenons une suite de circuits $(\Gamma_n(x_1, a))$ équivalent à $(A_n(x_1, a))$ pour les entrées x_1 appartenant à $Y = X \cap \mathbf{Q}(a)$, qui est un ensemble infini. Le paramètre a est transcendant sur $\mathbf{Q}$. Comme $2a^{f(n)} \in Y$, nous pouvons conclure comme au lemme 15. $\square$

Chapitre 3

Résolutions universelles pour des problèmes NP-complets

3.1 Introduction

Nous connaissons de nombreux problèmes NP-complets dans le cas classique. La façon habituelle de montrer qu'un problème est NP-complet est de le réduire à un autre problème NP-complet connu. Mais cela ne nous apprend rien sur sa structure. Pour y remédier et essayer de mieux comprendre la structure des problèmes NP-complets, Manindra Agrawal et Somenath Biswas ont introduit la notion d'universalité, qui est plus forte que la NP-complétude, bien qu'elle ne s'applique pas exactement aux mêmes objets. Un théorème permet alors de montrer de manière structurelle la propriété d'universalité, et donc la NP-complétude. Nous allons généraliser ce théorème pour d'autres structures que la structure classique de calcul sur les booléens. Nous nous plaçons dans le cadre de calcul défini pour une structure quelconque, comme il apparaît dans [15] et dans [27]. Nous pouvons alors donner des exemples d'applications de ce théorème. En dernier lieu, nous pouvons nous demander si tous les problèmes NP-complets ont la propriété d'universalité. La réponse est non, et nous donnons un contre-exemple.

3.2 Notations et définitions

3.2.1 Le problème SFR

Nous nous plaçons dans le cadre défini au chapitre IV de [27]. Nous considérons une structure M. Son langage $L(M)$, que nous prendrons fini, contient les symboles de constantes 0 et 1, des symboles de fonctions f d'arité a_f, dont l'identité, et des symboles de relations r d'arité a_r, dont l'égalité.

Dans [27], B.Poizat définit les **formules rudimentaires** comme la conjonction de formules qui peuvent être de quatre sortes :

1. les formules à paramètre a, élément de M : $x = a$
2. les formules faisant intervenir une fonction f du langage $L(M)$, et un uple $\bar{x}$ de variables de longueur a_f : $y = f(\bar{x})$
3. les formules faisant intervenir une relation r du langage $L(M)$, et un uple $\bar{x}$ de variables de longueur a_r : $r(\bar{x}) \vee (y = 0)$ et $\neg r(\bar{x}) \vee (y = 1)$. Dans le cas où y est une variable booléenne, la conjonction de ces deux formules est équivalente à $r(\bar{x}) \leftrightarrow (y = 1)$. Cela signifie que y est la valeur de vérité de $r(\bar{x})$.
4. les formules booléennes : $(x = 0) \vee (x = 1)$ et $(x = \varepsilon) \vee (y = \varepsilon') \vee (z = \varepsilon'')$ où $\varepsilon, \varepsilon'$ et ε'' valent 0 ou 1

Il fait observer que le problème de la satisfaisabilité des formules rudimentaires est NP-complet, au sens de M. Il remarque aussi que dans le cas des calculs classiques, où la structure est réduite aux booléens $\{0,1\}$, ce problème est pratiquement identique au problème SAT3 de Cook.

Nous allons ici utiliser une variante plus restreinte de ce problème. Nous appellerons **formule rudimentaire de base** les formules des trois premières sortes, c'est-à-dire faisant intervenir un paramètre, une fonction ou une relation de la structure, ainsi que les équations booléennes $(x = 0) \vee (x = 1)$ et $(x = 1) \vee (y = 1) \vee (z = 1)$. Nous obtenons un nouveau problème que nous noterons **SFR** : le problème de satisfaisabilité des conjonctions de formules rudimentaires de base. Ce problème est également NP-complet. En effet, puisque l'égalité fait partie des relations, nous pouvons exprimer la négation booléenne $x = \neg y$ par la satisfaisabilité du système qui fait intervenir une variable z supplémentaire :

$$[(x = 0) \vee (x = 1)] \wedge [(y = 0) \vee (y = 1)] \wedge [z = 0] \wedge [(x \neq y) \vee (z = 1)]$$

Cela nous permet, de remplacer les formules booléennes
$(x = \varepsilon) \vee (y = \varepsilon') \vee (z = \varepsilon'')$ où $\varepsilon, \varepsilon'$ et ε'' valent 0 ou 1 par des formules du type $(x = 1) \vee (y = 1) \vee (z = 1)$.

3.2.2 Les résolutions universelles

Soit M^* l'ensemble des suites finies d'éléments de M. Définissons une opération de projection : soit $\bar{x}$ un uple d'éléments de M, de longueur n. Un **masque** pour $\bar{x}$ est un uple α d'entiers naturels strictement positifs tous distincts, rangés en ordre croissant, et dont le dernier est au plus n (nous pouvons le représenter par un mot binaire, donc par un élément de M^*). La projection de $\bar{x}$ à travers α est le uple formé des éléments de $\bar{x}$ dont l'indice figure dans α. L'ordre est conservé. Cela se note :

$$\alpha = i_1 i_2 ... i_p \text{ avec } 0 < i_1 < i_2 < ... < i_p < n \text{ et } \bar{x} = x_1 x_2 ... x_n$$

$$\alpha(\bar{x}) = x_{i_1} x_{i_2} ... x_{i_p}$$

Pour un ensemble S de uples de longueur $(n + p)$ et un ensemble T de uples de longueur p, nous dirons que l'ensemble T est égal à la projection de S à travers le masque α s'il est égal à l'ensemble des projections des éléments de S à travers α. Dans [1], les auteurs remarquent que : $(\beta(\alpha))(S) = T$ équivaut à l'existence d'un ensemble S_1 tel que $\alpha(S) = S_1$ et $\beta(S_1) = T$. En effet, projeter à travers deux masques successifs revient à les superposer : $(\gamma(\beta))(\alpha) = \gamma(\beta(\alpha))$.

Si $\bar{x}$ est un uple, $|\bar{x}|$ désigne sa longueur. Si R est un sous-ensemble de $M^* \times M^*$, $(x, s) \in R$ est noté xRs.

Une **résolution** est un sous-ensemble R de $M^* \times M^*$, décidable en temps polynomial, et pour lequel il existe un polynôme p tel que si xRs, alors nécessairement s soit de longueur au plus $p(|x|)$. L'ensemble des s tels que xRs est appelé l'ensemble des solutions de x, et noté $\mathbf{sol_R(x)}$. Une résolution R est dite **admissible** si, pour toute entrée x, tous les uples de $sol_R(x)$ ont la même longueur. Cette longueur est notée $\mathbf{longsol_R(x)}$. Toute résolution se transformant aisément en résolution admissible, nous ne considérerons désormais que ces dernières. A toute résolution admissible R, est associé un ensemble NP : $L_R = \{x \text{ tel que } (\exists s) x R s\}$. Réciproquement, à tout ensemble A, supposé NP, nous pouvons associer une résolution admissible « naturelle ». Par exemple, si le problème NP se présente sous la forme de systèmes d'équations à résoudre, sa résolution naturelle est celle qui à un système associe des solutions ; $longsol_R(x)$ est alors le nombre d'inconnues du système.

Soient Q et R deux résolutions. Une fonction f de M^* dans $M^* \times \{0;1\}^*$ est **une réduction de Q à R, qui préserve les solutions** si elle est calculable en temps polynomial et satisfait aux conditions suivantes :

1. $f(x) = (f_1(x), f_2(x)) = (z, \alpha)$ avec $x \in M^*, z \in M^*$ et $|\alpha| = longsol_Q(x)$
2. $\alpha(sol_R(z)) = sol_Q(x)$

Remarquons que f_1 est une réduction du problème L_Q au problème L_R. En effet, un uple x appartient à L_Q si et seulement si son ensemble de solutions $sol_Q(x)$ n'est pas vide, c'est-à-dire si et seulement si l'ensemble de solutions $sol_R(z)$ n'est pas vide, ce qui équivaut bien à dire que z appartient à L_R.

Une résolution R est **universelle** si, pour toute résolution Q, il existe une réduction de Q à R, qui préserve les solutions. Dans ce cas là, et d'après la remarque précédente, L_R est NP-complet.

3.2.3 Un critère d'universalité

Plaçons-nous dans une structure M. Soit R une résolution admissible. Nous allons définir des conditions nécessaires et suffisantes pour que R soit universelle. Ces conditions sont faites pour que la résolution « naturelle » associée au problème SFR satisfasse le critère de manière évidente.

Le premier groupe de conditions (1) définit des « blocs », qui nous permettent de trouver les solutions des différents types de formules rudimentaires de base:
(1.1) Pour chaque élément a de M, il existe un élément de M^*, noté $bloc_a$, et un masque α_a de longueur 1 tels que :

$$\alpha_a(sol_R(bloc_a)) = \{a\}$$

Cela nous donne toutes les solutions de $x = a$.
Nous voulons de plus que la fonction ϕ, qui à a associe $bloc_a$ et α_a, soit calculable, et que son temps de calcul soit majoré, indépendamment de a.
(1.2) Pour chaque fonction f du langage $L(M)$ de M, et en particulier pour l'identité, il existe un élément $bloc_f$ de M^*, et un masque α_f de longueur $a_f + 1$ tels que :

$$\alpha_f(sol_R(bloc_f)) = \{x_1x_2...x_{a_f}y \text{ tel que } y = f(x_1x_2...x_{a_f})\}$$

Cela nous donne toutes les solutions de $y = f(\bar{x})$.
Remarquons que si α_{id} est le couple d'entiers (α_1, α_2), alors :

$$\alpha_1(sol_R(bloc_{id})) = M$$

(1.3) Pour chaque relation r de $L(M)$, il existe deux éléments $bloc_r$ et $bloc_{\neg r}$ de M^*, et deux masques α_r et $\alpha_{\neg r}$ de longueur $a_r + 1$ tels que :

$$\alpha_r(sol_R(bloc_r)) = \{x_1x_2...x_{a_r}y \text{ tel que } x_1x_2...x_{a_r} \in r \text{ et } y \in M\} \cup M^{a_r}.0$$

$$\alpha_{\neg r}(sol_R(bloc_{\neg r})) = \{x_1x_2...x_{a_r}y \text{ tel que } x_1x_2...x_{a_r} \notin r \text{ et } y \in M\} \cup M^{a_r}.1$$

Avec les solutions de $bloc_r$, nous avons toutes les solutions de $r(\bar{x}) \vee (y = 0)$ et $bloc_{\neg r}$ nous donne toutes les solutions de $\neg r(\bar{x}) \vee (y = 1)$.
(1.4) Il existe un élément $bloc$ de M^*, et un masque α de longueur 3 tels que :

$$\alpha(sol_R(bloc)) = \{0, 1\}^3 - \{(0; 0; 0)\}$$

Cela nous donne toutes les solutions booléennes de $(x = 1) \vee (y = 1) \vee (z = 1)$.
Remarquons que si α est le triplet d'entiers $(\alpha_1, \alpha_2, \alpha_3)$, alors :

$$\alpha_1(sol_R(bloc)) = \{0; 1\}$$

Les conditions (2) et (3) définissent des fonctions qui nous permettent de décrire l'ensemble des solutions lorsque nous avons affaire à la conjonction de plusieurs formules. La fonction de jonction nous permet de faire la conjonction de deux formules, en ayant pris soin de renommer les variables de manière à ce qu'une même variable n'apparaissent pas dans les deux formules de départ. Il suffit alors de concaténer les variables. Avec la fonction de couplage,

nous faisons en sorte qu'une variable qui a été renommée conserve cependant la même valeur : en fait, nous imposons des égalités entre des couples de variables.
(2) Il existe une fonction $join$, de $(M^*)^*$ dans $M^* \times \{0,1\}^*$, calculable en temps polynomial, qui vérifie : pour tous éléments $x_1, x_2, ..., x_n$ de M^*, il existe un élément z de M^* et un masque α de longueur $\sum_{k=1}^{n} longsol_R(x_k)$ tels qu'en projetant les solutions de z à travers α, nous obtenons la concaténation des solutions des x_i. Cela s'écrit : $join(x_1, ..., x_n) = (z, \alpha)$ et $\alpha(sol_R(z)) = \{s_1...s_n \text{ tel que } \forall k \leq n,\ s_k \in sol_R(x_k)\}$
(3) Il existe une fonction cpl calculable en temps polynomial telle que pour tout élément x de M^*, et pour tous les entiers $i_1, ..., i_n, j_1, ..., j_n$ compris entre 1 et $longsol_R(x)$, il existe un élément z de M^* et un masque α de longueur $longsol_R(x)$ tels que les solutions de z projetées à travers α donnent les solutions de x pour lesquelles les variables x_{i_m} et x_{j_m} prennent les mêmes valeurs :

$$cpl(x, (i_1, ..., i_n), (j_1, ..., j_n)) = (z, \alpha)$$

et

$$\alpha(sol_R(z)) = \{s \in sol_R(x) \text{ tel que } \forall m \leq n,\ s_{i_m} = s_{j_m}\}$$

Nous pouvons maintenant énoncer le théorème qui généralise le théorème 4.4 de [1], et qui nous permet de montrer de manière structurelle qu'un problème est NP-complet :

Théorème 14 *Une résolution R est universelle si et seulement si elle vérifie les conditions (1), (2) et (3).*

Nous allons associer au problème SFR une résolution, et vérifier qu'elle est universelle. Puis en s'appuyant sur ce résultat, nous montrerons le théorème.

Définition 9 *Soit S une conjonction de formules rudimentaires de base à n inconnues. S utilise les paramètres $a_1...a_k$ de M, qui apparaissent dans les formules de base du premier type. Dans l'écriture de la formule, nous remplaçons a_i par l'entier booléen $i+1$. La formule obtenue est représentée de manière classique par un mot booléen m. La conjonction S est finalement* *représentée par le mot $u = a_1 ... a_k m$.* **La résolution $\mathbf{R}_{SFR}$** *est définie par : pour tous les éléments $r_1 ... r_n$ de M, pour tout mot u sur M, $uR_{SFR}r_1 ... r_n$ si et seulement si u est le code d'une formule S à n variables libres et $r_1 ... r_n$ est solution de S.*

Lemme 17 *R_{SFR} est universelle.*

DÉMONSTRATION DU LEMME : La preuve est analogue à celle du théorème 6.3 de [27], montrant que SFR est NP-complet.

Soit R une résolution au sens de la structure M. Cherchons une réduction de R à R_{SFR}, qui préserve les solutions. Soit u un élément fixé de M^*, $n = longsol_R(u)$. Nous voulons déterminer $r_1 ... r_n$ tels que $uRr_1 ... r_n$. La résolution R étant décidable en temps polynomial, nous pouvons former le circuit qui décide, pour une entrée $r_1 ... r_n$ de taille n, si $uRr_1 ... r_n$ est vérifié. Nous étiquetons chaque porte de ce circuit par une variable. Les portes d'entrées qui ne sont pas des constantes sont étiquetées par $x_1 ... x_n$. A chaque porte du circuit est effectué soit un test $r(\bar{x})$, soit un calcul $y = f(\bar{x})$, où y est la variable qui étiquette la porte. Nous pouvons donc associer à chaque porte une conjonction de formules rudimentaires de base qui exprime le travail fait. La variable de sortie doit être posée égale à 1. Nous prenons la conjonction de toutes ces formules, et nous obtenons une formule à p variables. Elle est codée par v. Considérons le masque $\alpha = 1, 2, ..., n$: nous ne gardons que les variables des portes d'entrée. La fonction qui à u associe le couple (v, α) est une réduction de R à R_{SFR}, qui préservent les solutions. □

Lemme 18 *R_{SFR} satisfait (1), (2) et (3).*

DÉMONSTRATION DU LEMME :
Les éléments relatifs à R_{SFR} sont notés avec l'indice 1. Tous les masques sont triviaux (nous conservons toutes les variables). C'est pourquoi nous omettons de les donner dans cette démonstration.

(1.1) $bloc_{a,1}$ est le code de la formule $x_1 = a$.

(1.2) $bloc_{f,1}$ est le code de la formule $x_{a_f+1} = f(\bar{x})$.

(1.3) $bloc_{r,1}$ est le code de la formule : $r(\bar{x}) \vee (x_{a_r+1} = 0)$

$bloc_{\neg r,1}$ est le code de la formule : $\neg r(\bar{x}) \vee (x_{a_r+1} = 1)$

(1.4) $bloc_1$ est le code de la formule :

$$[(x_1 = 0) \vee (x_1 = 1)] \wedge [(x_2 = 0) \vee (x_2 = 1)] \wedge [(x_3 = 0) \vee (x_3 = 1)]$$
$$\wedge [(x_1 = 1) \vee (x_2 = 1) \vee (x_3 = 1)]$$

(2) Soient $s_1, ..., s_n$ les codes des formules rudimentaires S_1, ..., S_n. Nous renommons les variables de manière à ce que les variables de S_1 soient numérotées de 1 à i_1, celles de S_2 soient numérotées de $i_1 + 1$ à i_2, et ainsi de suite, jusqu'aux variables de S_n, qui sont numérotées de $i_{n-1} + 1$ à i_n. En faisant la conjonction de toutes les formules, nous obtenons une nouvelle formule S, de code s, et qui a pour variables $x_1 ..., x_{i_n}$.

(3) Soit s le code de la formule rudimentaire S à p inconnues, et des entiers $i_1, ..., i_n, j_1, ..., j_n$ compris entre 1 et $longsol_1(x)$. Une nouvelle formule S' est obtenue en faisant la conjonction de S et des formules : $x_{i_1} = x_{j_1}$ ... $x_{i_n} = x_{j_n}$. Elle est codée par s'. □

DÉMONSTRATION DU THÉORÈME :

Les éléments relatifs à R sont notés avec l'indice R.

• Si R est universelle, alors R vérifie (1),(2) et(3).

Soit f une réduction de R_{SFR} à R, et g une réduction de R à R_{SFR}, qui préservent les solutions. Posons $f(bloc_1) = (z, \alpha)$, où α est un masque qui, appliqué aux solutions de z pour R, donne les solutions de $bloc_1$ pour R_{SFR}, c'est-à-dire $\{0,1\}^3 - \{(0,0,0)\}$. Il nous suffit de prendre $bloc_R = z$, et $\alpha_R = \alpha (= \alpha_1(\alpha)$ car α_1 est trivial). Nous définissons de même tous les blocs pour R. Comme ϕ_1 est calculable, et a un temps de calcul majoré indépendamment de a, et puisque f est calculable en temps polynomial, alors ϕ_R est calculable, de temps de calcul majoré indépendamment de a.

Les constructions de cpl_R et $join_R$ demandent l'utilisation des deux réductions f et g. Cherchons, par exemple, à construire $join_R$. Soient $x_1, ..., x_n$ des instances pour R. Soient : $g(x_i) = (y_i, \alpha_i)$ pour i compris entre 1 et n, $(y, \gamma) = join_1(y_1, ..., y_n)$, $(x, \delta) = f(y)$. Posons : $S(i) = \sum_{j=1}^{i} longsol_{SFR}(y_j)$ et $\beta = \alpha_1, \alpha_2 + S(1), ..., \alpha_n + S(n-1)$.

Les solutions pour R de x, projetées à travers les masques successifs δ, γ puis β, sont les concaténées des solutions pour R des x_k, pour k variant de 1 à n. Il nous suffit donc de prendre $\lambda = \beta(\gamma(\delta))$ et nous avons : $join_R(x_1, ..., x_n) = (x, \lambda)$. Nous pouvons construire de manière analogue cpl_R.

• Si R vérifie (1),(2) et(3), alors R est universelle.
Pour montrer que R est universelle, il suffit de trouver une réduction f de R_{SFR} à R, qui préserve les solutions.

Soit S une conjonction de m équations rudimentaires de base. Elle a n variables libres. Nous cherchons un mot u et un masque β tels que les solutions de u, relativement à la résolution R, et projetées à travers β, soient exactement les solutions de la formule S. Pour cela, nous savons que nous pouvons considérer les formules de base une par une, puis renommer les variables, coupler et joindre. Il suffit de faire quelques calculs d'indices, omis ici.

Nous savons qu'il existe un masque α tel que la projection des solutions de $bloc_{id,R}$ à travers α soit M tout entier. Donc nous allons prendre autant de $bloc_{id,R}$ qu'il y a de variables différentes, c'est-à-dire n. Puis, pour chaque formule $x_i = a$, nous prenons un $bloc_{a,R}$: sa solution, qui projetée à travers $\alpha_{a,R}$ est a, de longueur 1, sera couplée avec la solution du $bloc_{id,R}$ qui est en i$^{\text{ème}}$ position, et qui représente la variable x_i. De même, respectivement pour chaque formule de base $y = f(\bar{x})$, $r(\bar{x}) \vee (y = 1)$, $\neg r(\bar{x}) \vee (y = 1)$ et $(x = 1) \vee (y = 1) \vee (z = 1)$, nous prenons un $bloc_{f,R}$, un $bloc_{r,R}$, un $bloc_{\neg r,R}$ et un $bloc_R$. La longueur de la projection de la solution d'un *bloc*, à travers le masque associé, est le nombre de variables de la formule de base correspondant. Nous couplons donc, pour chaque variable, sa valeur dans cette solution avec sa valeur dans la solution du $bloc_{id,R}$ associé à cette variable. Pour chaque formule de base $(x = 0) \vee (x = 1)$, nous prenons un $bloc_R$. En effet, nous savons qu'il existe un masque α_1 de longueur 1 tel que la projection des solutions de $bloc_R$ à travers α_1 soit $0, 1$. Nous avons donc $(m + n)$ «*blocs*», que nous pouvons calculer dans un temps qui ne dépend que linéairement de $(m + n)$. Par l'application successive des fonctions $join_R$ et cpl_R, nous déterminons l'élément u. Il est de longueur polynomiale. C'est l'image de S par la réduction f cherchée. R est donc une résolution universelle. $\square$

Remarques:
1. Ce théorème est une généralisation du théorème 4.4 de [1]. Dans cet article, les auteurs considèrent des calculs classiques qui, dans nos conventions, deviennent des calculs dans la structure réduite aux deux booléens 0 et 1.

Le critère (1) est réduit à (1.4), c'est-à-dire à l'existence de *bloc* et de α, la définition de $join$ est inchangée et dans la définition de *cpl*, la condition $s_{i_m} = s_{j_m}$ est remplacée par $s_{i_m} \neq s_{j_m}$. Pourquoi n'avons nous pas la même définition de *cpl*? Parce que, dans le cas de [1], nous n'avons pas le *bloc* de la fonction $\neg$. Il faut écrire : $x \neq y$. Et pour pouvoir coupler par l'égalité deux variables x et y, il suffit alors d'introduire une nouvelle variable z et d'écrire : $x \neq z$ et $z \neq y$. Les variables booléennes ne pouvant prendre que les deux valeurs 0 et 1, nous avons bien $x = y$. Dans le cas général, nous ne pouvons pas garder la définition de [1], car l'ensemble de base de la structure M peut avoir un nombre infini d'éléments.

2. Dans certains cas, tous les «*blocs*» ne sont pas utiles, car ils peuvent être obtenus à partir d'autres «*blocs*» et à l'aide des fonctions de jonction et de couplage $join$ et *cpl*. C'est en particulier le cas des corps ordonnés. Par exemple, la formule $x = y$ est équivalente à la conjonction des formules $x = y + z$ et $z = 0$. Nous n'avons donc pas besoin de $bloc_-$, non plus que de *bloc*, $bloc_\leq$, $bloc_>$, $bloc_=$ et $bloc_\neq$.

3.3 Exemples de résolutions universelles

3.3.1 Le cas booléen

Dans [1], les auteurs présentent plusieurs exemples de résolutions universelles dans le cas standard. Ces résolutions sont les résolutions «naturelles» associées à certains problèmes :

- savoir si un graphe orienté possède un sous-graphe hamiltonien
- savoir si un graphe orienté possède un ensemble dispersé de cardinal r
- problème du sac à dos

Mais si nous connaissons de nombreux problèmes NP-complets standards, nous ne pouvons pas en dire autant pour les autres structures.

3.3.2 Le cas de la structure $(\mathbf{R}, +, -, \times, \leq)$

Le problème 4-FEAS

Il apparaît dans [7] : étant donné un polynôme réel F à n variables et de degré au plus quatre, a-t-il un zéro ? Ce problème est NP-complet pour la structure du corps ordonné des réels. Nous pouvons montrer de plus qu'il admet une résolution universelle. Plaçons-nous dans la structure $(\mathbf{R}, +, -, \times, \leq)$. Le polynôme F est codé par f, liste des coefficients de ses monômes, qui est de taille polynomiale en n. Définissons la résolution naturelle associée à 4-FEAS par : si $a_1, ..., a_n$ sont des réels, $f\, R_4\, a_1 ... a_n$ si et seulement si $(a_1, ..., a_n)$ est racine du polynôme F. Montrons que cette résolution est universelle. Le $bloc_+$ est le code de l'équation $x = y + z$, et le $bloc_\times$ est le code de l'équation $x = yz$. D'après la deuxième remarque du paragraphe précédent, il est inutile de chercher les autres «*blocs*», car l'ensemble des réels est un corps ordonné. Pour joindre des équations, nous ajoutons des variables de manière à n'avoir plus que des équations de degré deux. Puis nous sommons leurs carrés. Nous obtenons bien une équation polynomiale de degré au plus quatre. Pour coupler, il suffit d'ajouter à l'équation la somme des $(x_{i_k} - x_{j_k})^2$. Cela montre que R_4 est universelle.

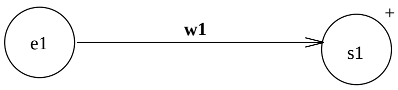
Les réseaux neuronaux

Dans [32], l'auteur expose un problème NP-complet. Nous allons montrer que nous pouvons associer à ce problème une résolution universelle.

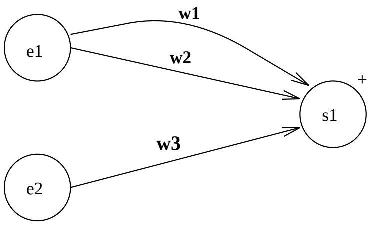
Un réseau de neurones est ici un graphe orienté fini, sans cycle orienté. Les sommets sont appelés neurones. Les neurones qui ne reçoivent pas de flèches sont les entrées, tandis que les neurones dont ne part aucune flèche sont les sorties. Chaque arête a un poids réel. A un réseau de neurones est associé le calcul suivant : au départ, nous donnons à chaque entrée une valeur réelle. Au pas de calcul suivant, nous pouvons calculer la valeur que prend un neurone, si celui-ci ne reçoit des flèches que des neurones d'entrée. Considérons un neurone N, qui reçoit n flèches, venant des neurones $N_1, ..., N_n$. La flèche qui provient du neurone N_i, qui a la valeur x_i, a un poids w_i. Le neurone N prend alors la valeur $\sum_{i=1}^{n} w_i x_i$. Ainsi, de proche en proche, et comme il n'y a pas de cycle orienté, nous pouvons calculer les valeurs prises par tous les neurones du réseau. La sortie est le uple constitué des valeurs des neurones de sortie.

Nous appellerons architecture A la donnée d'un réseau dont les arêtes sont étiquetées par des variables. Une même variable peut étiqueter plusieurs arêtes. Pour un réseau qui a n entrées et p sorties, une tâche T est un couple de $\mathbf{R}^n \times \mathbf{R}^p$. Un réseau satisfait la tâche $T = (a, b)$ si, après calcul sur l'entrée a, la sortie du réseau est b. Le problème considéré, que nous appellerons P1, est le suivant :
Etant données une architecture A et une tâche T, pouvons-nous assigner aux variables de A des valeurs réelles, de manière à ce que le réseau obtenu satisfasse la tâche T ?

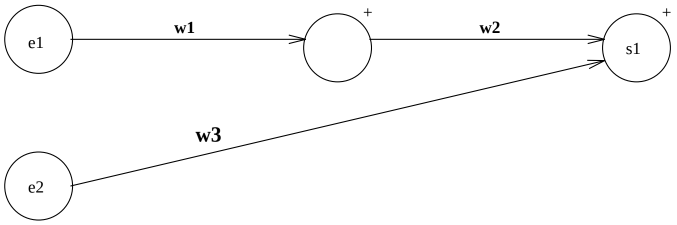
Nous pouvons immédiatement associer à ce problème une résolution naturelle R_1 : $(A, T)R_1\bar{w}$ si le réseau d'architecture A avec les poids $\bar{w}$ satisfait la tâche T. Comment représenter A par des uples de réels ? Nous pouvons coder le graphe de manière classique en binaire. La liste des poids $\bar{w}$ sera donnée dans l'ordre des variables. Nous pouvons maintenant montrer que R_1 est universelle en utilisant le théorème. Tous les masques sont triviaux. Nous omettons donc de les donner. Le signe «+» au dessus d'un neurone indique que celui-ci calcule une somme linéaire.
Vérification de (1) : les poids sont des réels, donc des éléments d'un corps ordonné.
$bloc_a$: c'est le *bloc* de la formule $w_1 = a$.
La tâche est T=(1)(a) et l'architecture est :



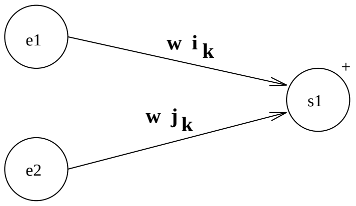
$bloc_+$: c'est le *bloc* de la formule $w_1 + w_2 = w_3$.
La tâche est T=(1;-1)(0) et l'architecture est :



$bloc_\times$: c'est le *bloc* de la formule $w_1 w_2 = w_3$.
La tâche est T=(1;-1)(0) et l'architecture est :



Pour joindre, il suffit de juxtaposer les réseaux, en ayant pris soin de renommer les poids, et de concaténer les entrées et les sorties des tâches.
Pour déterminer $cpl(x,(i_1,...,i_n),(j_1,...,j_n))$, il faut juxtaposer à x les réseaux, pour k compris entre 1 et n :



Il ne reste alors plus qu'à concaténer avec n tâches T=(1;-1)(0), de la même manière que pour la fonction $join$.

3.3.3 Le cas de la structure $(\mathbf{R},+,-,\leq)$

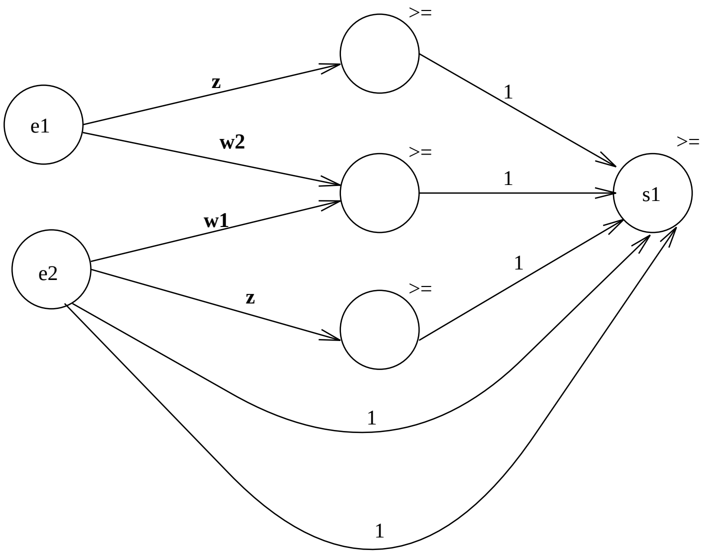
Nous allons considérer un réseau de neurones, mais qui diffère un peu du précédent. Les neurones qui ne sont pas des entrées peuvent être de deux natures différentes : somme linéaire ou test linéaire. Au cours d'un calcul, si un neurone «somme linéaire» reçoit n flèches, venant des neurones $N_1,...,N_n$, ayant les valeurs $x_1,...,x_n$, et avec des poids $w_1,...,w_n$, alors il prend la valeur $\sum_{i=1}^{n} w_i x_i$. Si le neurone effectue un test linéaire, il prend la valeur 1 si $\sum_{i=1}^{n} w_i x_i \geq 0$ et 0 sinon. De plus, certains poids peuvent être fixés, dans l'architecture, et non plus variables. Dans ce cas, ils doivent être égaux à 1 ou à -1. En particulier, tous les poids des arêtes ne partant pas d'un neurone entrée doivent être fixés. Ceci permet d'éviter d'avoir à effectuer une multiplication entre deux réels quelconques. Une architecture est maintenant composée d'un graphe, de la nature des neurones qui le composent, ainsi que des poids fixés à l'avance, et des variables pour les autres poids. Une même variable peut étiqueter plusieurs flèches. La tâche T est définie de même que précédemment, mais avec une condition supplémentaire : les éléments du uple de l'entrée doivent être choisis parmi les réels 1 et -1. Le problème est alors le suivant : étant données une architecture A et une tâche T, pouvons-nous assigner des valeurs réelles aux variables, de manière à ce que le réseau obtenu satisfasse la tâche T ?

La résolution naturelle associée à ce problème est définie comme dans le cas précédent. Une fois les valeurs des poids du réseau fixées, nous pouvons vérifier en temps polynomial qu'il vérifie la tâche. En effet, nous n'avons pas à effectuer de véritables multiplication. Montrons que la résolution est universelle. Les «*blocs*» des formules $x = a$, ainsi que $bloc_+$, sont les mêmes que précédemment. Nous pouvons également joindre et coupler de la même manière. Nous en déduisons les «*blocs*» de la soustraction et de l'identité. Il nous reste à trouver les «*blocs*» de l'égalité et de la relation d'ordre. Par exemple, la formule $(x \leq y) \vee (z = 0)$ est vraie si et seulement si au moins deux des tests suivants ont une réponse positive : $y - x \geq 0$, $z \geq 0$ et $-z \geq 0$. Cela équivaut à : la somme des réponses à ces tests, moins deux, est positive ou nulle.

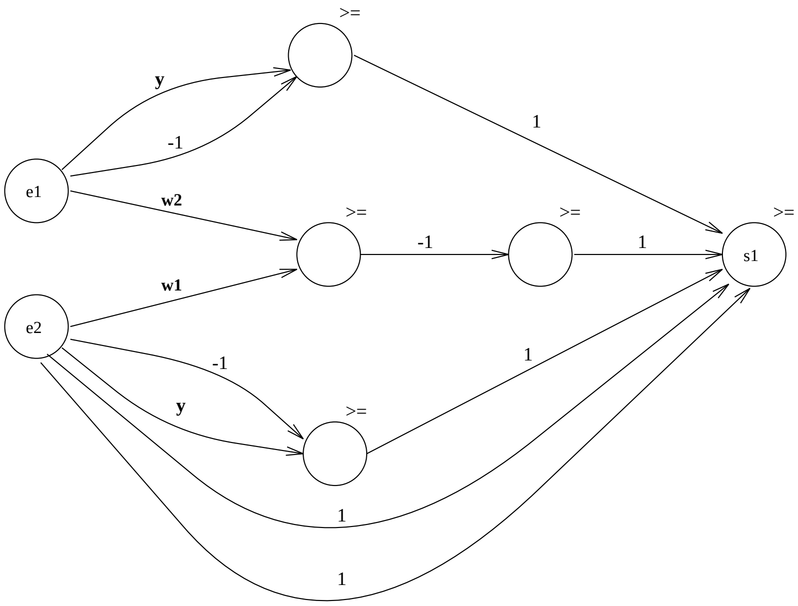
Tous les neurones des «*blocs*» suivants sont des tests linéaires (notés >=) :

$bloc_{\leq}$: c'est le *bloc* de la formule $(w_1 \leq w_2) \vee (z = 0)$.

La tâche est T=(1;-1)(0) et l'architecture est :

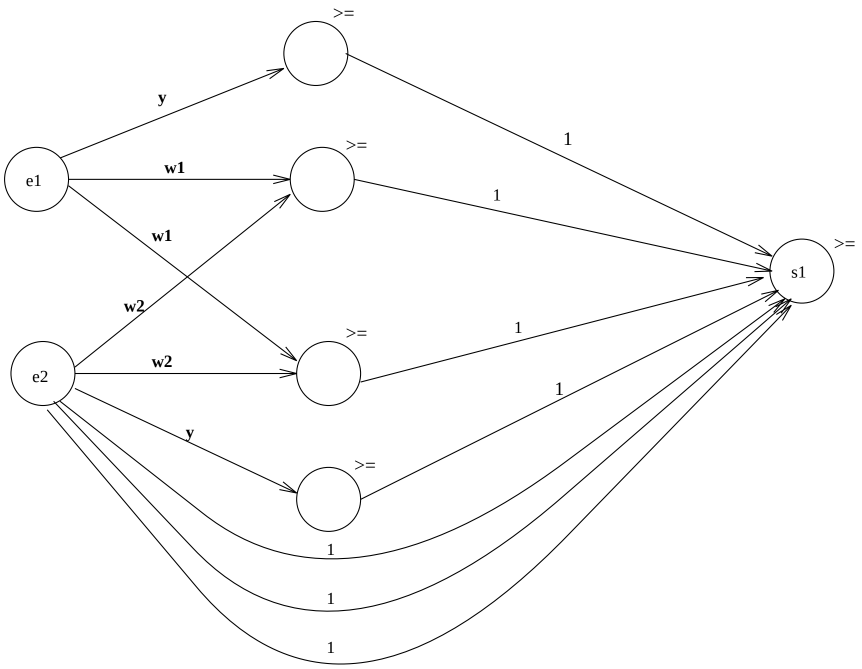


$bloc_{\neg\leq}$: c'est le *bloc* de la formule $(w_1 > w_2) \vee (y = 1)$.
La tâche est T=(1;-1)(0) et l'architecture est :



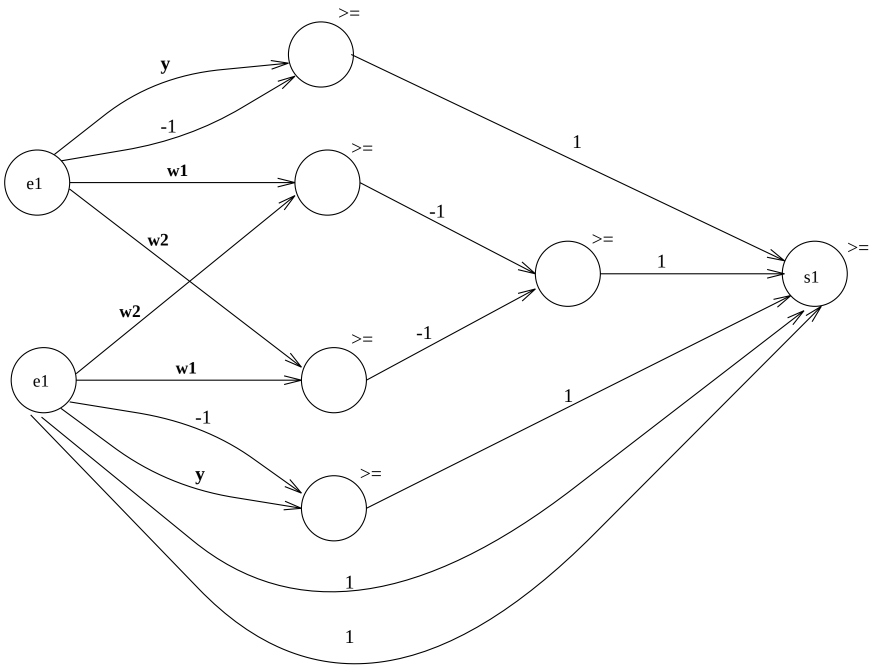
$bloc_{=}$: c'est le $bloc$ de la formule $(w_1 = w_2) \vee (y = 0)$.

La tâche est T=(1;-1)(0) et l'architecture est :

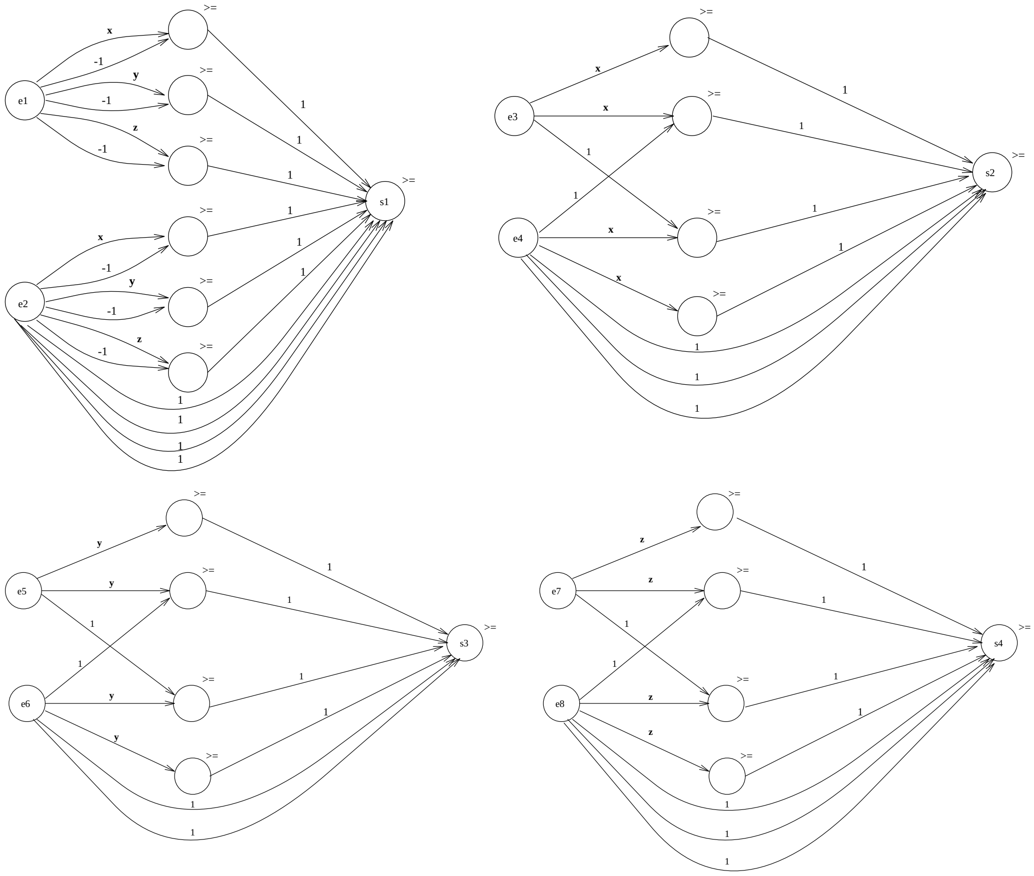


$bloc_{\neq}$: c'est le *bloc* de la formule $(w_1 \neq w_2) \vee (z = 1)$.

La tâche est T=(1;-1)(0) et l'architecture est :



$bloc$: c'est le $bloc$ de la formule $(x = 1) \vee (y = 1) \vee (z = 1)$. Il faut imposer que les trois variables soient booléennes, et que leur disjonction soit égale à 1.
La tâche est T=(1;-1;1;-1;1;-1;1;-1)(0;0;0;0) et l'architecture est :



3.3.4 Les contre-exemples

A tout problème NP-complet, A, auquel est associée une résolution R universelle, nous pouvons facilement associer une résolution R' non universelle. Nous définissons R' par : $s_1 s_2$ est solution de x pour R' si et seulement si s_1 ou s_2 est solution de x pour R. Or R' n'est pas universelle : $bloc_a$ n'existe pas. En effet, si s_2 est une solution de x pour R, et si s_1 est quelconque, $s_1 s_2$ est une solution de x pour R'. Or, si α_a est compris entre 1 et $longsol(x)$, c'est un élément s_{1_i} de s_1 qui est projeté, donc n'importe quel élément de M. Nous avons alors : $\alpha_a(sol_{R'}(bloc_a)) = M$, ce qui apporte une contradiction dès que M a plus de deux éléments. De même si α_a est compris entre $longsol(x)+1$ et $2longsol(x)$. Donc $bloc_a$ n'existe pas.

Dans [1], Manindra Agrawal et Somenath Biswas posent, pour le cas classique, le problème de savoir si tout problème NP-complet possède une résolution universelle. Ils affirment que nous avons, le cas échéant, $P \neq NP$, mais que même en supposant ceci, la question reste difficile. Par contre, nous pouvons facilement donner des exemples de structures M avec des problèmes NP-complets qui ne possèdent pas de résolution universelle. D'après la proposition 7.2 de [27] : dans toute structure de langage relationnel fini et qui élimine les quantificateurs, le problème de satisfaisabilité des formules sans quantificateurs et sans paramètres est un problème booléen NP-complet. Un ensemble infini muni de la seule relation d'égalité est une telle structure. Montrons qu'une résolution R associée à un problème NP-complet booléen A dans une structure infinie M ne peut satisfaire l'existence de tous les $bloc_a$. En effet, nous savons qu'il existe une fonction ϕ, qui à a associe $bloc_a$ et α_a, calculable, de temps de calcul majoré par m, indépendamment de a. Tous les $bloc_a$ sont donc des mots booléens de taille majorée par m. L'ensemble des $bloc_a$ est donc fini. Or M est infini et les $bloc_a$ sont tous distincts. Nous avons donc une impossibilité. Donc R n'est pas universelle. Nous avons bien un problème NP-complet auquel ne peut être associée aucune résolution universelle.

Bibliographie

[1] Agrawal (Manindra) et Biswas (Somenath). – Universal relations. *proceedings of the 7th Structure in Complexity Theory conference*, 1992, pp. 207–220. – To appear in Information and Computation in its journal form.

[2] Balcázar (José L.), Díaz (Josep) et Gabarró (Joaquim). – *Structural Complexity II*. – Springer-Verlag, 1990, *EATCS Monographs on Theoretical Computer Science*.

[3] Balcázar (José L.), Díaz (Josep) et Gabarró (Joaquim). – *Structural Complexity I*. – Springer-Verlag, 1995, second édition.

[4] Blum (Lenore). – *Generalized algebraic structures : A model theoretic approach*. – Thèse de PhD, Massachussetts Institute of Technology, 1968.

[5] Blum (Lenore), Cucker (Felipe), Shub (Michael) et Smale (Steve). – *Complexity and Real Computation*. – Springer-Verlag, 1997.

[6] Blum (Lenore), Cucker (Felipe), Shub (Mike) et Smale (Steve). – Algebraic settings for the problem "$P \neq NP$?". *in Mathematics of Numerical Analysis, série Lectures in Applied Mathematics, éd. Renegar et al.*, vol. 32, 1996, pp. 125–144.

[7] Blum (Lenore), Shub (Mike) et Smale (Steve). – On a theory of computation and complexity over the real numbers : NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, vol. 21, n° 1, July 1989, pp. 1–46.

[8] Boulier (François). – *Étude et implantation de quelques algorithmes en algèbre différentielle.* – Thèse de PhD, Université des Sciences et Technologies de Lille, 1994.

[9] Chapuis (Olivier) et Koiran (Pascal). – Saturation and stability in the theory of computation over the reals. – 1997. Prepublications de l'Institut Girard Desargues.

[10] Cucker (Felipe). – $P_{\mathbb{R}} \neq NC_{\mathbb{R}}$. *Journal of Complexity*, vol. 8, 1992, pp. 230–238.

[11] Cucker (Felipe) et Grigoriev (Dima). – *On the power of real Turing machines over binary inputs.* – Technical Report n° NC-TR-94-007, NeuroCOLT, August 1994.

[12] Cucker (Felipe), Shub (Mike) et Smale (Steve). – *Separation of Complexity Classes in Koiran's Weak Model.* – Research Report n° RC 19346 (84217), IBM Research Division, 1994.

[13] Fitchas (Noaï), Galligo (André) et Morgenstern (Jacques). – Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields. *Journal of Pure and Applied Algebra*, vol. 67, 1990, pp. 1–14.

[14] Garey (M.R.) et Johnson (D.S.). – *Computers and Intractability: A Guide to the theory of NP-Completeness.* – W.H. Freeman and Company, 1979.

[15] Goode (John B.). – Accessible telephone directories. *The Journal of Symbolic Logic*, vol. 59, n° 1, March 1993, pp. 92–105.

[16] Koiran (Pascal). – A weak version of the Blum, Shub & Smale model. *34th Annual IEEE Symposium on Foundations of Computer Science*, 1993, pp. 486–495.

[17] Koiran (Pascal). – Computing over the reals with addition and order. *Theoretical Computer Science*, vol. 133, 1994, pp. 35–48.

[18] Koiran (Pascal). – *Elimination of Constants from Machines over Algebraically Closed Fields.* – DIMACS Technical Report n° 96-24, 46, Allée d'Italie, 69364 Lyon Cedex 07, France, Laboratoire de l'Informatique et du Parallélisme, ENS Lyon, july 1996.

[19] Koiran (Pascal). – *Elimination of Parameters in the Polynomial Hierarchy.* – Research Report n° 97-37, 46, Allée d'Italie, 69364 Lyon Cedex 07, France, Laboratoire de l'Informatique et du Parallélisme, ENS Lyon, octobre 1997.

[20] Koiran (Pascal). – Randomized and deterministic algorithms for the dimension of algebraic varieties. *in Proc. 38th IEEE Symposium of Foundations of Computer Science*, 1997, pp. 36–45.

[21] Kolchin (E.). – *Differentially Algebra and Algebraic Groups.* – Academic Press, 1973.

[22] Lang (Serge). – *Fundamentals of Diophantine Geometry.* – Springer-Verlag, 1983.

[23] Marker (D.), Messmer (M.) et Pillay (A.). – *Model Theory of Fields.* – Springer, 1996, *Lecture Notes in Logic.*

[24] Meer (Klaus). – A note on a $P \neq NP$ result for a restricted class of real machines. *Journal of Complexity*, vol. 8, 1992, pp. 451–453.

[25] Michaux (Christian). – $P \neq NP$ over the non-standard reals implies $P \neq NP$ over R. *Theoretical Computer Science*, no133, 1994, pp. 95–104.

[26] Poizat (B.). – *Cours de théorie des modèles.* – Nur Al-Mantiq Walma'rifah, 1985.

[27] Poizat (B.). – *Les petits cailloux.* – ALEAS éditeur, 1995.

[28] Portier (Natacha). – Stabilité polynômiale des corps différentiels. – à paraître dans JSL.

[29] Portier (Natacha). – Résolutions universelles pour des problèmes NP-complets. *Theoretical Computer Science*, vol. 201, july 1998, pp. 137–150.

[30] Robinson (Abraham). – On the concept of a differentially closed field. *Bull. Res. Council Israel*, vol. F8, 1959, pp. 113–128.

[31] Wood (Carol). – The model theory of differential fields revisited. *Israel journal of mathematics*, vol. 25, 1976, pp. 331–352.

[32] Zhang (Xiao-Dong). – Complexity of neural network learning in the real number model. – January 1992. Preprint, University of Massachusetts at Amherst.

Index