ELSEVIER

Theoretical Computer Science 201 (1998) 137–150

Theoretical Computer Science

Résolutions universelles pour des problèmes NP-complets

Natacha Portier *

Inst. Girard Desargues, UPRES-A 5028 du CNRS Université Claude Bernard LYON-I, Bâtiment 101, 43, bld. 11 Novembre 1918, F-69622 Villeurbanne Cedex, France

Received June 1996
Communicated by F. Cucker

Abstract

Agrawal and Biswas (1992) define a notion stronger than NP-completeness. With every language X in NP is associated a polynomial-time verifiable binary relation Y, called a resolution, so that $\bar{x}$ is in X if and only if there exists $\bar{y}$, which size is a polynomial function of the size of $\bar{x}$, and $(\bar{x}, \bar{y})$ is in Y. Such an $\bar{y}$ is called a solution of $\bar{x}$ for X. If Y and Y' are resolutions associated with X and X', a solution-preserving reduction of Y to Y' is a reduction of X to X', so that the solutions of any instance for X can be quickly recovered from the solutions of the image of the instance under the reduction. A resolution Y is called universal if there exists a solution-preserving reduction from every resolution to Y. Then, Manindra Agrawal and Somenath Biswas give a theorem that help us to show that a resolution is universal, without searching for reduction. We generalize this definition and this theorem for languages over an arbitrary structure, and in particular over the reals, as it was defined by Blum et al. (1989). We then study examples with neural networks.

Keywords: Complexity; NP-completeness; Reduction; Neural network

1. Introduction

Nous connaissons de nombreux problèmes NP-complets dans le cas classique [3]. 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

* E-mail: portier@desargues.univ-lyon1.fr.

0304-3975/98/$19.00
PII S0304-3975(97)00029-7

apparaît dans [4] et dans [5]. 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 exemple.

2. Notations et définitions

2.1. Le problème SFR

Nous nous plaçons dans le cadre défini au chapitre IV de [5]. 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 [5], 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 $(x = 1) \vee (y = 1) \vee (z = 1)$.

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:

$$\alpha = i_1 i_2 \dots i_p \text{ avec } 0 < i_1 < i_2 < \cdots < i_p < n \quad \text{et} \quad \bar{x} = x_1 x_2 \dots x_n,$$
$$\alpha(\bar{x}) = x_{i_1} x_{i_2} \dots x_{i_p}.$$

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 polynômial, et pour lequel il existe un polynôme p tel que si xRs, alors nécessairement s est de longueur au plus $p(|x|)$. L'ensemble des s tels que xRs est appelé l'ensemble des solutions de x, et noté $sol_R(x)$. 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 $longsol_R(x)$. Toute résolution se transformant aisément en résolution admissible, nous ne considèrerons désormais que ces dernières. A toute résolution admissible R, est associé un ensemble NP : $L_R = \{x \text{ tel que } (\exists s) xRs\}$. 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, et d'après la remarque précédente, L_R est NP-complet.

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:

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

Cela nous donne toutes les solutions de $y = f(\bar{x})$. Remarquons que si $\alpha_{id} = \sigma_1\sigma_2$, 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:

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

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

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 $x_1, x_2, \ldots, x_n$ 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:

$$join(x_1, \ldots, x_n) = (z, \alpha) \text{ et } \alpha(sol_R(z)) = \{s_1 \ldots s_n \text{ tel que } \forall k \leqslant n,\ s_k \in sol_R(x_k)\}.$$

(3) Il existe une fonction *cpl* calculable en temps polynomial telle que pour tout élément x de M^*, et pour tous les entiers $i_1, \ldots, i_n, j_1, \ldots, j_n$ 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:

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

et

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

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. *Une résolution R est universelle si et seulement si elle vérifie les conditions* (1) *à* (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. Soit S une conjonction de formules rudimentaires de base à n inconnues. S utilise les paramètres $a_1 \ldots a_k$ 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 $u = a_1 \ldots a_k m$. *La résolution* R_{SFR} est définie par: pour tous les éléments $r_1 \ldots r_n$ de M, pour tout mot u sur M, $uR_{SFR}r_1 \ldots r_n$ si et seulement si u est le code d'une formule S à n variables libres et $r_1 \ldots r_n$ est solution de S.

Lemme. R_{SFR} *est universelle.*

Démonstration. La preuve est analogue à celle du théorème 6.3 de [5], 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 $r_1 \ldots r_n$ tels que $uRr_1 \ldots r_n$. La résolution R étant decidable en temps polynômial, nous pouvons former le circuit qui décide, pour une entrée $r_1 \ldots r_n$ de taille n, si $uRr_1 \ldots r_n$ 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 $x_1 \ldots x_n$. 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 $\alpha = 1, 2, \ldots, n$: 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. *R_{SFR} satisfait les conditions* (1) *à* (3).

Démonstration. 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 $s_1, \ldots, s_n$ les codes des formules rudimentaires $S_1, \ldots, S_n$. 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 $x_1, \ldots, x_{i_n}$.

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

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

(1) Si R est universelle, alors R vérifie les conditions (1) à (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 polynômial, 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 $x_1, \ldots, x_n$ des instances pour R.

Soient: $g(x_i)=(y_i,\alpha_i)$ pour i compris entre 1 et n, $(y,\gamma)=join_1(y_1,\ldots,y_n)$, $(x,\delta)=f(y)$. Posons: $S(i)=\sum_{j=1}^{i} longsol_{SFR}(y_j)$ et $\beta=\alpha_1,\alpha_2+S(1),\ldots,\alpha_n+S(n-1)$.

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: $join_R(x_1,\ldots,x_n)=(x,\lambda)$. Nous pouvons construire de manière analogue cpl_R.

(2) Si R vérifie les conditions (1) à (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. □

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 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 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*, $bloc_{\leqslant}$, $bloc_>$, $bloc_=$ et $bloc_{\neq}$.

3. Exemples de résolutions universelles

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.

3.2. Le cas de la structure $(\mathbf{R}, +, -, \times, \leqslant)$

3.2.1. Le problème 4-FEAS

Il apparaît dans [2]: 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 $(\mathbf{R}, +, -, \times, \leqslant)$. Le polynôme F est codé par f, liste des coefficients de ses monômes, qui est de taille polynômiale en n. Définissons la résolution naturelle associée à 4-FEAS par: si $a_1, \ldots, a_n$ sont des réels, $f R_4 a_1 \ldots a_n$ si et seulement si $(a_1, \ldots, a_n)$ 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 polynômiale 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.

3.2.2. Les réseaux neuronaux

Dans [6], l'auteur expose un problème NP-complet. Nous allons montrer que nous pouvons associer à ce problème une résolution universelle.

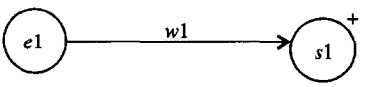
Un réseau de neurones est ici un graphe orienté fini, sans cycle orienté. Les sommets sont appelés neurones. Les neurones qui ne reçoivent pas de flèches sont les entrées, tandis que les neurones dont ne part aucune flèche sont les sorties. Chaque arête a un poids réel. A un réseau de neurones est associé 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 $N_1,\ldots,N_n$. 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 $R^n \times R^p$. 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:

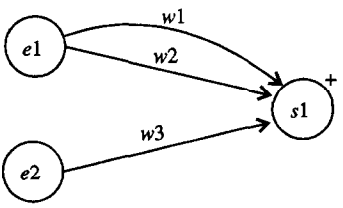
Etant données une architecture A et une tâche T, pouvons-nous assigner aux variables de A des valeurs réelles, de manière à ce que le réseau obtenu satisfasse la tâche T?

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:

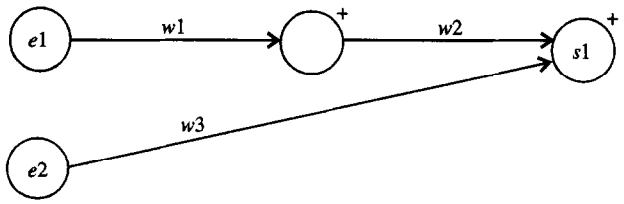


$bloc_+$: c'est le *bloc* de la formule $w_1 + w_2 = w_3$.
La tâche est $T=(1;-1)(0)$ et l'architecture est:

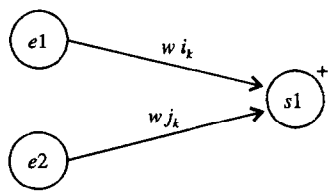


$bloc_\times$: c'est le *bloc* de la formule $w_1 w_2 = w_3$.

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 $cpl(x,(i_1,\ldots,i_n),(j_1,\ldots,j_n))$, il faut juxtaposer à x les réseaux, pour k compris entre 1 et n:



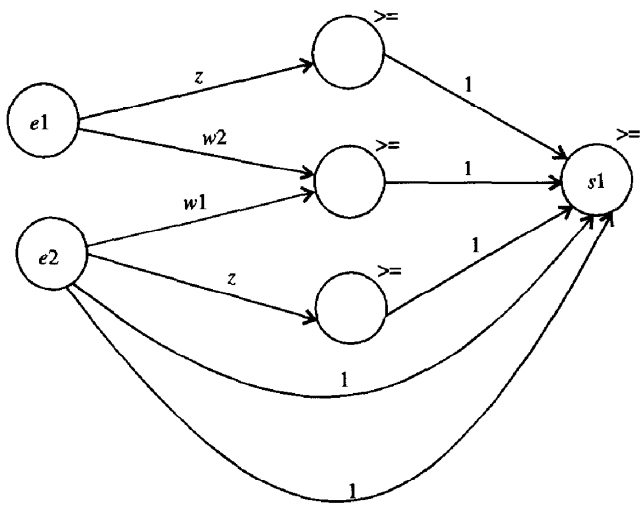
Il ne reste alors plus qu'à concaténer avec n tâches $T=(1;-1)(0)$, de la même manière que pour la fonction *join*.

3.3. *Le cas de la structure* $(\mathbf{R},+,-,\leqslant)$

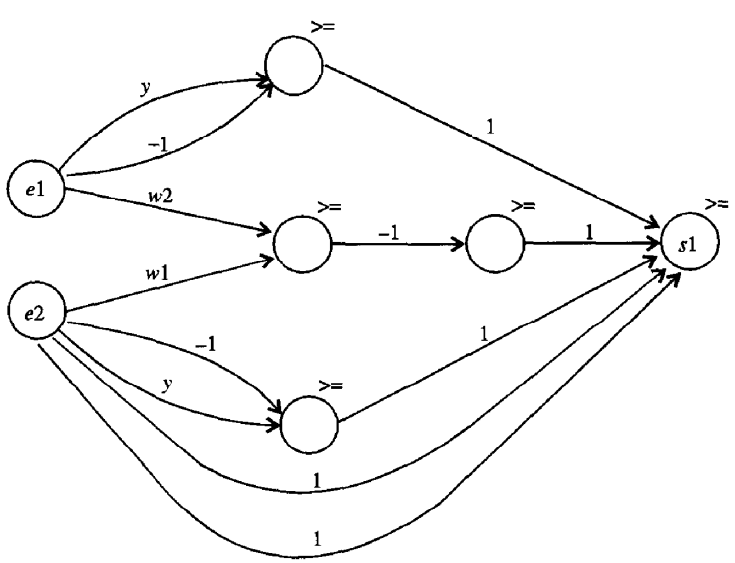
Nous allons considérer un réseau de neurones, qui diffère un peu du précédent. Les neurones qui ne sont pas des entrées peuvent être de deux natures différentes: somme linéaire ou test linéaire. Au cours d'un calcul, si un neurone "somme linéaire" reçoit n flèches, venant des neurones $N_1,\ldots,N_n$, ayant les valeurs $x_1,\ldots,x_n$, et avec des poids $w_1,\ldots,w_n$, 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 $\sum_{i=1}^{n} w_i x_i \geqslant 0$ 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 polynômial qu'il vérifie la tâche. En effet, nous n'avons pas à effectuer de véritables

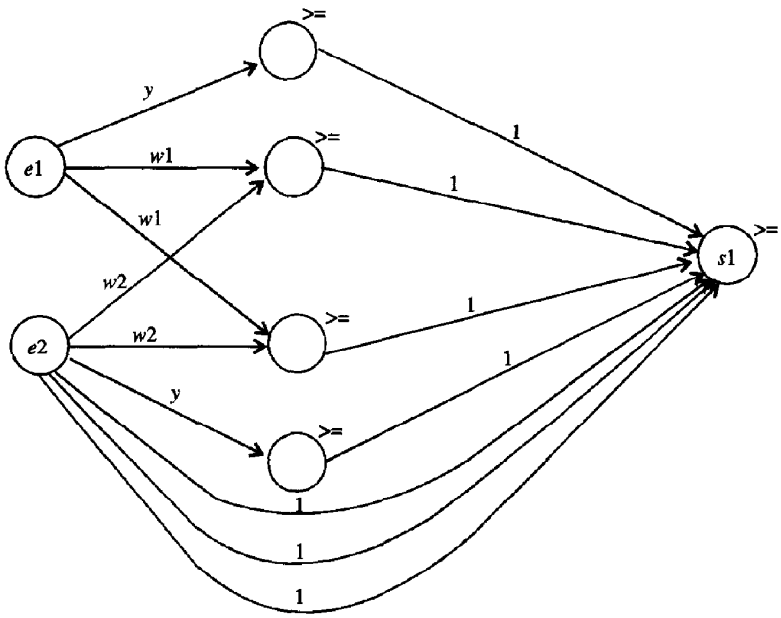
multiplication. Montrons que la résolution est universelle. Les "*blocs*" des formules $x=a$, ainsi que $bloc_{+}$, sont les mêmes que précédemment. Nous pouvons également joindre et coupler de la même manière. Nous en déduisons les "*blocs*" de la soustraction et de l'identité. Il nous reste à trouver les "*blocs*" de l'égalité et de la relation d'ordre. Par exemple, la formule $(x \leqslant y) \vee (z=0)$ est vraie si et seulement si au moins deux des tests suivants ont une réponse positive: $y-x \geqslant 0$, $z \geqslant 0$ et $-z \geqslant 0$. 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 $>=$):
$bloc_{\leqslant}$: c'est le *bloc* de la formule $(w_1 \leqslant w_2) \vee (z=0)$.
La tâche est $T=(1;-1)(0)$ et l'architecture est:



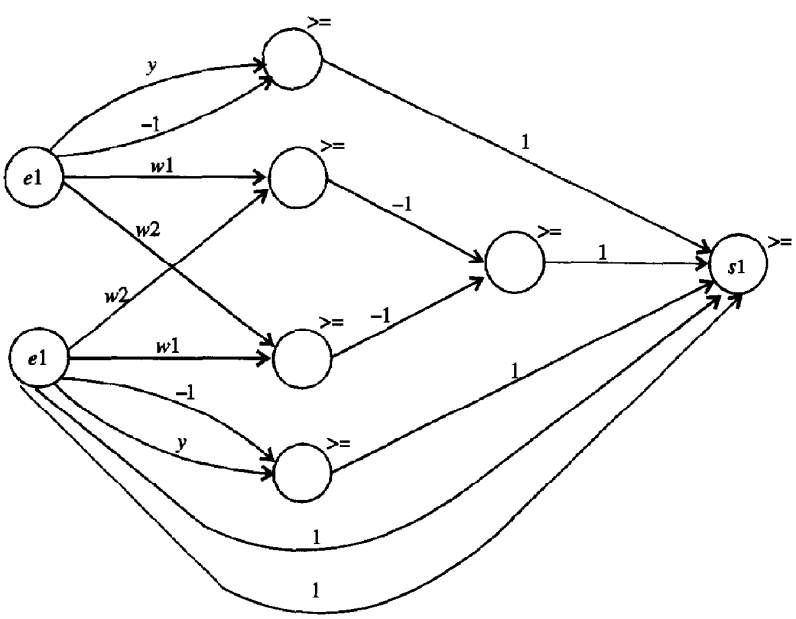
$bloc_{\neg\leqslant}$: c'est le *bloc* de la formule $(w_1 > w_2) \vee (y=1)$.
La tâche est $T=(1;-1)(0)$ et l'architecture est:



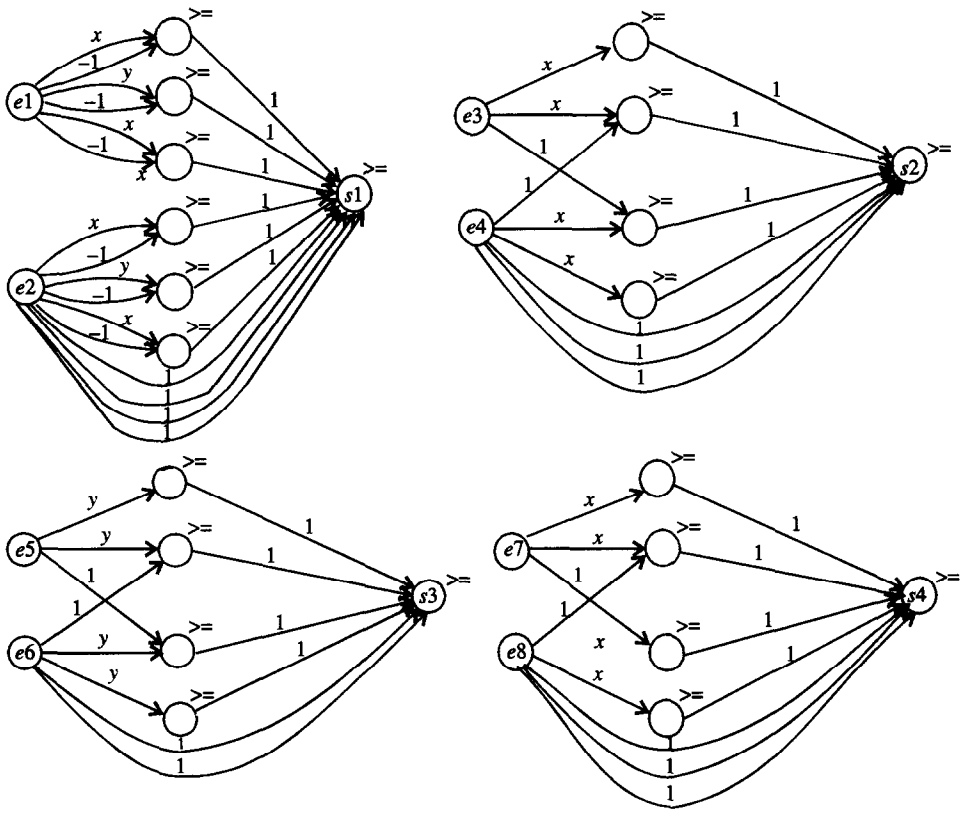
$bloc_{=}$: c'est le *bloc* de la formule $(w_1 = w_2) \vee (y = 0)$.
La tâche est $T = (1; -1)(0)$ et l'architecture est:



$bloc_{\neq}$: 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:



3.4. *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], Agrawal et 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 [5]: dans toute structure de langage relationnel fini et qui élimine les quanteurs, le problème de satisfaisabilité des formules sans quanteurs 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 ditincts. 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.

Remerciements

Je voudrais remercier Bruno Poizat pour avoir attiré mon attention sur ce sujet, et pour ses nombreux commentaires et ses suggestions très utiles.

References

[1] M. Agrawal, S. Biswas, Universal relations, in: Proc. of the 7th Structure in Complexity Theory Conf., 1992, pp. 207–220. To appear in Information and Computation.

[2] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: Np-completeness, recursive functions and universal machines, Bull. Amer. Math. Soc. 21(1) (July 1989) 1–46.

[3] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the theory of NP-Completeness, W.H. Freeman, 1979.

[4] J.B. Goode, Accessible telephone directories, J. Symbolic Logic 59(1), (March 1993) 92–105.

[5] B. Poizat, Les petits cailloux, Aleas éditeur, 1995.

[6] X.-D. Zhang, Complexity of Neural Network Learning in the Real Number Model (January 1992). Preprint, University of Massachusetts at Amherst.