Résolutions universelles pour des problèmes NP-complets.

Natacha Portier

Considérons une structure, c'est-à-dire un ensemble M , un nombre fini de fonctions f_i de M^{n_i} dans M, où n_i est un entier, et un nombre fini de relations r_j, où r_j est un sous-ensemble de M^{m_j} et m_j un entier. On s'intéresse aux ensembles définissables par cette structure.

Par exemple, la structure standard a pour ensemble celui des booléens, i.e. $\{0;1\}$, pour fonctions la conjonction booléenne $\wedge$ et la negation booléenne $\neg$, et pour relation l'égalité $\{(0;0);(1;1)\}$ tous les sous-ensembles de $\{0;1\}^n$ sont définissables pour la structure standard. Un autre exemple est la structure des réels : l'ensemble est $\mathbb{R}$, les fonctions sont l'addition, la soustraction et la multiplication, et les relations sont la relation d'ordre $\leq$ et l'égalité. Les ensembles définissables dans cette structure sont les ensembles semi-algébriques qu'étudie la géométrie algébrique réelle.

Définition : *Un problème X sur M* est un ensemble de mots sur M, c'est-à-dire un ensemble de suites finies d'eléments de M.

On étudie la question suivante :
Question : Etant donnés un mot $\overline{x}$ et un problème X, pouvons-nous savoir si $\overline{x} \in X$? Et si oui, combien de temps faut-il pour obtenir la réponse ?

Prenons un exemple : soit X_0 l'ensemble des mots $(a_0, \ldots, a_n)$ sur $\mathbb{R}$ tels que le polynôme $a_0 + a_1X + \ldots + a_nX^n$ admette une racine réelle. Si $\overline{a} = (a_0, \ldots, a_n)$, se demander si $\overline{a} \in X_0$, c'est se demander si le polynôme $a_0 + a_1X + \ldots + a_nX^n$ admet une racine réelle.

Définition : *Un problème X sur M est P_M*, i.e. polynomial pour la structure M, si la question $\overline{x} \in X$ peut être décidée en temps polynomial.

Le temps représente le nombre d'opérations (calcul de fonction ou test d'appartenance à une relation) à effectuer pour obtenir la réponse. Il est polynomial s'il est borné par un polynôme de la longueur du mot $\overline{x}$. A priori, un algorithme polynomial est plus rapide qu'un algorithme exponentiel. En pratique, ce n'est pas toujours le cas en raison de la taille des constantes.

Définition : *Un problème X sur M est NP_M* s'il existe un problème Y, qui est P_M, et tel que $\overline{x} \in X$ si et seulement s'il existe un mot $\overline{y}$ pas trop long et $\overline{yx} \in Y$. Plus précisément, il existe un polynôme qui borne la taille de $\overline{y}$ en fonction de celle de $\overline{x}$. Y est appelée résolution associée à X, et $\overline{y}$ est une solution pour $\overline{x}$.

Si nous reprenons l'exemple précédent, nous pouvons montrer que X_0 est NP_M. Soit Y_0 l'ensemble des $b\overline{a}$ tels que b soit racine du polynôme $a_0 + a_1X + \ldots + a_nX^n$. Nous avons : $\overline{a} \in X_0$ si et seulement si $a_0+a_1X+\ldots+a_nX^n$ a une racine réelle, c'est-à-dire si et seulement s'il existe b tel que $b\overline{a} \in Y_0$. D'autre part, vérifier si $b\overline{a} \in Y_0$

ne nécessite que le calcul de $a_0 + a_1 b + \ldots + a_n b^n$, c'est-à-dire $2n-1$ multiplications et n additions, donc un temps $3n-1$, qui est bien polynomial en n. Donc X_0 est NP_M. Notons que ce problème est aussi P_M grâce au théorème de Sturm.

Un problème X qui est P_M est NP_M. En effet, il suffit de prendre par exemple $Y = X$, et y est la suite vide. Qu'en est-il de la réciproque : tout problème NP_M est-il P_M ? C'est la question $P_M = NP_M$?, bien connue dans le cas standard. Nous connaissons des structures pour lesquelles on peut répondre non à la question, mais nous ne connaissons pas de structure, même triviale, pour laquelle nous pourrions répondre oui. C'est pour nous intéresser à cette question que nous allons définir les problèmes NP_M-*complets*. En effet, il faut et il suffit qu'un seul problème NP_M-*complet* soit P_M pour avoir l'égalité $P_M = NP_M$.

Définition : *Un problème X sur M est NP_M-complet* s'il est NP_M, et si pour tout problème X' qui est NP_M, et pour tout mot $\overline{x'}$ nous pouvons trouver rapidement un mot $\overline{x}$ tel que $\overline{x'} \in X'$ si et seulement si $\overline{x} \in X$. Rapidement signifie en un temps borné polynomialement par la taille de $\overline{x}$. Nous pouvons aussi écrire que X est NP_M-*complet* si tous les autres problèmes NP_M peuvent être réduits à X. Le problème X est alors plus compliqué que tous les autres problèmes NP_M.

Connaître un problème NP_M-*complet* et savoir le résoudre nous permet de répondre à la question $\overline{x} \in X$ pour tout problème X qui est NP_M et pour tout mot $\overline{x}$, mais cela ne nous permet pas de trouver les solutions $\overline{y}$ de $\overline{x}$. D'autre part, la méthode généralement utilisée pour montrer qu'un problème X est NP_M-*complet* est soit de réduire un problème X' déjà connu et NP_M-*complet* à X, soit de réduire tout problème $X' NP_M$ à X. Mais cela ne nous éclaire en rien sur la structure de X. Pour ces deux raisons, la notion d'universalité va nous être utile.

Définition : Soit X un problème NP_M, et soit Y une résolution associée à X. Nous dirons que Y *est universelle* si pour tout problème X' qui est NP_M, il existe une résolution Y' associée à X' qui se reduit à Y : pour tout mot $\overline{x'}$ on peut trouver rapidement un mot $\overline{x}$ tel que $\overline{x'} \in X'$ si et seulement si $\overline{x} \in X$, et tel que les solutions $\overline{y'}$ de $\overline{x'}$ se déduisent des solutions $\overline{y}$ de $\overline{x}$ par projection.

Bien sûr, si la résolution Y est universelle, alors X est NP_M-*complet*. Nous avons un théorème structurel qui nous permet de montrer qu'une résolution est universelle, et donc qu'un problème est NP_M-*complet* :

Théorème : *La résolution Y est universelle si et seulement si :*
(1.1) Pour chaque élément a de M, existe un mot $bloc_a$ dont les solutions donnent, par projection, $\{a\}$.
(1.2) Pour chaque fonction f, il existe un mot $bloc_f$ dont les solutions donnent, par projection, $\{x_1 x_2 \ldots x_p y$ tels que $y = f(x_1 x_2 \ldots x_p)\}$.
(1.3) Pour chaque relation r, il existe deux mots $btoc_r$ et $btoc_{\neg r}$ dont les solutions donnent, par projection, respectivement :

$\{x_1x_2\dots x_py$ tels que $x_1x_2\dots x_p \in r$ *et* $y \in M\} \cup M^{a_r}.0$ *et*
$\{x_1x_2\dots x_py$ tels que $x_1x_2\dots x_p \notin r$ *et* $y \in M\} \cup M^{a_r}.1$.
(1.4) Il existe un mot bloc dont les solutions donnent, par projection $\{0,1\}^3 - \{(0;0;0)\}$
(2) Il existe une fonction join calculable en temps polynomial qui à plusieurs mots $\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}$ *associe un mot dont les solutions sont les concaténés des solutions des* $\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}$.
(3) il existe une fonction cpl, calculable en temps polynomial qui, à un mot $\overline{x}$ *et à deux suites d'indices* $i_1, \dots, i_n$, *et* $j_1, \dots, j_n$ *associe un mot dont les solutions sont celles de* $\overline{x}$ *pour lesquelles les éléments d'indice* i_k *et d'indice* j_k *sont égaux.*

Appliquons immédiatement ce théorème à un exemple, qui nous permettra de retrouver le résultat principal de [3], qui a inspiré notre travail. Nous nous plaçons dans la structure des réels $\mathbb{R}$ décrite plus haut. X est le problème 4-FEAS, c'est-à-dire l'ensemble des polynômes réels, à un nombre quelconque de variables, et de degré total au plus quatre, qui ont une racine réelle. Nous lui associons la résolution $Y : \overline{y}P$ appartient à Y si et seulement si P a n variables et $\overline{y} = y_1 \dots y_n$ est une racine de P. Montrons que Y est universelle.

(1.1) Pour chaque élement a de M, $bloc_a$ est "$x - a$"
(1.2) Pour l'addition, $btoc_+$ est "$x_1 + x_2 - y$"
Pour la soustraction, $bloc_-$ est "$x_1 - x_2 - y$"
Pour la multiplication, $bloc_\times$ est "$x_1 \times x_2 - y$"
(1.3) Pour l'égalité, $bloc_=$ est "$x_1 - x_2$" et $btoc\neg =$ est "$tx_1 - tx_2 - 1$". Dans ce dernier cas, nous ne gardons que les variables x_1 et x_2, et pas la variable t.
Pour la relation d'ordre, $bloc_\leq$ est "$x_2 - x_1 - t^2$" et $bloc_{\neg\leq}$ est "$t^2x_1 - t^2x_2 - 1$"
Dans les deux cas, nous ne gardons que les variables x_1 et x_2.
(1.4) *bloc* est "$[x_1(x_1-1)]^2 + [x_2(x_2-1)]^2 + [x_3(x_3-1)]^2 + [t - (x_1-1)(x_2-1)1^2 + [t(x_3-1)]^2$".
(2) Pour la fonction *join*, il faut renommer les variables des polynômes de manière à ce qu'elles soient toutes différentes. Puis, nous ajoutons des variables pour n'avoir plus que des équations de degré deux. Nous sommons alors leurs carrés. Nous obtenons bien une équation polynomiale de degré au plus quatre.
(3) Pour la fonction *cpl*, nous mettons le polynôme sous la forme d'une somme de carrés de polynômes de degré au plus deux, puis nous ajoutons à l'équation la somme des $(x_{i_k} - x_{j_k})^2$.

Il n'y a pas beaucoup de structures M pour lesquelles nous connaissions un problème NP_M-*complet*. Dans certains cas, nous connaissons des problèmes NP_M-*complets* qui n'ont pas de résolution universelle.

Bibliographie

[1] *M. Agrawal, S. Biswas*, Universal Relations. To appear in Information and Computation in its journal form (1992)

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

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

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

[5] *N. Portier*, Résolutions universelles pour des problèmes NP-complets. à paraître dans TCS

Natacha Portier
Institut Girard Desargues
Bâtiment du doyen Jean Braconnier (101)
43, boulevard du 11 novembre 1918
69622 Villeurbanne Cedex, France
portier@desargues.univ-lyonl.fr