Cours d'algorithmique du 14 décembre 1999
Natacha Portier

Le théorème de Stockmeyer et Meyer (1973)

Définition 1 *Une expression rationnelle arithmétique E et le sous-ensemble $L(E)$ de $\mathbf{N}$ qu'elle représente sont définis de la manière récursive suivante :*

- *Les expressions 0 et 1 sont des ERA(expressions rationnelles arithmétiques) et représentent respectivement $L(0) = \{0\}$ et $L(1) = \{1\}$.*
- *Si E_1 et E_2 sont des ERA alors $(E_1 \cup E_2)$ est une ERA et $L(E_1 \cup E_2) = L(E_1) \cup L(E_2)$.*
- *Si E_1 et E_2 sont des ERA alors $(E_1.E_2)$ est une ERA et $L(E_1.E_2) = L(E_1) + L(E_2) = \{x + y \mid x \in L(E_1) \text{ et } y \in L(E_2)\}$.*

  Si E est une ERA, nous noterons : $E^0 = 0$, $E^1 = E$ et pour tout n entier naturel $E^{n+1} = E^n.E$
- *Si E est une ERA alors (E^*) est une ERA et $L(E^*) = \bigcup_{n \geq 0} L(E^n)$.*

Exemples :

- $L(0^*) = \{0\}$
- $L(1^*) = \mathbf{N}$
- $L(1^n) = \{n\}$
- $(1^2)^*$ représente les entiers pairs.
- $(1^2)^*.1$ représente les entiers impairs.

Remarques :

- Nous prendrons des libertés avec les parenthésages des ERA quand il n'y a pas d'ambiguïté.
- Une ERA est une expression formelle, un mot sur un certain alphabet. Deux ERA sont égales si et seulement si elles s'écrivent exactement pareil.
- Deux ERA différentes peuvent représenter le même ensemble, par exemple 1^*, $(1^2)^* \cup 1.(1^2)^*$ et $0 \cup 1.1^*$ représentent $\mathbf{N}$.

  De même E^* et $(E^*)^*$ représentent le même ensemble.

– Propriété (admise) : Les ensembles représentés par les ERA sont exactement les unions finies de progressions arithmétiques.

Nous pouvons maintenant énoncer le théorème de Stockmeyer et Meyer (1973) :

Théorème 1 *Savoir si une ERA ne représente pas* $\mathbf{N}$ *tout entier est un problème NP-complet.*

Remarque : la taille de l'ERA est mesurée par le nombre de symboles 0 et 1 qui interviennent dans son écriture.

Nous allons commencer par montrer que le problème est NP-dur. Puis nous montrerons des lemmes qui nous permettrons de prouver que le problème est NP, ce qui achèvera la preuve de la NP-complétude.

Nous réduisons 3-SAT à ce problème pour montrer qu'il est NP-dur. Une formule 3SAT est de la forme $\wedge_{k=1}^{m} C_k$ où $C_k = (v_{k,1} \vee v_{k,2} \vee v_{k,3})$ avec $v_{k,i} \in X = \{x_1, \neg x_1, x_2, \neg x_2, x \ldots, x_n, \neg x_n, x\}$.

Soient $p_1, ..., p_n$ les n premiers nombres premiers (nous pouvons les obtenir par exemple grâce à un crible d'Eratosthène sur les entiers entre 1 et n^2 pour n suffisemment grand). Considérons le système :

$$\left\{ \begin{array}{c} z \equiv z_1 \quad [p_1] \\ \ldots \\ z \equiv z_n \quad [p_n] \end{array} \right.$$

Nous dirons que z est un code de $(z_1, \ldots, z_n)$ s'il satisfait ce système. D'après le théorème des restes chinois il a une solution. De plus, il existe une solution t comprise entre 0 et $\Pi p_i - 1$ et l'ensemble des solutions positives est l'ensemble des termes de la suite arithmétique de premier terme t et de raison Πp_i. Nous allons construire une ERA qui représente l'ensemble des entiers qui ne codent pas une solution de notre formule 3SAT fixée. Il peut y avoir deux raisons pour lesquelles $(z_1, \ldots, z_n)$ ne satisfait pas la formule. La première est qu'un des z_i ne soit pas un booléen, la deuxième est qu'une des clauses ne soit pas satisafaite.

L'entier z ne code pas un booléen si pour un certain i compris entre 1 et n, l'entier z_i n'est pas booléen, i.e. est compris entre 2 et $p_i - 1$. Donc z ne code pas un booléen s'il appartient au sous-ensemble représenté par :

$$E_0 = \bigcup_{1 \leq i \leq n,\ 2 \leq k \leq p_i} 1^k.(1^{p_i})^*$$

Si z n'est pas le code d'un uple qui satisfasse la formule alors il existe k compris entre 1 et m tel que $(z_1, \ldots, z_n)$ ne satisfasse pas C_k. Cela signifie que :

$$\begin{cases} z \equiv z_{i_1} \quad [p_{i_1}] \\ z \equiv z_{i_2} \quad [p_{i_2}] \\ z \equiv z_{i_3} \quad [p_{i_3}] \end{cases}$$

avec $C_k = (v_{k,1} \vee v_{k,2} \vee v_{k,3})$, $z_{i_1} = 0$ si $v_{k,1} = x_{i_1}$ et $z_{i_1} = 1$ si $v_{k,1} = \neg x_{i_1}$, $z_{i_2} = 0$ si $v_{k,2} = x_{i_2}$ et $z_{i_2} = 1$ si $v_{k,2} = \neg x_{i_2}$, $z_{i_3} = 0$ si $v_{k,3} = x_{i_3}$ et $z_{i_3} = 1$ si $v_{k,3} = \neg x_{i_3}$. Soit t_k un entier compris entre 0 et $p_{i_1}p_{i_2}p_{i_3}$ qui satisfasse ce système. Alors z n'est pas le code d'un uple qui satisfasse C_k s'il appartient au sous-ensemble représenté par :

$$E_k = 1^{t_i}.(1^{p_{i_1}p_{i_2}p_{i_3}})^*$$

Donc finalement z ne code pas une solution de la formule si et seulement s'il appartient au sous-ensemble représenté par l'expression $\bigcup_{1 \leq k \leq m} E_k \cup E_0$.
□

Montrons maintenant que le problème est NP. Pour cela, nous commençons par montrer un théorème.

Théorème 2 *Soit G un graphe orienté et S_d et S_f deux sommets différents de G. Existe t'il un entier n tel qu'il n'y ait pas de chemin orienté de longueur n de S_d à S_f ? Ce problème est NP.*

Pour montrer ce théorème nous utilisons trois lemmes. Le lemme suivant fait le lien entre ERA et graphes orientés et nous permet de montrer que le problème est NP :

Lemme 1 *Soit E une ERA de taille N. Il existe un graphe orienté G de taille au plus $3N - 1$ et deux sommets S_d et S_f tels que pour tout entier strictement positif n, il existe un chemin de longueur n de S_d à S_f si et seulement si n appartient à $L(E)$.*

La preuve se fait par induction sur l'ensemble des ERA. Pour chaque expression E de taille N nous construisons (G, S_d, S_f, e) où G est un graphe orienté de taille au plus $4N - 2$, S_d et S_f sont deux sommets du graphe, e est égal à 0 si 0 appartient à $L(E)$ et 1 sinon et pour tout entier strictement positif n il existe un chemin de longueur n de S_d à S_f si et seulement si n appartient à $L(E)$.

- Pour l'expression 0 le graphe a deux sommets S_d et S_f, aucune arête et e est égal à 0.

Pour l'expression 1 le graphe a deux sommets S_d et S_f, une arête de S_d à S_f et e est égal à 1.

Soient E' et E'' des ERA et supposons avoir construit (G', S'_d, S'_f, e') et (G'', S''_d, S''_f, e''). Nous construisons alors (G, S_d, S_f, e) pour l'expression E de la manière suivante.

- Si $E = E'^*$ alors le graphe G est le graphe G' auquel nous ajoutons des arêtes. Les sommets S_d et S_f sont égaux à S'_d et S'_f. Si dans G' il existe une arête d'un sommet S à S'_f alors dans G il y a une arête de S à S_d. Pour finir $e = 0$.

- Soit $E = E'.E''$. Le graphe G est l'union du graphe G' et du graphe G'' à laquelle nous ajoutons des arêtes. Les sommets S_d et S_f sont égaux à S'_d et S''_f. S'il y a dans G' une arête entre un sommet S' et S'_f alors nous ajoutons dans G une arête entre S' et S''_d. Si $e' = 0$ et s'il y a dans G'' une arête entre un S''_d et un sommet S'' alors nous ajoutons dans G une arête entre S_d et S''. Si $e'' = 0$ et s'il y a dans G' une arête entre un sommet S' et S'_f alors nous ajoutons dans G une arête entre S' et S_f. Enfin, e est nul si et seulement si e' et e'' sont nuls.

- Soit $E = E' \cup E''$. Le graphe G est l'union des graphes G' et G'' à laquelle nous ajoutons un nouveau sommet S_f ainsi que de nouvelles arêtes. Le sommet S_d est égal à S'_d. Si dans G'' il y a une arête entre S''_d et un sommet S'' alors nous ajoutons dans G une arête entre S_d et S''. Si dans G'' il y a une arête entre S''_d et S''_f alors dans G il y a une arête entre S_d etS_f. Si dans G' ou G'' il y a une arête entre un sommet S et S'_f ou S''_f alors dans G il y a une arête entre S et S_f. Enfin, e est nul si et seulement si e' ou e'' est nul.

□

Lemme 2 *Soit G un graphe orienté avec N sommets et S_d et S_f deux sommets différents de G. Soit n_0 un entier tel qu'il n'existe pas de chemin orienté de longueur n_0 de S_d à S_f. Alors il existe un entier n_1 strictement inférieur à 2^N tel qu'il n'existe pas de chemin orienté de longueur n_1 de S_d à S_f.*

Démonstration : Soit $T_n(S_d)$ l'ensemble des sommets S de G tels qu'il existe un chemin de longueur n de S_d à S. Si $T_n(S_d) = T_m(S_d)$ alors nous avons $T_{n+1}(S_d) = T_{m+1}(S_d)$ et donc pour tout k entier naturel $T_{n+k}(S_d) = T_{m+k}(S_d)$. Or il y a 2^N sous-ensembles possibles des sommets de G. Donc S_f n'appartient pas à un des $T_n(S_d)$ si et seulement s'il existe $n_0 < 2^N$ tel que S_f n'appartienne pas à $T_{n_0}(S_d)$.

$\square$

Lemme 3 *Soit G un graphe orienté avec N sommets et S_d et S_f deux sommets différents de G. Soit n un entier naturel écrit en base 2 : $n = \epsilon_0 + 2\epsilon_1 + \cdots + 2^p\epsilon_p$. Alors il existe un algorithme qui décide s'il existe un chemin de longueur n dans le graphe de S_d à S_f en temps polynomial en N et $p = \lfloor \log_2 n \rfloor$.*

Démonstration : Pour tout sommet s du graphe et pour tout entier naturel m on définit $T_m(s)$ comme l'ensemble des sommets du graphe accessible à partir de s par un chemin de longueur m.

On a la propriété :

$$T_{m+l}(s) = \bigcup_{x \in T_m(s)} T_l(x) = \bigcup_{x \in T_l(s)} T_m(x)$$

Donc

$$T_n(S_d) = \bigcup_{x_0 \in T_{\epsilon_0}(S_d)} T_{2\epsilon_1 + \cdots + 2^p\epsilon_p}(x_0)$$

et

$$T_n(S_d) = \bigcup_{x_0 \in T_{\epsilon_0}(S_d)} \bigcup_{x_1 \in T_{2\epsilon_1}(x_0)} \cdots \bigcup_{x_{p-1} \in T_{2^{p-1}\epsilon_{p-1}}(x_{p-2})} T_{\epsilon_p 2^p}(x_{p-1})$$

L'algorithme cherché est le suivant. Nous calculons récursivement $T_1(s)$ pour tous les sommets s, puis $T_2(s)$ pour tous les sommets s, $T_4(s)$ pour tous les sommets s et ainsi de suite jusqu'à $T_{2^p}(s)$. Nous en déduisons $T_n(S_d)$ grâce à la formule précédente. Remarquons que tous les $T_m(s)$ sont de cardinal au plus N.

$\square$

Nous pouvons maintenant montrer le Théorème 1.

Soit E une ERA de taille N. Nous construisons (et il est clair que l'algorithme est polynomial) un graphe orienté G de taille au plus $3N - 1$ tel que pour tout entier strictement positif n, il existe un chemin du sommet S_d au sommet S_f dans G si et seulement si n appartient à $L(E)$. Il est facile de vérifier si 0 appartient ou non à $L(E)$. S'il existe n non nul n'appartenant pas à $L(E)$, alors il existe n inférieur à 2^{3N-1} tel qu'il n'y ait pas de chemin de longueur n entre S_d et S_f. Si l'entier n est donné par son écriture en base 2 alors nous pouvons le vérifier en temps polynomial, ce qui prouve que le problème est NP.

$\square$

Références

[1] *Stockmeyer, L. J. and Meyer, A. R.*, Word problems requiring exponential time: preliminary report. Fifth Annual ACM Symposium on Theory of Computing (Austin, Tex., 1973), pp. 1–9, Assoc. Comput. Mach., New York, (1973)