Théorème de Cook

Manon Ruffini

Définition 1 (*SAT*)

Entrée : *Une formule du calcul propositionnel*
Sortie : *Oui lorsque φ est satisfiable*

Théorème 1

Le problème SAT est NP-complet

Preuve :

D'abord, le problème SAT et NP. En effet, étant donnée un valuation, on peut déterminer en temps polynômial si elle satisfait φ. Il suffit d'évaluer l'arbre correspondant à la formule depuis les feuilles jusqu'à la racine.

Maintenant, montrons que SAT est NP-dur. Pour ça, on va montrer que tout problème de NP se réduit à SAT.
Soit A un problème de NP, qui est décidé en temps polynômial par une machine de Turing non déterministe M à ruban bi-infini. Pour toute entrée w de M, montrons qu'il existe une formule φ_w de taille polynômial en la taille de w, telle que : φ_w est satisfiable ssi w est accepté par M.
Soit $n = |w|$. On peut supposer qu'il existe $k \in \mathbb{N}$, tel que M décide A en temps n^k. Quitte à modifier M, on suppose que tout calcul acceptant est de longueur exactement n^k (on peut ajouter des états, des transitions pour prolonger autant que nécessaires les calculs acceptants).
Comme la machine M fonctionne en temps n^k, elle utilise au plus n^k cellules sur le ruban. Les configurations sont alors au plus de longueur n^k, et on peut même supposer qu'elles sont exactement de longueur $2n^k - 1$ quitte à rajouter des symboles blancs sur le ruban.

On peut écrire un diagramme espace-temps (1) qui représente l'évolution du ruban au cours du temps. On note $*$ la position du curseur.

On veut écrire une fomule φ_w qui affirme que le calcul commence dans la **configuration initiale**, que le diagramme est **cohérent** avec la relation de transition et qu'on termine dans un **état acceptant**.
On note la machine $M = (Q, \Gamma, q_0, q_a, \delta)$, où Q est un ensemble d'états, Γ est l'alphabet, $q_0 \in Q$ est

t \ e	$-n^k$	...	-1	0	1	2 ...	n^k		état
0			w_{-1}	w_0 $*$	w_1	w_2			q_0
1			w_{-1}	w'_0	w_1 $*$	w_2			q_1
2			w_{-1}	w'_0	w'_1 ...	w_2 ... $*$			q_2 ...
n^k			a'	a	b	d	...		q_a

t = temps ; e = espace

FIGURE 1 – Graphe des configurations

l'état initial, $q_a \in Q$ est l'état acceptant, et $\delta : Q \times \Sigma \to \mathcal{P}(Q \times \Sigma \times \{\leftarrow, \rightarrow\})$ est l'ensemble des transitions.

Au départ, j'avais écrit le développement avec une machine de Turing à un ruban infini à une extrémité seulement. Je ne réécris pas tout, mais il faut adapter les indices des variables introduites dans la suite.

On introduit les variables suivantes :

- $\forall \gamma \in \Gamma, \forall 0 \leqslant i \leqslant n^k$, $\forall 0 \leqslant j \leqslant n^k$, $c_{\gamma,i,j} =$ "Le symbole γ est dans la case i au temps j"
- $\forall 0 \leqslant i, j \leqslant n^k$, $p_{i,j} =$ "La tête de lecture est sur la case i au temps j"
- $\forall q \in Q, \forall 0 \leqslant j \leqslant n^k$, $e_{q,j} =$ "La machine est dans l'état q au temps j"

On a donc : $(n^k+1)^2|\Gamma| + (n^k+1)^2 + (n^k+1)|Q|$ variables. Leur nombre est donc polynômial en n, qui est la taille de l'entrée ($|\Gamma|$ et $|Q|$ dépendent uniquement de la machine, qui est fixée).

Rappel : on veut écrire une formule qui affirme que le calcul commence dans la configuration initiale, que les transitions sont licites et qu'on termine dans un état acceptant. Pour ça, on définit les formules suivantes :

- Cohérence C :
  - A tout moment, chaque contient exactement un symbole (qui peut être le symbole blanc) :

$$C_1 = \bigwedge_{(i,j)\in[|0,n^k|]^2} \bigvee_{\gamma\in\Gamma} \left(c_{\gamma,i,j} \wedge \bigwedge_{\gamma'\neq\gamma} \neg c_{\gamma',i,j} \right)$$

  - A tout moment, la tête de lecture est exactement à une position :

$$C_2 = \bigwedge_{j\in[|0,n^k|]} \bigvee_{i\in[|0,n^k|]} \left(p_{i,j} \wedge \bigwedge_{i'\neq i} \neg p_{i',j} \right)$$

  - A tout moment, on est exactement dans un état :

$$C_2 = \bigwedge_{j\in[|0,n^k|]} \bigvee_{q\in Q} \left(e_{q,j} \wedge \bigwedge_{q'\neq q} \neg e_{q',j} \right)$$

  Finalement :

$$C = C_1 \wedge C_2 \wedge C_3$$

- Initialisation I :
  - L'état de départ est q_0 :

$$I_1 = e_{q_0,0}$$

  - Le ruban contient exactement le mot $w = \gamma_1 \ldots \gamma_n$:

$$I_2 = \left(\bigwedge_{1\leqslant i\leqslant n} c_{x_i,i_0} \right) \wedge \left(c_{\underset{i>n}{\#,i,0}} \right)$$

  - La tête de lecture est en position 0 :

$$I_3 = p_{0,0}$$

  Finalement :

$$I = I_1 \wedge I_2 \wedge I_3$$

- Calcul acceptant A :

$$A = e_{q_a,n^k}$$

- Transitions licites $(T_j)_{1\leqslant j\leqslant n^k}$: Soit $1 \leqslant j \leqslant n^k$:

— Les cases où n'était pas la tête de lecture n'ont pas changé :

$$T_j^1 = \bigwedge_{0 \leqslant i \leqslant n^k} \left(\neg p_{i,j-1} \Rightarrow \left(\bigwedge_{\gamma \in \Gamma} (c_{\gamma,i,j} \Leftrightarrow c_{\gamma,i,j-1}) \right) \right)$$

— La case sur laquelle était la tête de lecture, la position de la tête de lecture et l'état de M ont évolué selon une transition valide :

$$T_j^2 = \bigwedge_{i,q,\gamma} \left((e_{q,j-1} \wedge p_{i,j-1} \wedge c_{\gamma,i,j-1}) \Rightarrow \bigvee_{(q',\gamma',d') \in \delta(\gamma,q)} (e_{q',j} \wedge c_{\gamma',i,j} \wedge p_{i+d',\gamma}) \right)$$

Puis :

$$\forall 0 \leqslant j \leqslant n^k T_j = T_j^1 \wedge T_j^2$$

Enfin, on pose :

$$\varphi_w = C \wedge I \wedge A \wedge \bigwedge_{0 \leqslant j \leqslant n^k} T_j$$

La taille de φ_w est polynômiale en n et par construction : φ_w est satisfiable ssi il existe un chemin acceptant à partir de w.
Ainsi, SAT est bien NP-difficile.

Finalement, **SAT est NP-complet**.

Références

[1] Olivier Carton, *Langages formels*. Vuibert.

[2] Sylvain Perifel, *Compléxité algorithmique*. Ellipses, 2014.