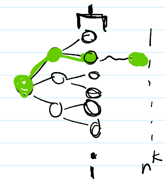
Cook-Levin
19 février 2021 11:11

Théorème : SAT est NP-complet
(de Cook-Levin)



① SAT $\in$ NP car une assignation sat. = certificat

② SAT est NP-difficile

Il faut montrer que $\forall L \in NP$, $L \leq_p SAT$

Soit $L \in NP$. Donc $\exists M$ une MT non-dét. dont le langage accepté est L. De plus, M accepte en temps $O(n^k)$, où $k \in \mathbb{N}$. On va supposer que M accepte en temps $\leq n^k$, $\forall w \in L$, $|w| = n$.

Une configuration de M est contenu de la bande + position de tête + état.
On peut représenter une config. par une chaîne de caractères sur alphabet $\underbrace{Q}_{\text{états}} \cup \underbrace{\Gamma}_{\text{symboles}}$

comme suit:

$$\boxed{\# \; s_1 \, s_2 \, s_3 \ldots s_{i-1} \; q \; s_i \ldots s_{n^k} \; \#}$$

On peut limiter le nb de caractères à n^k parce qu'on sait que M n'aura jamais besoin de plus pour accepter.

où $s_i \in \Gamma$ (incluant ␣)

$q \in Q$ état actuel

Position de q = position de tête

q voit s_i (symbole à droite)

Idée : un chemin acceptant = séq. de configs
$(c_1, c_2, \ldots, c_{n^k})$
où c_i peut mener à c_{i+1} selon les specs de M.

Représenter une séq. de cfg M valide par une formule booléene φ

On veut que $w \in L \Leftrightarrow \varphi \in SAT$

$\exists$ séq. config $(c_1, c_2, \ldots, c_{n^k})$ valide selon M qui accepte w

$\Leftrightarrow$

$\varphi \in SAT$

Tableau de config : séq. de n^k config pouvant être exécutée par M

$$\left.\begin{array}{ll}
1) & \# \; q_0 \, w_1 \, w_2 \, w_3 \ldots w_n \; ␣␣␣␣␣␣ \; \# \\
2) & \# \; w_1 \, q_1 \, w_2 \, w_3 \ldots w_n \; ␣␣␣␣␣␣ \; \# \\
 & \ldots \\
 & \# \; w_1 w_2 w_3 w_4 \ldots \; q_a \; ␣␣␣␣␣␣ \; \# \\
 & \# \; w_1 w_2 w_3 w_4 \ldots \; a \; q_b ␣␣␣␣␣ \; \# \\
 & \ldots \\
n^k) & \underbrace{\# \; a ␣ b c ␣ \ldots x \, y \, z \, q_i \, z \, z ␣ \; \#}_{n^k}
\end{array}\right\} n^k$$

Taille du tableau (représente une exec possible)
$= n^k \cdot n^k = n^{2k}$ (poly)

$x_{1,1,\#} = T$

$x_{1,2,q_0} = T$

$x_{2,3,w_2} = T$

Empêcher

$\cancel{x_{1,1,q_0}} = T$

Fait : $w \in L \Leftrightarrow \exists$ Tableau avec une rangée dans un état acceptant

But : encoder les tableaux possibles φ

$\exists$ tabl. ... acceptant $\Leftrightarrow \exists$ ... φ

$\underline{\text{But}}$: encoder les tableaux possibles φ

$\exists$ tableau qui accepte $\Leftrightarrow$ $\exists$ assignation sat. φ

Pour encoder les tableaux, on va représenter les tableaux acceptant comme suit dans φ

Un tableau est valide et acceptant $\Leftrightarrow$ $\underline{\varphi_{start}} \wedge \underline{\varphi_{cell}} \wedge \underline{\varphi_{accept}} \wedge \underline{\varphi_{move}}$ est sat.

où

→ φ_{start} = la config initiale est valide (état q_0 et w sur la bande)

→ φ_{cell} = chaque case du tableau a une valeur bien définie

→ φ_{accept} = il y a une cfg acceptante dans le tableau

→ φ_{move} = chaque paire de rangées $(i, i+1)$ représente une transition valide selon M.

Pour définir φ, on va utiliser les variables $x_{i,j,s}$

où $i \in \{1, .., n^k\}$ (rangée)

$j \in \{1, .., n^k\}$ (colonne/pos)

$s \in Q \cup \Gamma$

Interprétation : $x_{i,j,s} = T \Leftrightarrow$ le symbole du tableau en rangée i col. j est égal à s.

$\varphi_{start} = x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \wedge \dots \wedge x_{1,n+2,w_n} \wedge x_{1,n+3,\sqcup} \wedge \dots \overline{x_{1,n^k,\sqcup} \wedge x_{1,n^k+1,\#}}$

φ_{cell} = (pour chaque case i,j) $\left(\exists\, x_{i,j,s} = T \wedge \left(\overline{x_{i,j,s} = T \wedge x_{i,j,t} = T}, \forall s \neq t \right) \right)$

$$\bigwedge_{i,j \in \{1,..,n^k\}} \left[\left(\bigvee_{s \in Q \cup \Gamma} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s \neq t \\ s,t \in Q \cup \Gamma}} \overline{(x_{i,j,s} \wedge x_{i,j,t})} \right) \right]$$

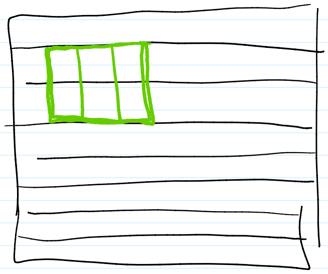
$$\left(\bigwedge_{1 \leq i \leq n} \varphi_i = \varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \right)$$

$$\varphi_{accept} = \bigvee_{i,j \in \{1..n^k\}} x_{i,j,q_{accept}}$$

φ_{move} = encoder que $\underline{\forall \text{ cfg possible à } i}$, cfg $i+1$ est une transition valide

nb exponentiel $\Rightarrow$ ne peut pas encoder ttes cfg possibles

Encoder des fenêtres légales 2×3 du tableau



ex :

a	b	q_1
a	q_2	b

On dit qu'une fenêtre 2×3 est légale s'il est possible que M contienne 2 cfgs successives contenant cette fenêtre

En d'autres termes, la fenêtre ne brise aucune règle des specs de M.

[On peut énumérer toutes les fenêtres légales et les encoder, en temps polynomial.

ex: M: $\delta(q_1, a) = \{(q_1, b, R)\}$ $\quad \delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$

Fenêtres légales :

a	a	b
a	a	b

a	q_1	b
a	a	q_2

a	a	q_1
a	a	b

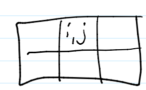
illégale

a	b	a
a	a	a

a	q_1	b
q_1	a	a

Lemme : si un tableau T démarre avec une cfg init valide et que toutes ses fenêtres 2×3 sont légales, alors T représente une exécution possible selon M.

φ_{move} = encoder que chaque fenêtre 2×3 = légale

$= \bigwedge_{i,j \in \{1,\dots,n^k\}}$ (fenêtre centrée sur i,j est légale) 

chaque fenêtre légale F possible peut être encodée avec une sous-formule φ_F contenant les 6 pos. de la fenêtre

$F =$

	i,j	
a	a	b
a	a	b

$x_{i,j-1,a} \wedge x_{i,j,a} \wedge x_{i,j+1,b} \wedge \dots$

$$= \bigwedge_{i,j \in \{1,\dots,n^k\}} \ \bigvee_{\text{fenêtres légales } F} (\varphi_F \text{ est à la position } i,j)$$

Si $\varphi = \varphi_{start} \wedge \varphi_{cell} \wedge \varphi_{accept} \wedge \varphi_{move}$, alors

φ encode le fait qu'on doit choisir un symbole à chaque i, j qui représente un tableau d'exéc acceptant

$w \in L \iff \exists$ tableau acceptant sur entrée $w \iff \varphi \in SAT$

$\varphi \in SAT$ veut dire qu'on <u>peut</u> choisir un symbole par cellule

$\Rightarrow$ SAT est NP-difficile