Chapitre 5 : Le théorème de Cook-Levin

ENSIIE - Théorie de la complexité

Dimitri Watel (dimitri.watel@ensiie.fr)

2023

Objectif du cours

Le problème (SAT ?)

Soit φ une formule booléenne sous forme normale conjonctive et $(x_1, x_2, \dots, x_n)$ ses variables. Existe-il une manière d'affecter VRAI ou FAUX aux variables de sorte que φ soit vrai ?

Théorème de Cook-Levin

(SAT ?) est NP-Complet.

Appartenance à NP

Lemme

(SAT ?) $\in$ NP

Car il existe une machine de Turing non déterministe résolvant (SAT ?) en temps polynomial.

NP-Difficulté

Lemme

(SAT ?) est NP-Difficile.

Pour le démontrer, montrons que, pour tout problème $\Pi \in NP$, $\Pi \preccurlyeq (SAT?)$.

Simuler une machine de Turing avec (SAT ?)

Soit $\Pi \in \mathrm{NP}$, il existe un polynome p et une machine de Turing $\mathcal{M}$ non déterministe résolvant Π avec une complexité $O(p(n))$. Simulons $\mathcal{M}$ avec une formule booléenne φ de taille polynomiale de sorte que $\mathcal{M}$ s'arrête sur un état acceptant si et seulement si φ est satisfiable.

Simuler une machine de Turing avec (SAT ?)

Simuler une machine de Turing avec (SAT ?)

$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

$$\begin{aligned}
\neg T_{-1,0,0} &= T_{-1,1,0} = \neg T_{-1,B,0} = \top \\
T_{0,0,0} &= \neg T_{0,1,0} = \neg T_{0,B,0} = \top \\
\neg T_{1,0,0} &= \neg T_{1,1,0} = T_{1,B,0} = \top \\
\neg T_{2,0,0} &= T_{2,1,0} = \neg T_{2,B,0} = \top
\end{aligned}$$

…	1	0		1	…
	-1	0	1	2	

Simuler une machine de Turing avec (SAT ?)

$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

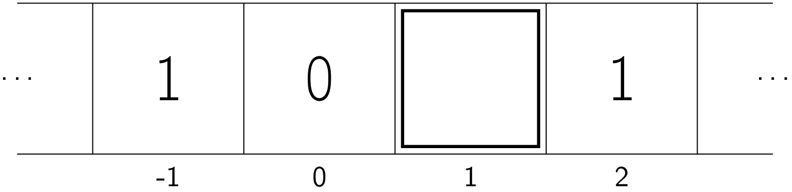
$$\begin{array}{llll}
\neg T_{-1,0,0} = T_{-1,1,0} = \neg T_{-1,B,0} = \top & \neg H_{-1,0} = H_{0,0} = \top \\
T_{0,0,0} = \neg T_{0,1,0} = \neg T_{0,B,0} = \top & \neg H_{1,0} = \neg H_{2,0} = \top \\
\neg T_{1,0,0} = \neg T_{1,1,0} = T_{1,B,0} = \top & \\
\neg T_{2,0,0} = T_{2,1,0} = \neg T_{2,B,0} = \top &
\end{array}$$

…	1	$\boxed{0}$		1	…
	-1	0	1	2	

Simuler une machine de Turing avec (SAT ?)

$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

$$\begin{array}{llll}
\neg T_{-1,0,1} = T_{-1,1,1} = \neg T_{-1,B,1} = \top & & \neg H_{-1,1} = \neg H_{0,1} = \top \\
T_{0,0,1} = \neg T_{0,1,1} = \neg T_{0,B,1} = \top & & H_{1,1} = \neg H_{2,1} = \top \\
\neg T_{1,0,1} = \neg T_{1,1,1} = T_{1,B,1} = \top & & \\
\neg T_{2,0,1} = T_{2,1,1} = \neg T_{2,B,1} = \top & &
\end{array}$$



Simuler une machine de Turing avec (SAT ?)

$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

$$\begin{array}{llll}
\neg T_{-1,0,2} = & T_{-1,1,2} = & \neg T_{-1,B,2} = \top & \neg H_{-1,2} = \neg H_{0,2} = \top \\
T_{0,0,2} = & \neg T_{0,1,2} = & \neg T_{0,B,2} = \top & H_{1,2} = \neg H_{2,2} = \top \\
\neg T_{1,0,2} = & \neg T_{1,1,2} = & T_{1,B,2} = \top & \\
\neg T_{2,0,2} = & T_{2,1,2} = & \neg T_{2,B,2} = \top &
\end{array}$$

…	1	0	$\square$	1	…
	-1	0	1	2	

$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

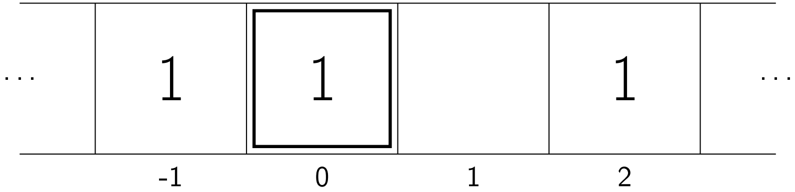
$$
\begin{array}{llll}
\neg T_{-1,0,3} = & T_{-1,1,3} = & \neg T_{-1,B,3} = \top & \neg H_{-1,3} = H_{0,3} = \top \\
T_{0,0,3} = & \neg T_{0,1,3} = & \neg T_{0,B,3} = \top & \neg H_{1,3} = \neg H_{2,3} = \top \\
\neg T_{1,0,3} = & \neg T_{1,1,3} = & T_{1,B,3} = \top & \\
\neg T_{2,0,3} = & T_{2,1,3} = & \neg T_{2,B,3} = \top &
\end{array}
$$

...	1	$\boxed{0}$		1	...
	-1	0	1	2	

Simuler une machine de Turing avec (SAT ?)

$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

$$\begin{array}{llll}
\neg T_{-1,0,4} = & T_{-1,1,4} = \neg T_{-1,B,4} = \top & \neg H_{-1,4} = & H_{0,4} = \top \\
\neg T_{0,0,4} = & T_{0,1,4} = \neg T_{0,B,4} = \top & \neg H_{1,4} = & \neg H_{2,4} = \top \\
\neg T_{1,0,4} = & \neg T_{1,1,4} = T_{1,B,4} = \top & & \\
\neg T_{2,0,4} = & T_{2,1,4} = \neg T_{2,B,4} = \top & &
\end{array}$$

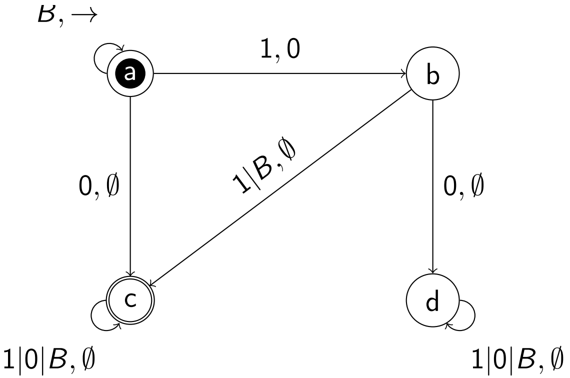


$T_{i,s,k}$: vrai ssi le symbole s est sur la case i à l'étape k.
$H_{i,k}$: vrai ssi la tête est sur la case i à l'étape k.

$$
\begin{array}{llll}
\neg T_{-1,0,5} = & T_{-1,1,5} = & \neg T_{-1,B,5} = \top & \neg H_{-1,5} = \; H_{0,5} = \top \\
\neg T_{0,0,5} = & \neg T_{0,1,5} = & T_{0,B,5} = \top & \neg H_{1,5} = \neg H_{2,5} = \top \\
\neg T_{1,0,5} = & \neg T_{1,1,5} = & T_{1,B,5} = \top & \\
\neg T_{2,0,5} = & T_{2,1,5} = & \neg T_{2,B,5} = \top &
\end{array}
$$

…	1	$\boxed{}$		1	…
	-1	0	1	2	

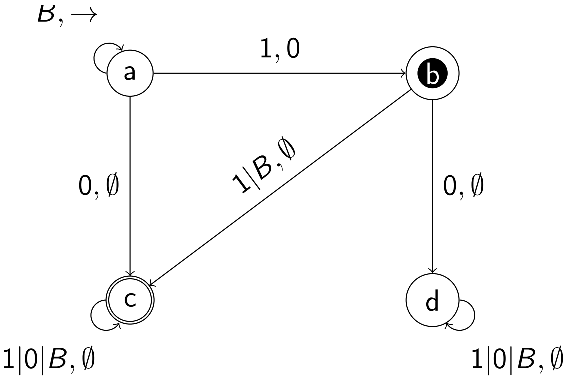
Simuler une machine de Turing avec (SAT ?)



$R_{q,k}$: vrai ssi le registre d'état pointe sur l'état q à l'étape k.

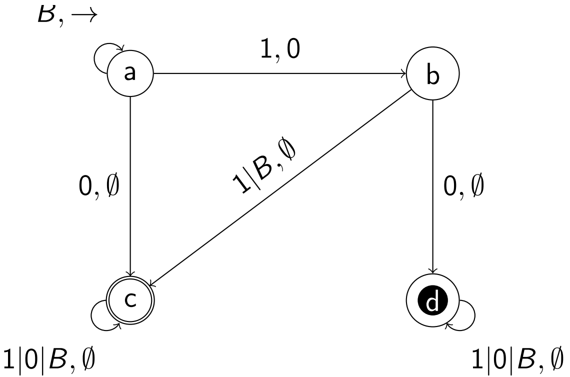
$$R_{a,0} = \neg R_{b,0} = \neg R_{c,0} = \neg R_{d,0} = \top$$

Simuler une machine de Turing avec (SAT ?)



$R_{q,k}$: vrai ssi le registre d'état pointe sur l'état q à l'étape k.

$$\neg R_{a,1} = \quad R_{b,1} = \neg R_{c,1} = \neg R_{d,1} = \top$$



$R_{q,k}$: vrai ssi le registre d'état pointe sur l'état q à l'étape k.

$$\neg R_{a,2} = \neg R_{b,2} = \neg R_{c,2} = \quad R_{d,2} = \top$$

Petite idée

Fait

Si la machine calcule durant $p(n)$ operations, elle utilise au plus $p(n)$ cases de la bande.

- $T_{i,s,k} : i \in [-p(n), p(n)], s \in \Sigma = \{0, 1, B\}, k \in [0, p(n)]$
- $H_{i,k} : i \in [-p(n), p(n)], k \in [0, p(n)]$
- $R_{q,k} : q \in V, k \in [0, p(n)]$

Le nombre de variables est polynomial par rapport à la n. (La taille de V ne dépend que de Π, pas de n.)

Clauses : Exclusion des variables

$$\varphi = \varphi_{excl} \wedge \cdots$$

$$\varphi_{excl} =$$

$$\bigwedge_{i=-p(n)}^{p(n)} \bigwedge_{s\in\Sigma} \bigwedge_{\substack{s'\in\Sigma \\ s\neq s'}} \bigwedge_{k=0}^{p(n)} T_{i,s,k} \Rightarrow \neg T_{i,s',k}$$

$$\wedge \bigwedge_{q\in V} \bigwedge_{\substack{q'\in V \\ q\neq q'}} \bigwedge_{k=0}^{p(n)} R_{q,k} \Rightarrow \neg R_{q',k}$$

$$\wedge \bigwedge_{i=-p(n)}^{p(n)} \bigwedge_{\substack{i'=-p(n) \\ i\neq i'}}^{p(n)} \bigwedge_{k=0}^{p(n)} H_{i,k} \Rightarrow \neg H_{i',k}$$

Clauses : Initialisation des variables

$$\varphi = \varphi_{excl} \wedge \varphi_{init} \wedge \cdots$$

Si l'entrée est x, avec $|x| = n$ et x_i le $i+1^{\text{e}}$ symbole de x si $i \in [0, n-1]$ et 0 sinon. Si l'état initial est q.

$$\varphi_{init} = \bigwedge_{i=-p(n)}^{p(n)} T_{i,x_i,0} \wedge R_{q,0} \wedge H_{0,0}$$

Clauses : S'arrêter sur un état acceptant

$$\varphi = \varphi_{excl} \wedge \varphi_{init} \wedge \varphi_{accept} \wedge \cdots$$

F_Y : Ensemble des états acceptants

$$\varphi_{accept} = \bigvee_{q \in F_Y} \bigvee_{i=0}^{p(n)} R_{q,i}$$

Clauses : Seule la case pointée par la tête peut changer

$$\varphi = \varphi_{excl} \wedge \varphi_{init} \wedge \varphi_{accept} \wedge \varphi_{tape} \wedge \cdots$$

$$\varphi_{tape} = \bigwedge_{i=-p(n)}^{p(n)} \bigwedge_{s \in \Sigma} \bigwedge_{k=0}^{p(n)-1} T_{i,s,k} \wedge \neg H_{i,k} \Rightarrow T_{i,s,k+1}$$

Clauses : transitions

$$\varphi = \varphi_{excl} \wedge \varphi_{init} \wedge \varphi_{accept} \wedge \varphi_{tape} \wedge \varphi_{trans}$$

$r(q, q')$: symbole de lecture de (q, q').

$$\varphi_{trans} = \bigwedge_{i=-p(n)}^{p(n)} \bigwedge_{s\in\Sigma} \bigwedge_{q\in V} \bigwedge_{k=0}^{p(n)} \left(R_{q,k} \wedge H_{i,k} \wedge T_{i,s,k} \Rightarrow \bigvee_{\substack{(q,q')\in A \\ r(q,q')=s}} \varphi_{q,q',i,k,s} \wedge R_{q',k+1} \right)$$

$\varphi_{q,q',i,k,s}$ dépend de l'actin de (q, q').

Si l'action de (q, q') est d'écrire s' sur la case en cours.

$$\varphi_{q,q',i,k,s} = T_{i,s',k+1} \wedge H_{i,k+1}$$

Clauses : transition de déplacement

Si l'action de (q, q') est de déplacer la tête

vers la droite :$\varphi_{q,q',i,k,s} = T_{i,s,k+1} \wedge H_{i+1,k+1}$

vers la gauche :$\varphi_{q,q',i,k,s} = T_{i,s,k+1} \wedge H_{i-1,k+1}$

Clauses : transition sans effet

Si l'action de (q, q') est de ne rien faire.

$$\varphi_{q,q',i,k,s} = T_{i,s,k+1} \wedge H_{i,k+1}$$