Chapter 5 : The Cook-Levin theorem

ENSIIE - Computational complexity theory

Dimitri Watel (dimitri.watel@ensiie.fr)

2023

Objective of the course

The (SAT ?) problem

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

The Cook-Levin theorem

(SAT ?) is NP-Complete.

Belonging to NP

Lemma

(SAT ?) $\in$ NP

Because there exists a non deterministic Turing machine that solve (SAT ?) in polynomial time.

NP-Hardness

Lemma

(SAT ?) is NP-Hard.

To prove this lemma, we will show that $\Pi \in NP$, $\Pi \preccurlyeq (SAT?)$.

Simulate a Turing machine with (SAT ?)

Let $\Pi \in$ NP, there exists a polynomial p and a non-deterministic Turing machine $\mathcal{M}$ solving Π with complexity $O(p(n))$.
We will simulate $\mathcal{M}$ with a boolean formula φ of polynomial size such that $\mathcal{M}$ stops on an accepting state if and only if φ can be satisfied.

Simulate a Turing machine with (SAT ?)

Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step k.

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

…	1	0		1	…
	-1	0	1	2	

Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step 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	

Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step k.

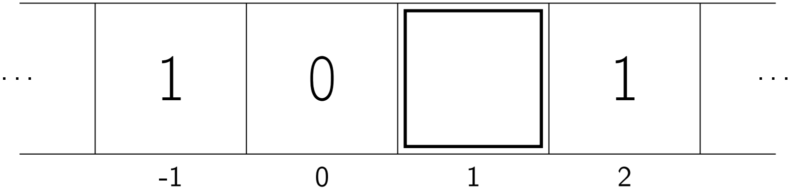
$$
\begin{array}{ll}
\neg T_{-1,0,1} = T_{-1,1,1} = \neg T_{-1,B,1} = \top & \neg H_{-1,1} = \neg H_{0,1} = \top \\
T_{0,0,1} = \neg T_{0,1,1} = \neg T_{0,B,1} = \top & H_{1,1} = \neg H_{2,1} = \top \\
\neg T_{1,0,1} = \neg T_{1,1,1} = T_{1,B,1} = \top & \\
\neg T_{2,0,1} = T_{2,1,1} = \neg T_{2,B,1}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

…	1	0	$\square$	1	…
	-1	0	1	2	

Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step k.

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



Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step k.

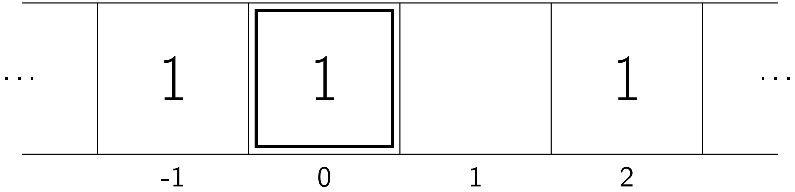
$$\begin{array}{llll}
\neg T_{-1,0,3} = & T_{-1,1,3} = & \neg T_{-1,B,3} = \top & \neg H_{-1,3} = \quad H_{0,3} = \top \\
T_{0,0,3} = & \neg T_{0,1,3} = & \neg T_{0,B,3} = \top & \neg H_{1,3} = \neg H_{2,3} = \top \\
\neg T_{1,0,3} = & \neg T_{1,1,3} = & T_{1,B,3} = \top & \\
\neg T_{2,0,3} = & T_{2,1,3}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

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

Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step k.

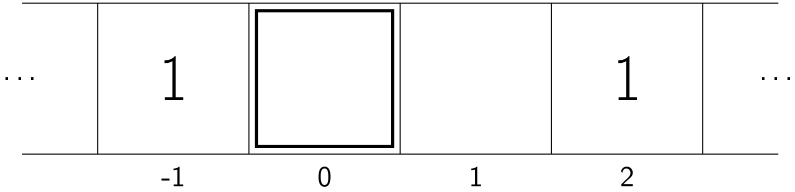
$$\begin{array}{llllll}
\neg T_{-1,0,4} = & T_{-1,1,4} = & \neg T_{-1,B,4} = \top & & \neg H_{-1,4} = & H_{0,4} = \top \\
\neg T_{0,0,4} = & T_{0,1,4} = & \neg T_{0,B,4} = \top & & \neg H_{1,4} = & \neg H_{2,4} = \top \\
\neg T_{1,0,4} = & \neg T_{1,1,4} = & T_{1,B,4} = \top & & & \\
\neg T_{2,0,4} =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



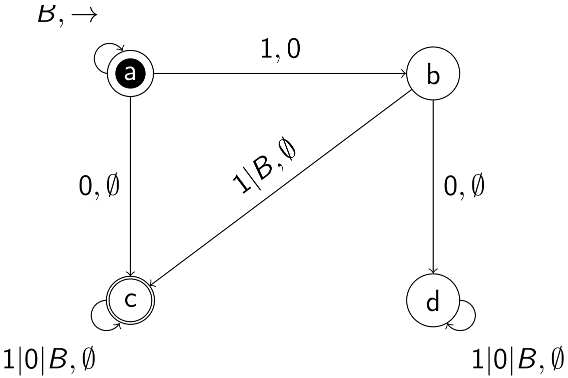
Simulate a Turing machine with (SAT ?)

$T_{i,s,k}$: true iff the symbol s is written on the cell i at step k
$H_{i,k}$: true iff the head reads the cell i at step k.

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



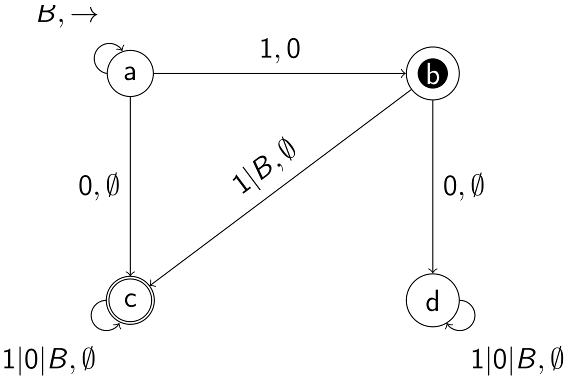
Simulate a Turing machine with (SAT ?)



$R_{q,k}$: true iff the state register points to the state q at step k.

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

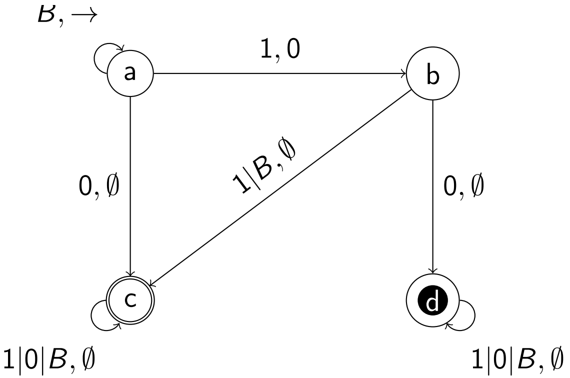
Simulate a Turing machine with (SAT ?)



$R_{q,k}$: true iff the state register points to the state q at step k.

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

Simulate a Turing machine with (SAT ?)



$R_{q,k}$: true iff the state register points to the state q at step k.

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

Idea

Fact

If the machine computes during $p(n)$ steps, it uses at most $p(n)$ cells of the tape.

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

The number of variables is polynomial in n (the size of V depends only on Π, not on n.)

Clauses: variables exclusion

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

$$\varphi_{excl} =$$

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

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

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

Clauses: variables initialisation

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

If the input is x where $|x| = n$ and where x_i is the $i+1$-th symbol of x if $i \in [0, n-1]$ and 0 otherwise. If the input state is q.

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

Clauses: stop on an accepting state

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

F_Y : Set of accepting states

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

Clauses: on the cell pointed to by the head may change

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

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

Clauses: transitions

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

$r(q, q')$: reading symbol of (q, q').

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

$\varphi_{q,q',i,k,s}$ depends on the action of (q, q').

Clause: writing transition

If the action of (q, q') consists in writing s' on the current cell.

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

Clause: moving transition

If the action of (q, q') consists in moving the head

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

to the left:$\varphi_{q,q',i,k,s} = T_{i,s,k+1} \wedge H_{i-1,k+1}$

Clause: transition with no effect

If the action of (q, q') consists in doing nothing.

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