

# THEOREME DE COOK

Leçons 913, 915, 928

Références Introduction à la calculabilité, Wolper p.185  
LFCC, Couston p.203

## Théorème

SAT est NP-complet

## Résumé

I - SAT  $\in$  NP

II - Comment décrire une machine de Turing?

III - Définition d'un ensemble de propositions pour représenter chacun des éléments de II en utilisant SAT.

I • On peut définir un algorithme non déterministe qui :

i - génère une fonction d'interprétation de façon non déterministe

ii - vérifie que cette fonction d'interprétation rend la formule vraie.

• Donc SAT  $\in$  NP.

II • Soit une machine de Turing

$$M = (Q, \Gamma, \Sigma, \Delta, s, B, f).$$

• Une exécution de  $M$  peut être décrite comme :

i - un tableau  $R$  de taille  $(p(n)+1, p(n)+1)^{(*)}$  qui, pour chaque configuration, donne le contenu du ruban.

ii - un vecteur  $Q$  donnant, pour chaque configuration, l'état.

iii - un vecteur  $P$  donnant, pour chaque configuration, la position de la tête de lecture.

iv - un vecteur  $C$  donnant le choix non déterministe effectué par  $M$  à chaque étape. C'est un nombre plus petit que  $c = \text{nb max de choix}$ .

taille de  
tableau  
 $p(n)+1$



Où  $p(n)$  est un polynôme bornant la complexité de  $\Pi$  et  $w = w_1 \dots w_n$  un mot de  $\Sigma^*$ .

- La machine  $M$  accepte  <sup>$w$</sup>  alors si et seulement si il existe une exécution de  $M$  sur  $w$  de longueur au plus  $p(n)$  qui mène à un état accepteur.

② notons que, la tête de lecture ne se déplaçant que d'une case à chaque étape, le nombre de symboles non blancs sur le ruban est aussi borné par  $p(n)+1$ .

III • Notre objectif est de montrer qu'il existe une transformation qui, à tout mot  $w$  et à tout langage LEXP, associe une instance de  $L_{SAT}$  (le langage de l'encodage des instances positives de SAT) qui est positive si et seulement si  $w \in L$ .

→ on veut donc associer à  $\Pi$  et  $w$  une instance de SAT qui est positive si et seulement si  $M$  accepte  $w$ .

• On cherche donc à produire une formule qui n'est satisfaite que par un contenu de  $R, Q, P$  et  $C$  définissant une exécution de  $\Pi$  qui accepte  $w$ . Introduisons alors une variable pour chaque case de ce tableau et pour chaque valeurs qu'elles peuvent prendre :

- i -  $r_{ij\alpha}$  pour  $0 \leq i, j \leq p(n)$  et  $\alpha \in \Gamma$
- ii -  $q_{ik}$  pour  $0 \leq i \leq p(n)$  et  $k \in Q$
- iii -  $p_{ij}$  pour  $0 \leq i, j \leq p(n)$
- iv -  $c_{ik}$  pour  $0 \leq i \leq p(n)$  et  $1 \leq k \leq r$

• Il y a ici  $O(p(n)^2)$  propositions : c'est bien polynomial !

- La convention utilisée sera donc que  $r_{ijk} = 1$  si  $R[i, j] = \alpha$  et  $q_{ik} = 1$  si  $Q[i] = k$ .

→ il faut donc s'assurer qu'il n'y a qu'un symbole par case du tableau.



Pour cela on définit

$$* (1a) \bigwedge_{0 \leq i, j \leq p(n)} \left( \left( \bigvee_{\alpha \in \Gamma} r_{ij\alpha} \right) \wedge \left( \bigwedge_{\alpha \neq \alpha' \in \Gamma} (\neg r_{ij\alpha} \vee \neg r_{ij\alpha'}) \right) \right)$$

il y a un symbole en  $(i, j)$

si il y a un symbole, il ne peut pas y en avoir un autre.

+ on peut écrire des formules (1b) et (1c) similaires pour  $q$  et  $c$ .

Ces formules sont en FNC, de longueur  $O(p(n)^2)$ .

$$* (2) \bigwedge_{0 \leq i \leq p(n)} \left( \left( \bigvee_{0 \leq j \leq p(n)} p_{ij} \right) \wedge \left( \bigwedge_{\alpha(j \neq j') \leq p(n)} (\neg p_{ij} \vee \neg p_{ij'}) \right) \right)$$

la tête de lecture est quelque part

elle ne peut pas être à deux endroits différents

Ces formules sont en FNC, de longueur  $O(p(n)^3)$ .

→ il faut s'assurer que, de plus:

- la première configuration doit être la configuration initiale:

$$(3) \left( \bigwedge_{0 \leq j \leq n-1} r_{0jw_{j+1}} \wedge \bigwedge_{m \leq j \leq p(n)} r_{0jB} \right) \wedge q_{00} \wedge p_{00}$$

il est écrit  $w$  sur le ruban, suivi uniquement de blancs

état initial

tête de lecture au début.

- Chaque configuration est obtenue à partir de la précédente suivant la relation de transition de la machine de Turing; c'est-à-dire

• les cases n'étant pas sous la tête de lecture ne sont pas modifiées:

$$(4a) \bigwedge_{\substack{0 \leq i < p(n) \\ 0 \leq j \leq p(n) \\ \alpha \in \Gamma}} (r_{ij\alpha} \wedge \neg p_{ij}) \supset r_{(i+1)j\alpha} \text{ soit } \bigwedge_{\substack{0 \leq i < p(n) \\ 0 \leq j \leq p(n) \\ \alpha \in \Gamma}} (\neg r_{ij\alpha} \vee p_{ij} \vee r_{(i+1)j\alpha})$$

soit il est écrit autre chose qu' $\alpha$ , soit il y a la tête de lecture, soit il y a  $\alpha$  après.



• la case sous la tête de lecture, l'état et la position de la tête de lecture sont modifiés en adéquation avec  $\Delta$ , la relation de transition. On remarque que grâce à  $c_{ik}$ , la relation s'exprime bien :

dans la configuration  $i$  }  
 l'état  $k$  }  $\Delta$  précise alors de façon unique  
 la position de la tête de lecture  $j$  } l'état suivant  $k'$   
 le symbole lu  $a$  } le symbole à écrire  $a'$   
 le choix non déterministe } le déplacement de la tête  $d \in \{\pm 1\}$ .

Cela donne :

(4b)

$$\begin{array}{l}
 0 \leq i < p(n) \\
 0 \leq j \leq p(n) \\
 a \in \Gamma \\
 k \in Q \\
 1 \leq d \leq r
 \end{array}$$

$$\begin{aligned}
 & ((q_{ik} \wedge p_{ij} \wedge r_{j\alpha} \wedge c_{ik}) \supset q_{(i+1)k'}) \\
 & \wedge ((q_{ik} \wedge p_{ij} \wedge r_{j\alpha} \wedge c_{ik}) \supset r_{(i+1)j\alpha'}) \leftarrow \text{on écrit } \alpha' \\
 & \wedge ((q_{ik} \wedge p_{ij} \wedge r_{j\alpha} \wedge c_{ik}) \supset p_{(i+1)(j+d)}) \leftarrow \begin{array}{l} \text{la tête se} \\ \text{déplace de } +1 \text{ ou } -1. \end{array}
 \end{aligned}$$

l'état suivant est  $k'$

sachant que  $(a \wedge b \wedge c \wedge d) \supset e$  s'écrit  $\neg a \vee \neg b \vee \neg c \vee \neg d \vee e$

Les formules (4a) et (4b) ont une longueur  $O(p(n)^2)$ .

→ enfin, il faut arriver dans un état acceptant :

(5)  $\bigvee_{\substack{0 \leq i < p(n) \\ k \in F}} q_{ik}$

• Conclusion la conjonction des formules (4a), (4b), (4c), (2), (3), (4a), (4b) et (5) est satisfaisable si et seulement si  $M$  accepte  $w$  : une  $\varphi$  satisfaisant donne une exécution de  $M$  acceptant  $w$  et inversement. De plus, les  $c_{ik}$  donnent les choix faits dans l'exécution. Enfin, la longueur de la formule est  $O(p(n)^3)$  donc elle peut être construite en temps polynomial. Elle doit ensuite être encodée, ce qui se fait aussi en temps polynomial : on a donc une transformation poly de L<sub>ENP</sub> dans L<sub>SAT</sub>.