# Théorème de Cook

**Problème SAT** : Etant donnée une formule de la logique propositionnelle, décider si elle est satisfiable.
ou
Etant donnée une formule de la logique propositionnelle, exciste-t-il une valuation des variables qui la satisfait.

**Théorème de Cook** : SAT est NP-complet

*Preuve :* Pour montrer qu'un problème est NP-complet il faut d'une part montrer qu'il est NP et d'autre part montrer qu'il est NP-difficile

*Rappel* P est NP-difficile si tous les problèmes NP se réduisent à P.

## 1) SAT est NP

- Une caractérisation des problèmes NP est l'existence d'un algo qui étant donnés une instance du problème et une proposition de solution décide s'il s'agit ou non d'une solution au pb en un temps polynomial en la taille de la proposition

- Etant donnés une formule $\Phi$ (instance du problème) et une valuation $v$ (solution proposée) on peut vérifier en temps linéaire si $v$ satisfait $\Phi$.
On procède récursivement sur la structure inductive de $\Phi$.
Chaque littéral représente un coût constant : un accès à la valuation.
Chaque connecteur ($\wedge$ ou $\vee$) représente un coût constant : une opération booléenne sur les évaluations récursives des sous-formules.
Finalement on a bien un coût de l'ordre de la taille de la formule puisqu'il est de l'ordre du nombre de noeuds et feuilles dans l'arbre de construction de la formule.

## 2) SAT est NP-difficile

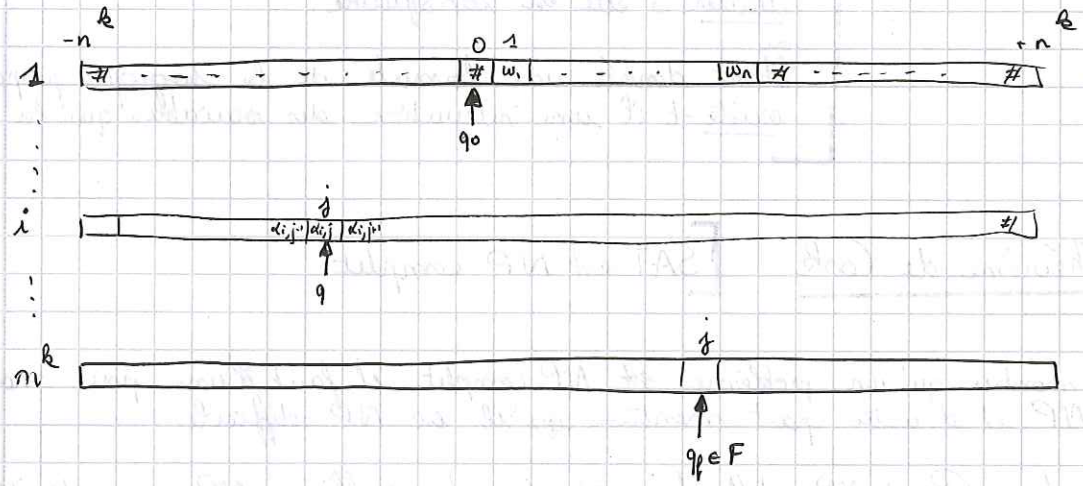
- On utilise la définition à l'aide des machines de Turing d'un problème NP quelconque puis on montre qu'on le réduit en SAT en un temps polynomial.

- Soit P un problème NP. Soit $w$ une instance de P. On note $n = |w|$
Soit $\mathcal{M} = (\Sigma, Q, F, \delta : Q \times \Sigma \to \mathcal{P}(Q \times \Sigma \times \{-1, 0, 1\}))$ une machine de Turing résolvant P.

$\Sigma$ : alphabet ; $Q$ : états ; $F$ : états finals ; $\delta$ : fonction de transition non déterministe

On veut traduire par une formule le fait que $M$ lisant $\omega$ atteint un état final. Ce calcul étant polynomial, convenons, quitte à ajouter des transitions nulles, qu'il se fait en $n^k$ étapes.

Représentons les configurations successives de $M$ (dans le cas où $P(\omega)$ a pour réponse OUI)




On pose $I = [\![1 .. n^k]\!]$
$J = [\![-n^k, n^k]\!]$
$\Sigma = \{0, 1, \#\}$

On introduit pour $(i,j,a) \in I \times J \times \Sigma$ la variable $x_{i,j,a}$

$(i,j,q) \in I \times J \times Q$ ——— $e_{i,j,q}$.

- $x_{i,j,a}$ représentera le fait que le caractère $a$ soit écrit en position $j$ à l'étape $i$
- $e_{i,j,q}$ ——— la tête de lecture soit à la position $j$ et dans l'état $q$ à l'étape $i$

Traduisons l'existence d'une telle suite de configurations

$$\varphi_0 = \varphi_{\substack{\text{unicité} \\ \text{caractère}}} = \bigwedge_{i\in I} \bigwedge_{j\in J} \bigwedge_{a\in\Sigma} \bigwedge_{a'\in\Sigma} (\neg x_{i,j,a} \vee \neg x_{i,j,a'})$$

traduit le fait que le caractère sur le ruban à la $m$-ième position et une $m$-ième étape est bien unique (ce fixé)

$$\varphi_1 = \varphi_{\substack{\text{existence} \\ \text{caractère}}} = \bigwedge_{i\in I} \bigwedge_{j\in J} \bigvee_{a\in\Sigma} x_{i,j,a}$$

traduit le fait qu'il y a bien un caractère sur le ruban dans chaque case et à chaque étape (ce caractère pouvant être # codant une case vide)

$$\varphi_2 = \varphi_{\substack{\text{existence et unicité} \\ \text{de l'état et la position}}} = \bigwedge_{i\in I} \bigvee_{q\in Q} \bigvee_{j\in J} \Big( \underbrace{e_{i,j,q}}_{\text{existence}} \wedge \underbrace{\bigvee_{\substack{j'\in J \\ (j,q)\neq(j',q')}} \bigvee_{q'\in Q} \neg e_{i,j',q'}}_{\text{unicité}} \Big)$$

traduit le fait qu'à chaque étape la tête de lecture est bien dans une position et une seule, dans un état et un seul.

$$\varphi_3 = \varphi_{\text{final}} = \bigvee_{j\in J} \bigvee_{q_f\in F} e_{n^k,j,q_f}$$

traduit le fait qu'à la fin, ie à l'étape $n^k$, la tête de lecture est à une position quelconque mais dans un état final

$$\varphi_4 = \varphi_{\substack{\text{première} \\ \text{ligne}}} = \bigwedge_{j=1}^{n} x_{1,j,\omega_j} \wedge e_{1,0,q_0}$$

traduit le fait que la configuration initiale est celle où la tête de lecture est juste à droite de la donnée, qui est bien écrite sur le ruban

Pour expliquer que les différentes configurations correspondent bien à des transitions valides on introduit des formules à paramètres intermédiaires.

Pour $(i,j,q,q',a,a',\varepsilon) \in I \times J \times Q^2 \times \Sigma^2 \times D$ avec $i \geq 2$ tel que $(q',a',\varepsilon) \in \delta(q,a)$

$$\Delta_{i,j,q,q',a,a',\varepsilon} = x_{i-1,j-\varepsilon,a} \wedge x_{i,j-\varepsilon,a'}$$ "là où il y avait a on a écrit a'"

$$\wedge e_{i-1,j-\varepsilon,q} \wedge e_{i,j,q'}$$ "on est passé de l'état q à l'état q' de la position $j-\varepsilon$ à la position $j$"

$$\wedge \bigwedge_{\substack{j' \\ j' \neq j-\varepsilon}} \bigwedge_{b \in \Sigma} x_{i-1,j',b} \to x_{i,j',b}.$$ "partout ailleurs le ruban n'a pas été changé"

Cette formule traduit la transition $\delta(q,a) \ni (q',a',\varepsilon)$ réalisée à l'étape $i$, la position courante passant de $j-\varepsilon$ à $j$.

$$\varphi_5 = \bigwedge_{i=2}^{n^k} \left( \bigvee_{(j,q,q',a,a',\varepsilon) \in J \times Q^2 \times \Sigma^2 \times D \text{ tq } (q',a',\varepsilon) \in \delta(q,a)} \Delta_{i,j,q,q',a,a',\varepsilon} \right)$$

traduit une succession valide de configurations.

Rq il s'agit d'une disjonction non vide car il existe bien des transitions $(q,a) \xrightarrow{\delta} (q',a')$ avec déplacement $\varepsilon \in D$.

On pose $\varphi(\omega) = \bigwedge_{i=0}^{5} \varphi_i$

- Si la réponse au problème $\mathcal{P}(\omega)$ est oui, alors cette succession de configurations de $\mathcal{A}$ existe et fournit alors naturellement une valuation satisfaisant $\varphi_\omega$ autrement dit $\varphi_\omega$ est alors une instance positive de SAT (ie la réponse de SAT($\varphi$) est oui).

- Réciproquement si $\varphi_\omega$ est une instance positive de SAT il existe une valuation qui satisfait $\varphi_\omega$ et cette valuation décrit une succession de configurations possibles pour $\mathcal{A}$, ce qui signifie qu'une des évaluations possibles de $\omega$ par $\mathcal{A}$ "répond oui", autrement dit $\omega$ est une instance positive de $\mathcal{P}$.

Ainsi $\mathcal{P}$ se réduit bien à SAT.

$\rightsquigarrow$ RQ cette réduction se fait en temps polynomial.

Le temps de calcul de $\varphi_\omega$ revient à son temps d'écriture c'est pourquoi on est ramené à montrer que le nombre de littéraux est polynomial en $|\omega| = n$.

$\mathrm{Card}(I) = n^k$ et $\mathrm{Card}(J) = 2 \times n^k$ (valeurs dépendant de l'instance $\omega$)

Notons $n_q = \mathrm{Card}(Q)$, $n_f = \mathrm{Card}(F)$ et $n_\Sigma = \mathrm{Card}\,\Sigma$ (ces valeurs sont des constantes au sens où elles ne sont relatives qu'à $\mathcal{A}$ et non à $\omega$)

$|\varphi_0| = 4 \times n_q^2 \times n^{2k}$ $\quad |\varphi_1| \leq n^k \times n_q \times 2n^k \times 2n^k \times n_q = 4 n_q \times n^{3k}$

...

Les $\varphi_i$ sont de taille polynomiale en $n = |\omega|$ donc la réduction se fait bien en temps polynomial.

Donc SAT est bien NP-difficile.