P vs NP

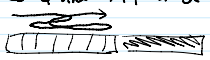
3 février 2022 10:13

P = langages décidables en temps poly par une MT

NP = " " " " " par une MT non-déterministe

P = NP ?

- Temps d'une MT : # de transitions faites avant l'arrêt
- Une MT M est en temps $O(f(n))$ si :
  $\forall$ entrée w, M prend un temps $O(f(n))$, où $|w| = n$
- Espace d'une MT : # de cellules distinctes utilisées



Espace $\leq$ Temps

DTIME($f(n)$) : un langage L est dans DTIME($f(n)$) s'il $\exists$ un MT qui décide L en temps $O(f(n))$

Classe de langages P (poly)

$$P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$$

= pbs résolubles en temps "raisonnable" (n^{1000} raisonnable)

ex : $L_+ = \{\langle a,b,c \rangle : a+b=c\} \in P$

PATH = $\{\langle G, s, t, k \rangle$: G est un graphe dans lequel le chemin le + court de s à t est $\leq k\}$

fouille en largeur $O(|V(G)| + |E(G)|)$ (sommets, arêtes) $\in P$

Dijkstra $O(|V(G)| \log |E(G)|)$

MT non-déterministe

Q : états

Σ : alphabet

$s \in Q$: état initial

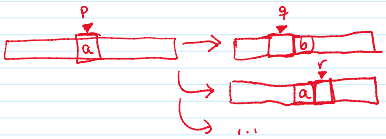
(blank) $\sqcup$: vide

$A \subseteq Q$: états acceptants

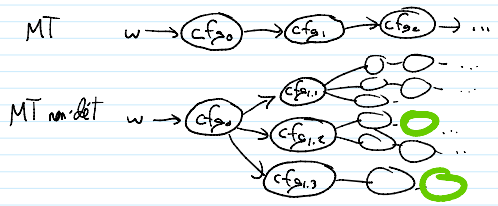
MT déf. $\delta : Q \setminus A \times \Sigma \to Q \times \Sigma \times \{L, R\}$

$$\delta \subseteq (Q \setminus A \times \Sigma) \times (Q \times \Sigma \times \{L, R\})$$

état - symbole — état - symbole - gauche/droite



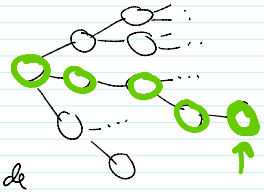
Intuition



Acceptation

Acceptation

Une MT non-dét. M accepte $w \in \Sigma^*$ s'il existe une séquence de choix de transitions qui mènent à un état acceptant.
Sinon, M rejette w.



Temps d'une MT non-dét

Sur entrée w, le temps de M:

- si w est accepté, le temps de M est le # de transitions sur le chemin le + court menant à l'acceptation de M.
- si w n'est pas accepté, le temps de M est 1.

$NTIME(f(n))$: un langage L est dans $NTIME(f(n))$ s'il existe une MT non-dét. dont le langage accepté est L et dont le temps est tjrs $O(f(n))$.

$$NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$$

ex: $SAT = \{\varphi : \varphi$ est une formule booléenne satisfaisable$\}$

φ: $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_4 \vee \bar{x}_1) \wedge (\bar{x}_1 \vee \bar{x}_4)$

F F V ... F ∧ ? ∧ ?

$\varphi \in SAT$ si on peut assigner $x_i = T$ ou $x_i = f$ $\forall x_i$ de façon à ce que φ évalue à T.

$x_1 = V$, $x_2 = F$, $x_3 = F$, $x_4 = F$

$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_4 \vee \bar{x}_1) \wedge (\bar{x}_1 \vee \bar{x}_4)$

$(V \vee F \vee V) \wedge (V \vee F \vee F) \wedge (F \vee V)$

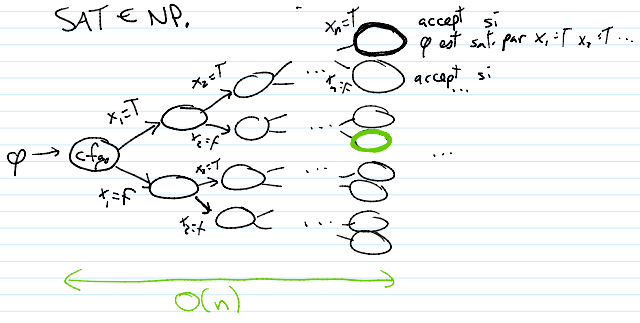
$V \wedge V \wedge V$

$V \quad \varphi \in SAT$

$x_1 \wedge \bar{x}_1 \notin SAT$

On ne sait pas si $SAT \in P$.

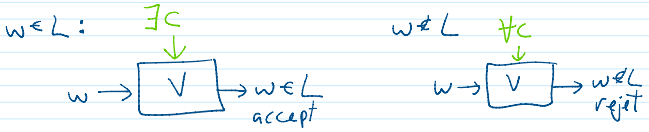
$SAT \in NP$.



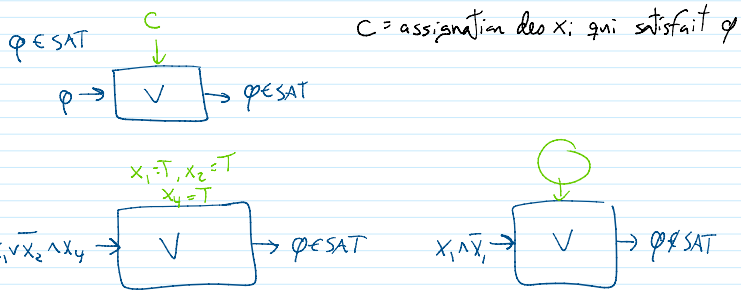
Déf. équivalente de NP par vérificateur

Soit L un langage. Une MT V est un vérificateur polynomial pour L si :

- $w \in L \iff \exists c \in \Sigma^*$ tel que V accepte $\langle w, c \rangle$ et $|c| \in O(|w|^k)$, $k \in \mathbb{N}$
- V s'exécute en temps polynomial



ex : vérif. pour SAT



Intuition : c est une "preuve" que le mot est dans le langage.
Si le mot n'est pas dans le lang., toute preuve échouera.

Vérificateur pour SAT

sur entrée $\langle \varphi, c \rangle$
vérifier que c encode une assignation des x_i
(si non, rejeter)
si c satisfait φ
accepter
sinon
rejeter

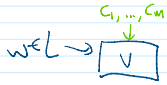
Thm : $L \in NP \iff \exists$ un vérificateur polynomial pour L

($\Rightarrow$) $L \in NP \Rightarrow \exists$ vérif

Soit $L \in NP$. Alors, $\exists$ MT non-dét. M dont langage est L et qui prend un temps $O(n^k)$, $k \in O(1)$.
On utilise M pour construire un vérif. V.

[Soit $w \in L$. Soit $c_1, \dots, c_m$ la séquence de configs de M menant à accepter w.

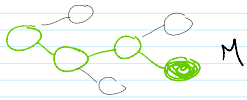
Le vérif. V attend $(c_1, \dots, c_m)$ comme certificat que $w \in L$.



V : sur entrée $\langle w, (c_1, \dots, c_m) \rangle$

Soit M la MT non-dét. dont le langage est L
Vérifier que c_1 est la cfg initiale de M sur w
pour $i = 1, \dots, m-1$
| si c_i ne peut pas mener à c_{i+1} sur M
| | rejeter

Vérifier que C_1 est la cfg initiale de M sur w
pour $i = 1,..,m-1$
  si C_i ne peut pas mener à C_{i+1} sur M
    rejeter
si C_m est acceptant
  accepter
sinon
  rejeter

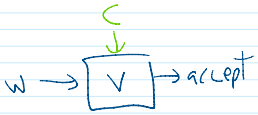


Si $w \in L$, V acceptera w si on lui passe $C_1, \dots, C_m$ empruntées par M, puisque M accepte w à C_m.

Si $w \notin L$, V rejettera w car M n'a aucune séq. de transitions acceptant w.

ex: V est en temps polynomial

($\Leftarrow$) $\exists$ vérif V pour $L \Rightarrow L \in NP$

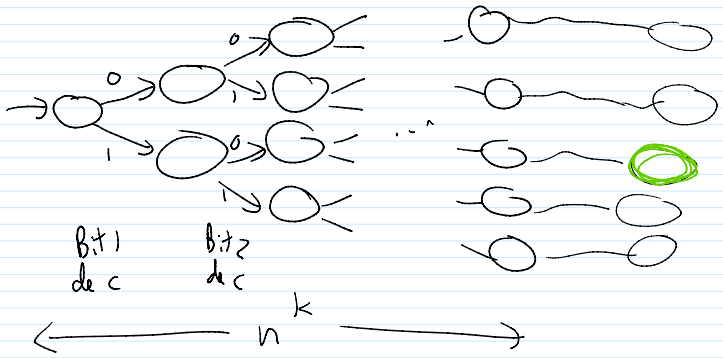


Soit V un vérif. poly pour L.
($w \in L \Leftrightarrow \exists c$, $\langle w, c \rangle$ est accepté par V)

On construit une MT non-dét pour L.
Soit M qui, sur entrée w:

- "deviner" un certificat c de façon de façon non-dét.
- simule V sur $\langle w, c \rangle$
- accepte ssi V a accepté

Ce M prend un temps poly non-dét. car tout c est de taille $O(n^k)$, $k \in O(1)$. Simuler V se fait aussi en temps poly.



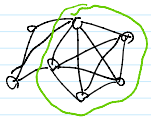
Si $w \in L$, M acceptera car elle trouvera le bon c pour que V accepte $\langle w, c \rangle$.

Si $w \notin L$, M n'acceptera pas car V rejette tout $\langle w, c \rangle$. ∎

Conséquence: SAT $\in$ NP. On peut utiliser une assignation des x_i qui satisfait un φ donné comme certificat et il est facile de vérifier qu'une telle assignation satisfait φ.

CLIQUE $\in$ NP

CLIQUE = $\{\langle G, k\rangle : G$ est un graphe contient une clique de taille $k\}$



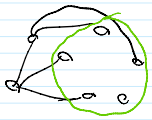
contient clique de taille $k = 5$

C est une clique si: $C \subseteq V(G)$ et $\forall u, v \in V(G)$ avec $u \neq v$, $uv \in E(G)$

CLIQUE $\in$ NP car on peut utiliser une clique $C \subseteq V(G)$ comme certificat, et on peut facilement vérifier que C est une clique de taille k.

IND-SET = $\{\langle G, k\rangle : G$ est un graphe contient un ensemble indépendant de taille $k\}$

$C \subseteq V(G)$ est un ens. indép. si: $\forall u, v \in V(G)$ avec $u \neq v$, $uv \notin E(G)$



$\in$ NP, certificat = un ens. indép.

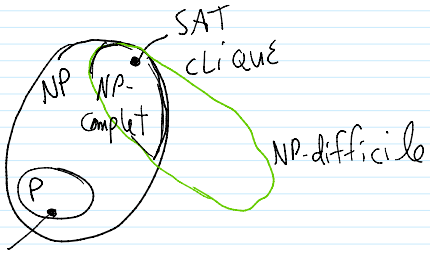
P = pbs résolubles en temps polynomial

NP = pbs <u>vérifiables</u> en temps polynomial

P = NP ??

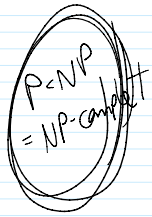
$P \subseteq NP$ $\quad$ $NP \subseteq P$??

Ce qu'on pense:



PATH
$\{\langle a,b,c\rangle : a+b=c\}$

Personne n'a exclut ceci:



<u>Thm</u>: si $L \in NP$, alors $L \in DTIME(2^{n^k})$, $k \in O(1)$