P vs NP

8 février 2023 08:21

P = langages décidables "rapidement"

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

$P \overset{?}{=} NP$

- Temps d'une MT : # de transitions faites avant l'arrêt

- Une MT est en temps $O(f(n))$ si
  $\forall$ entrée w, la MT prend un temps $O(f(n))$
  où $n = |w|$

- Espace d'une MT : # de cellules distinctes utilisées

Espace $\leq$ Temps

- $DTIME(f(n))$ = un langage L est dans $DTIME(f(n))$ s'il $\exists$ une MT M qui décide L en temps $O(f(n))$.

- Classe de langages P = décidables en temps polynomial

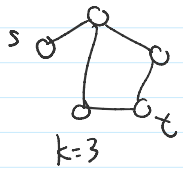
$$P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$$

P les décidables en temps "raisonnable"

$n \quad n^2 \quad n \log n \quad n^3 \quad n^{1000}$

- 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 de longueur $\leq k\}$
  $\in P$



ex : Dijkstra, $O(n \log n)$; fouille en largeur $O(n)$

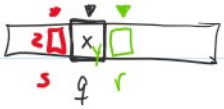
k=3 ex: Dikjstra, fouille en largeur
$O(n \log n)$ $O(n)$

MT non-déterministe (NP: non-déterministe polynomial)

Q : états
Σ : alphabet
$\in Q$: état initial
$\sqcup$: vide
$A \subseteq Q$: états acceptants

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

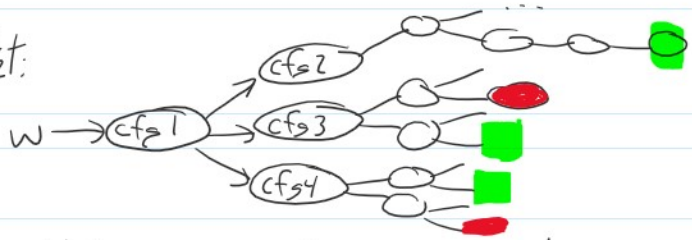
état-symbole état-symbole-gauche/droite



$(q, x), (r, y, R)$
$(q, x), (s, z, L)$
$(q, x), \ldots$

Intuition: MT $w \to cfg1 \to cfg2 \to \ldots \to fin$

MT-non-dét:



Config: string état - position - contenu de la bande
$q\,|0.\,|1|0|1|0|1|1$

- Acceptation

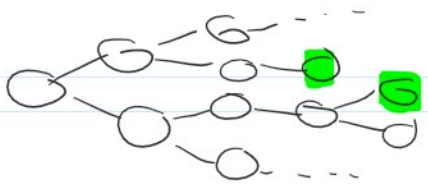
  Une MT non-det <u>accepte</u> un mot $w \in \Sigma^*$ s'il <u>existe</u> une séquence de choix de transitions qui termine dans un état acceptant sur entrée w.

- Temps d'une MT non-dét

  Sur entrée w, le temps d'une MT M est:

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

• sinon, le temps de M est 1



$NTIME(f(n))$ = un langage L est dans $NTIME(f(n))$ s'il $\exists$ une MT non-dét dont le langage accepté est L et qui accepte tous ses mots en temps $O(f(n))$

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

---

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

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

$x_1 = V$
$x_2 : f$
$x_3 = f$
$x_4 = f$

$(V \vee f \vee V) \wedge (V \vee V \vee f) \wedge (f \vee V)$

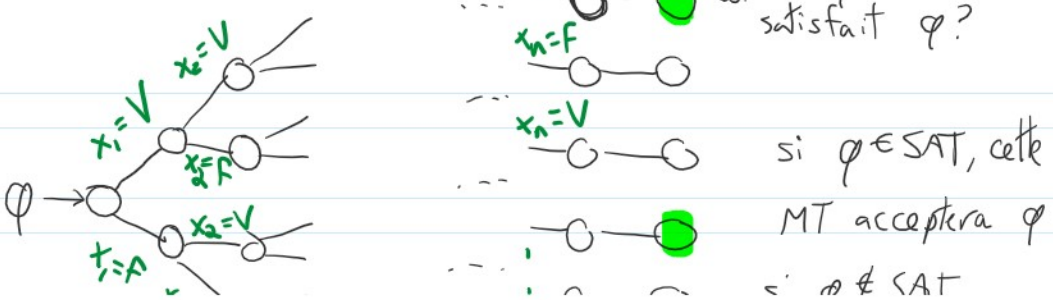
$V \quad \wedge \quad V \quad \wedge \quad V$

V

$\varphi \in SAT$

MT non-dét temps $O(n^k)$

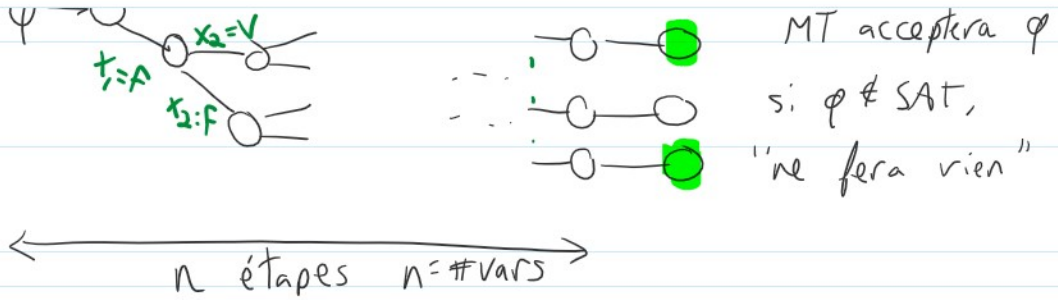
sur entrée φ, essayer toutes les assignations de façon non-dét



est-ce que l'assign. satisfait φ ?

si $\varphi \in SAT$, cette MT acceptera φ

si $\varphi \notin SAT$



Arborescence a 2^n noeuds, profondeur $O(n^k)$

Déf. équivalente de NP par vérificateur

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

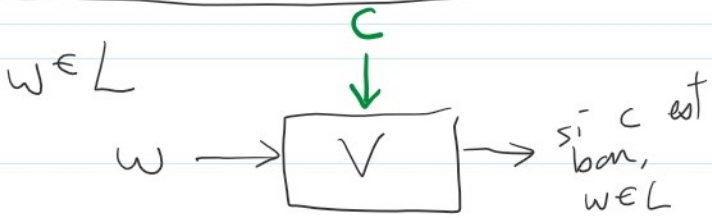
- $w \in L \iff \exists c \in \Sigma^*$ tel que V accepte $\langle w, c \rangle$ en temps $O(|w|^k)$, $k \in \mathbb{N}$

c = certificat

ex: vérificateur pour SAT

V attend en entrée $\langle \varphi, c \rangle$

- V vérifie que c correspond à une assign. des x_i (sinon rejette)
- V vérifie que c satisfait φ
- si oui, accepte, sinon rejette



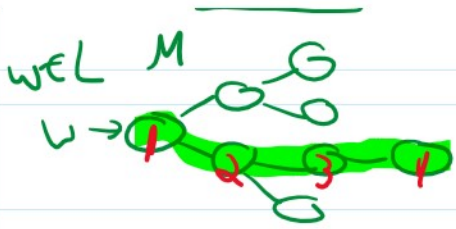
Théorème: $L \in NP \iff \exists$ un vérificateur polynomial pour L

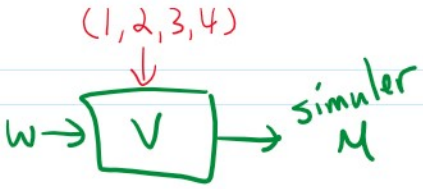
($\Rightarrow$) $L \in NP \Rightarrow \exists$ verif. V poly pour L

$w \in L$ M

(1,2,3,4)

simuler





Soit $L \in NP$. Alors $\exists$ MT non-dét. M dont le langage accepté est L en temps $O(n^k)$, $k \in \mathbb{N}$.
On utilise ce M pour construire V.
Ce V attend en entrée $\langle w, c \rangle$ où
$c = (c_1, c_2, \dots, c_\ell)$ est une séquence de cfg.

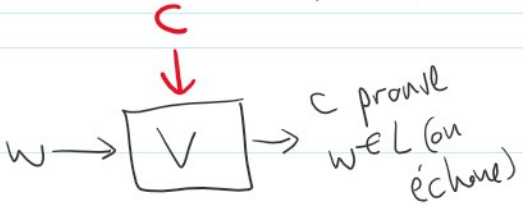
V: Sur entrée $\langle w, (c_1, c_2, \dots, c_\ell) \rangle$
Soit M la MT non-dét. pour L.
Vérifier que c_1 est cfg initiale de M (sinon rejet)
pour $i = 1 .. \ell - 1$
  | vérifier que c_i peut mener à c_{i+1} dans M (sinon rejet)
Vérifier que c_ℓ est acceptant (sinon rejet)
Accepte

si $w \in L$, $\exists c_1, \dots, c_\ell$ où M accepte $\Rightarrow$ V accepte
si $w \notin L$, $\nexists c_1, \dots, c_\ell$ où M accepte $\Rightarrow$ V rejette

exercice: V roule en temps polynomial (car $\ell \in O(n^k)$)

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

Soit V un vérif. poly pour L. Soit k tel que V est en temps $O(n^k)$



$w \in L \Leftrightarrow \exists c$ tel que V accepte $\langle w, c \rangle$

On construit une MT non-dét. M pour L en temps $O(n^k)$.

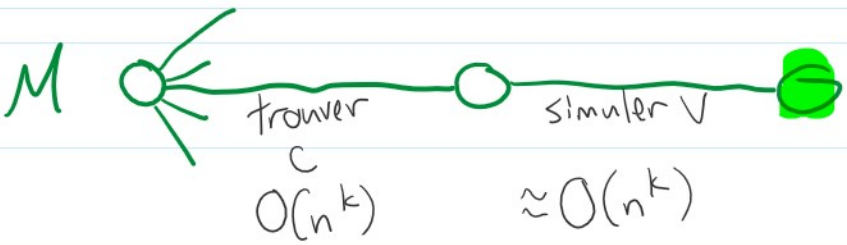
M : sur entrée w
M "devine" un certificat c de taille $O(n^k)$ | devine, tout tester par non-dét.
Simule V sur $\langle w, c \rangle$
Accepte ssi V accepte

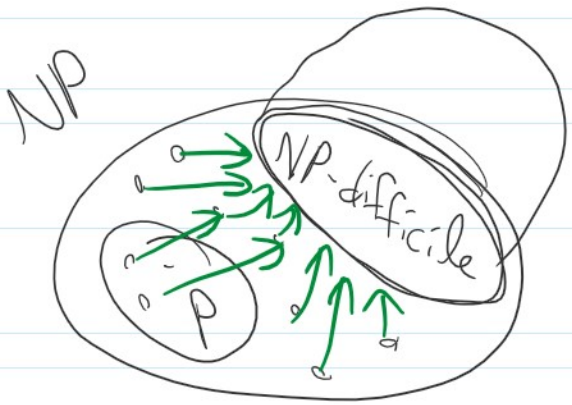
si $w \in L$, $\exists c$ tel que V accepte $\langle w, c \rangle$
$\Rightarrow$ cette MT M trouvera ce c et acceptera sur au moins un chemin

si $w \notin L$, $\nexists c$ tel que V accepte $\langle w, c \rangle$
$\Rightarrow$ cette MT M ne trouvera jamais de c et n'acceptera sur aucun chemin

Donc, le langage de M est L.

Pour dire que $L \in NP$, il faut que M soit en temps $O(n^k)$ (sur son chemin le + court quand $w \in L$)
La raison est que V est en temps $O(n^k)$ avec un $|c| \in O(n^k)$
$\Rightarrow$ M peut simuler V en temps polynomial.



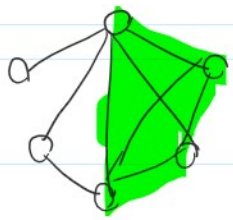


ex: $SAT \in NP$

Pour une formule φ, on peut utiliser une assign. A qui satisfait φ comme certificat.
Il est facile de vérifier en temps poly si A satisfait φ.

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

clique : $X \subseteq V(G)$ tel que $\forall u, v \in X$ distincts, $uv \in E(G)$

G  $k = 4$

$CLIQUE \in NP$.

On peut utiliser comme certificat un $X \subseteq V(G)$ avec $|X| = k$ qui forme une clique.
On peut vérifier que X est une clique en temps $O(n^2)$.