

## P vs NP

11 février 2021 10:17

Temps d'une MT = nb de transitions effectuées avant l'arrêt

Une MT M est  $O(f(n))$  si l'entrée I de taille n, (temps) M prend un temps  $O(f(n))$  sur entrée I.

Espace d'une MT M : # de cellules différentes accédées pendant l'exécution. (espace)

Espace  $\leq$  Temps

$\text{DTIME}(f(n))$  : un langage L est dans  $\text{DTIME}(f(n))$  si il  $\exists$  une MT en temps  $O(f(n))$  qui décide L.

Classe de langage P (polynomial)

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

P = les langages décidables en temps polynomial  
= les problèmes résolubles en temps "raisonnable"  
( $n^{10000}$  est raisonnable)

Langages dans P :

- $L_+ = \{ \langle a, b, c \rangle : a + b = c \} \in \text{DTIME}(n) \subseteq P$

$n = \# \text{ de bit pour représenter } a, b, c$

- $\text{PATH} = \{ \langle G, s, t, k \rangle : G \text{ est un graphe dans lequel la distance des sommets } s \text{ à } t \text{ est } \leq k \} \subseteq P \quad (\in \text{DTIME}(|V(G)| + |E(G)|))$

ex: file en largeur

- $\text{PGCD} = \{ \langle n, m, p \rangle : p \text{ est le plus grand diviseur commun à } n \text{ et } m \} \subseteq P \quad (\text{algo. Euclide})$

## MT non-déterministe

$Q$  = états

$\Sigma$  = alphabet

$\sqcup \in Q$  = état initial

(blank)  $\sqcup \in \Sigma$  = vide

$A \subseteq Q$  = états finaux d'acceptation

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

déterministe

$$\delta: Q \setminus A \times \Sigma \rightarrow Q \times \Sigma \times \{L, R\}$$

- $A \in Q$  : états finaux d'acceptation
  - $\delta \subseteq (Q \setminus A \times \Sigma) \times (Q \times \Sigma \times \Sigma^L, R)$
- (transitions)    état-symb. LR 1  
état-symb.  $\xrightarrow{a}$  état-symb. LR 2  
 $\xrightarrow{a}$  état-symb. LR 3

$$\delta: Q \setminus A \times \Sigma \rightarrow Q \times \Sigma \times \Sigma^L, R$$

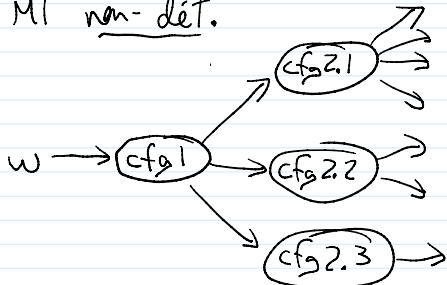
(fonction)

ex:  $(q_1, a)$  mène à  $(r, b, L)$   
 $(q_1, a)$  mène à  $(s, b, R)$   
 $(q_1, a)$  mène à  $(r, c, L)$

Intuition: MT déterministe

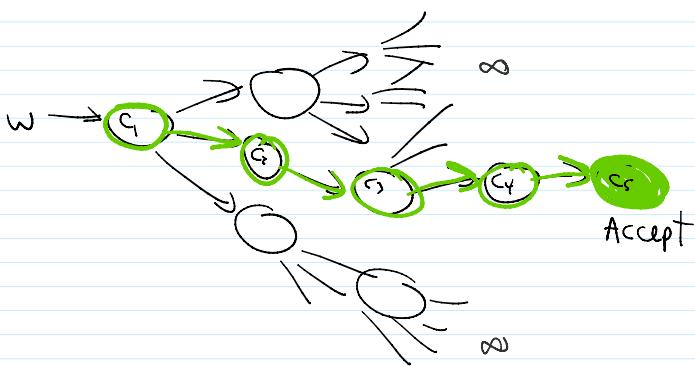


MT non-déterm.



## Acceptation

On dit qu'un MT non-déterm. accepte  $w \in \Sigma^*$  s'il existe une séquence de choix de transitions qui mènent à un état acceptant.  
Sinon, on dit la MT rejette  $w$ .



- Temps d'une MT non-déterm sur entrée  $w$ :

- si  $w$  est accepté, le temps est le # de transitions du plus court chemin menant à l'acceptation de  $w$ .
- si  $w$  n'est pas accepté, le temps est 1

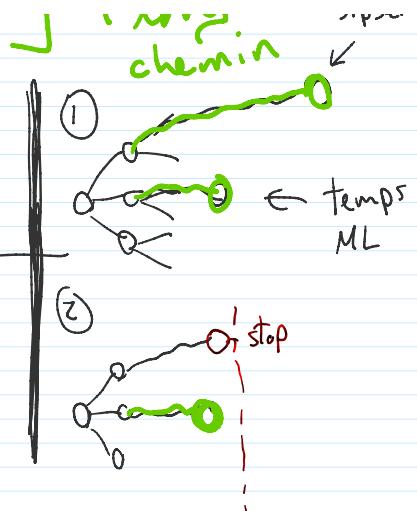
dans Sipser  
temps :  
+ long chemin  
temps Sipser

• si  $w$  n'est pas accepté, le temps est  $\infty$

Une MT non-déterm. est  $O(f(n))$  si son temps est  $O(f(n))$  l'entrée  $w \in \Sigma^*$  de taille  $n$ .

NTIME( $f(n)$ ): un langage  $L$  est dans NTIME( $f(n)$ ) si il  $\exists$  un MT non-déterm. en temps  $O(f(n))$  dont les mots acceptés sont précisément  $L$ .

$L$  langage d'une MT non-déterm = mot accepté par la MT non-déterm.



### Classe NP

(non-déterministe polynomial)  
~~(non polynomial)~~

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

Les langages acceptés par une MT non-déterm en temps "raisonnable"

### Déf. équivalente de NP par vérificateur

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

- $w \in L \iff \exists c \in \Sigma^*$  tel que  $V$  accepte  $\langle w, c \rangle$
- $V$  s'exécute en temps polynomial sur toute entrée.

Intuition: pour  $w \in L$ ,  $c$  est une "preuve" que  $w \in L$ .

$V$  s'assure de vérifier que la "preuve" est vraie.

On appelle  $c$  un certificat.

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

satisfaisable =  $\exists$  assignation true ou false aux variables tq. la formule évalue à True.

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

certificat:  $c = [x_1 = T, x_2 = F, x_3 = F, x_4 = F]$

$\vee = OU$   
 $\wedge = ET$   
 $\neg = neg.$

Vérificateur  $V$ : assigner les vars selon  $c$ , calculer le résultat

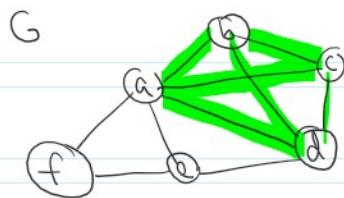
$$(T \vee F \vee \bar{F}) \wedge (\bar{F} \vee F \vee \bar{T}) \wedge (\bar{T} \vee \bar{F})$$

$$(T \vee F \vee T) \wedge (T \vee F \vee F) \wedge (F \vee T)$$

$$\begin{array}{c}
 \hookrightarrow (\top \vee F \vee \bar{F}) \wedge (\bar{F} \vee F \vee \bar{\top}) \wedge (\bar{\top} \vee \bar{F}) \\
 (\top \vee F \vee T) \wedge (\top \vee F \vee F) \wedge (F \vee T) \\
 T \quad \wedge \quad T \quad \wedge \quad T \\
 T \Rightarrow \text{satisfaisable } \in \text{SAT}
 \end{array}$$

ex: CLIQUE = { $\langle G, k \rangle$ :  $G$  est un graphe contenant une clique de  $k$  sommets}

- $C$  est une clique de  $G$  si •  $C \subseteq V(G)$  ( $V(G)$  = sommets)  
•  $\forall u, v \in C$  avec  $u \neq v$ ,  $uv \in E(G)$  ( $E(G)$  = arêtes)



{a, b, c, d} est clique de  $G$

{a, b, e, d} n'est pas une clique (manque be)

Certificat:  $c = \text{clique de taille } k$

$V$ : sur entrée  $\langle \langle G, k \rangle, c \rangle$

si  $|c| < k$ , rejeter

sinon si  $\forall u, v \in c$  tel que  $u \neq v$ , on a  $uv \in E(G)$ , accepter

sinon rejeter

Théorème:  $L \in NP \iff$   
il existe un vérificateur polynomial pour  $L$ .

$\Rightarrow (L \in NP \Rightarrow \exists \text{ vérif. } V)$

Soit  $L \in NP$ . Alors  $\exists$  MT non-déterm.  $M$  dont le langage est  $L$ .

On va utiliser  $M$  pour construire un vérificateur  $V$ .

Soit  $w \in L$ . Soit  $c_1, c_2, \dots, c_m$  la + courte séquence de configuration de  $M$  menant  $M$  à accepter  $w$ .

On "crée" un vérificateur  $V$  qui attend  $(c_1, c_2, \dots, c_m)$  comme certificat.

Sur entrée  $\langle w, (c_1, c_2, \dots, c_m) \rangle$ ,  $V$  (connaît  $M$ )

- simulate  $M$  sur entrée  $w$  pendant  $m$  étapes
- vérifie que  $\forall i \in \{1, 2, \dots, m-1\}$ ,  $c_i$  peut bel et bien menon:  $c_{i+1}$  selon les spécifications de  $M$ .

- vérifie que  $H_i \in \{1, 2, \dots, m-1\}$ ,  $c_i$  peut bel et bien mener à  $c_{i+1}$  selon les spécifications de  $M$ .
- sinon, rejeter
- si oui, accepter ssi  $c_m$  est une config acceptante.

Il faut que  $V$  accepte  $\langle w, (c_1, \dots, c_m) \rangle \Leftrightarrow w \in L$ .

S:  $w \in L$ ,  $V$  acceptera car  $M$  accepte  $w$  sur  $c_1, c_2, \dots, c_m$ .

S:  $w \notin L$ ,  $V$  n'accepte pas car  $M$  n'accepte jamais  $w$ .

$\Rightarrow V$  est un vérificateur pour  $L$ .

exercice: est-ce que  $V$  est un vérificateur en temps polynomial ?

danger: si une  $C_i$  nécessite  $2^n$  bits, problème  
à argumenter:  $|C_i|$  est  $n^k$

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

Soit  $V$  un vérif. poly tel que

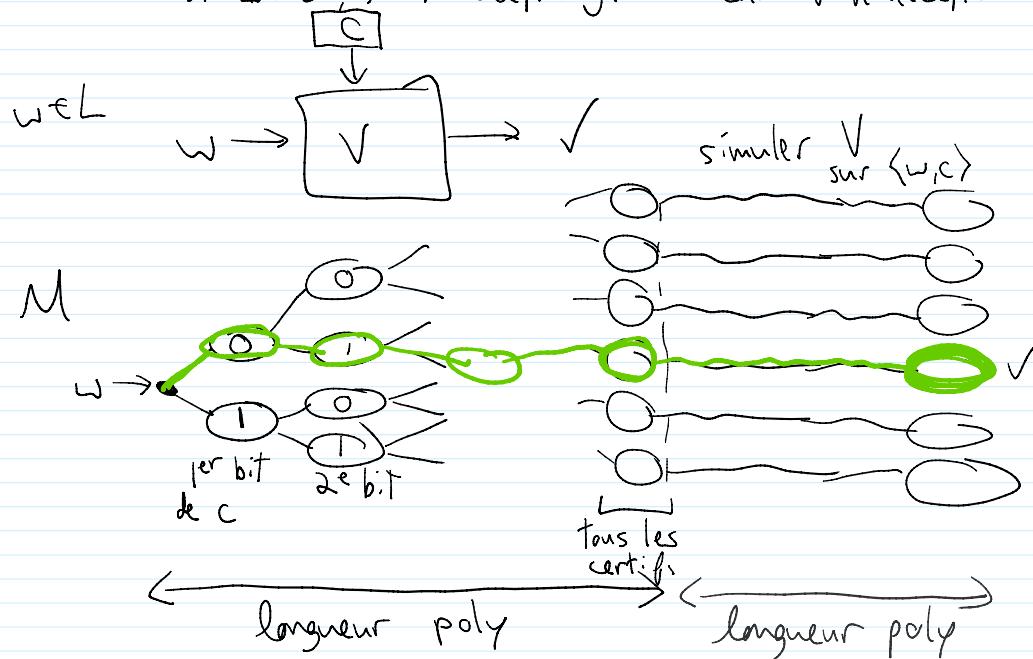
$w \in L \Leftrightarrow \exists c \text{ t.q. } V \text{ accepte } \langle w, c \rangle$ .

On va construire une MT non-déterm. dont le langage est  $L$ .

Soit  $M$  qui, sur entrée  $w$ :

- "choisir" un certificat  $c$  de façon non-déterministe (les essaie tous)
- simule  $V$  sur  $\langle w, c \rangle$
- $M$  accepte ssi  $V$  accepte

$\rightarrow$  si  $w \in L$ , un des chemins de  $M$  acceptera  $w$  car  $V$  accepte  $\langle w, c \rangle$   
si  $w \notin L$ ,  $M$  n'accepte jamais car  $V$  n'accepte aucune  $\langle w, c \rangle$  ■

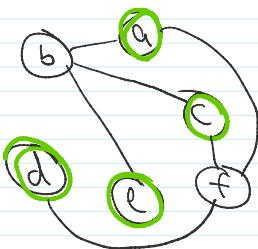


Consequences : SAT  $\in$  NP (vérif. par assignation)  
 CLIQUE  $\in$  NP (vérif. qui reçoit la clique)

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

$I$  est un ensemble indép. si :

- $I \subseteq V(G)$
- $\forall u, v \in I, uv \notin E(G)$



$\{a, c, d, e\}$  est un ens. indép.

IND-SET  $\in$  NP. Certificat :  $I \subseteq V(G)$  formant un ens. indép.  
 Vérificateur :  $|I| \geq k$   
 $\forall u, v \in I, uv \notin E(G)$

P = problèmes résolubles rapidement  $O(n^k)$

NP = problèmes vérifiables rapidement  $O(n^k)$  vérif déterministe

Question : est-ce que  $P = NP$  ?

•  $P \subseteq NP$  : parce que ce qu'on peut faire avec une MT dét., on peut le faire avec une MT non-dét..

•  $NP \subseteq P ???$  / M\$ USD

Importance : beaucoup de pb dans NP qu'on voudrait résoudre en temps poly

$\notin$  NP

- route optimale de livraison (camion voyageur)
- assembler des séquences d'ADN
- planifier des horaires de vol selon des contraintes
- résoudre un tableau Sudokus
- ... (+6000 autres)

Théorème: soit  $L \in NP$ . Alors il existe une MT déterministe en  $O(2^{n^k})$  pour une constante  $k \in \mathbb{N}$  qui décide  $L$ .

Preuve: puisque  $L \in NP$ ,  $\exists$  MT non-déterm.  $M$  qui accepte  $L$  en temps  $O(n^k)$ .

Pour décider  $L$  de façon déterm., on va utiliser l'algorithme:  $L \leq_{c.n^k} L$

sur entrée  $w$ : appeler  $\text{simuler}_M(w, \text{cfgInit de } M, 0)$

$\text{simuler}_M(w, \text{cfg}, \text{nbTrans})$

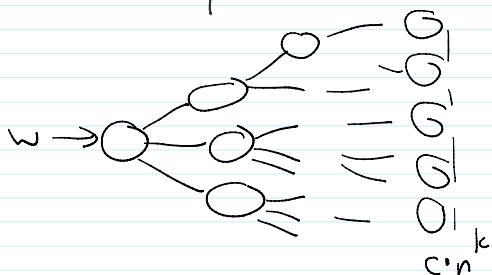
si  $\text{nbTrans} > c \cdot n^k$

| rejeter

pour chaque config  $c$  possible sur  $M$  à partir de  $\text{cfg}$

| si  $c$  acceptante, accepter

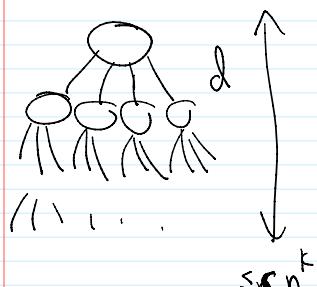
| sinon  $\text{simuler}_M(w, c, \text{nbTrans} + 1)$



Pour une config  $\text{cfg}$ , le nb de config  $c$  possibles est borné par

$$|Q| \cdot |\Sigma| \cdot |\{L, R\}|$$

$\underbrace{\quad \quad \quad}_{\text{constante } d}$



Donc, on explore un arbre de récursion où chaque noeud a  $\leq d$  enfants, et la profondeur est  $\leq c \cdot n^k$

$\Rightarrow$  le temps est borné par  $O(d^{c \cdot n^k})$

$$\Rightarrow O(d^{c \cdot n^k}) = O(2^{\log d \cdot c \cdot n^k})$$

$$= O(2^{n^{k'}}) \text{ où } k' \in \mathbb{N}. \blacksquare$$

co-NP

Un langage  $L$  est dans co-NP si son complément est dans NP.

$$L \in \text{co-NP} \iff \bar{L} \in \text{NP} \quad (\text{où } \bar{L} = \{w : w \notin L\})$$

ex: UNSAT =  $\{\varphi : \varphi \text{ n'est pas satisfaisable}\}$

$$\overline{\text{UNSAT}} = \text{SAT} \in \text{NP} \Rightarrow \text{UNSAT} \in \text{co-NP}$$

Est-ce que co-NP ⊂ NP ?

Est-ce que UNSAT ∈ NP ?

Personne ne sait.

Difficulté: certificat pour vérifier qu'une formule  $\varphi$  n'est pas satisfaisable?

Certificat possible: Toutes les assignations possibles.

problème: pas vérifiable en temps poly.