Les classes de complexité servent à classifier des problèmes selon leur difficulté.

I Généralités

1) Complexité

Rappel 0 : On note (q,α,β) les configurations d'une machine de Turing : q est l'état courant, α le mot strictement à gauche de la tête de lecture et β celui à droite.

def 1 : Soit M une MT, w un mot et γ un calcul de M sur w : $\gamma_0 = (q_0, \varepsilon, w) \vdash \dots \vdash \gamma_k = (q_k, \alpha_k, \frac{\beta}{2}) \vdash^* \gamma_m$

La complexité spatiale de M est :

$$s_M(n) = \max_{|w|=n} \max_{\gamma, \gamma_0 = (q_0,\varepsilon,w)} \max(|\alpha_i|+|\beta_i|)$$

Le temps de γ est $t_M(\gamma) = m$ et la complexité temporelle de M est $t_M(n) = \max_{|w|=n} \max_{\gamma, \gamma_0=(q_0,\varepsilon,w)} t_M(\gamma)$

prop 2 : Soit M une MT. $\exists K>0$ tel que $t_M(n) \leq 2^{K s_M(n)}$ et $s_M(n) \leq \max(n, t_M(n))$

thm 3 : Soit $k \in \mathbb{N}^*$, M une MT telle que $t_M(n) = \Omega(n)$. Alors $\exists M' \sim M$ telle que $t_{M'}(n) \leq t_M(n)/k$

thm 4 : Si M est une MT à ruban bi-infini $\exists M' \sim M$ usuelle telle que $t_{M'}(n) = \Theta(t_M(n))$

thm 5 : Si M est une MT à k bandes, $\exists M' \sim M$ usuelle telle que $t_{M'}(n) = \mathcal{O}(t_M(n)^2)$

thm 6 : Soit M une MT non déterministe. $\exists M' \sim M$ déterministe telle que $t_{M'}(n) = 2^{\mathcal{O}(t_M(n))}$

thm 7 : Si $s(n) \geq \log n$. $\forall M$ non déterministe en espace $s(n)$, $\exists K>0$ et M' non déterministe d'espace $Ks(n)$ acceptant $L(M)^c$

thm 8 (Savitch) : Si $s(n) = \Omega(n)$, $\forall M$ non déterministe de complexité spatiale $s(n)$, $\exists M' \sim M$ déterministe en espace $\mathcal{O}(s^2(n))$ (DEV)

2) Classes de complexité et réduction.

def 9 :
- $\mathrm{Time}(f(n))$ est l'ensemble des problèmes décidés par M T déterministe en temps $\mathcal{O}(f(n))$
- $\mathrm{NTime}(f(n))$ est l'ensemble des problèmes décidés par MT en temps $\mathcal{O}(f(n))$
- $P = \bigcup_{k>0} \mathrm{Time}(n^k)$, $NP = \bigcup_{k>0} \mathrm{NTime}(n^k)$, $\mathrm{EXPTIME} = \bigcup_{k>0} \mathrm{Time}(2^{n^k})$, $\mathrm{NEXPTIME} = \bigcup_{k>0} \mathrm{NTime}(2^{n^k})$
- De même, on définit (N)Space$(f(n))$, (N)PSPACE et (N)EXPSPACE

prop 10 : Par Savitch, PSPACE = NPSPACE et EXPSPACE = NEXPSPACE.

prop 11 : Si $P = NP$, $\mathrm{EXPTIME} = \mathrm{NEXPTIME}$.

def 12 : Si C est une classe, co-C est l'ensemble des problèmes dont le complémentaire est dans C.

prop 13 : Pour les classes déterministes et les classes spatiales, $C = \text{co-}C$.

prop 14 : $P \subset NP \cap \text{co}NP$

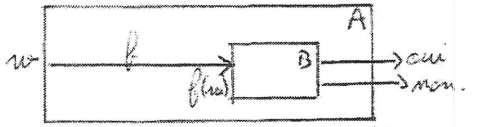
$NP \cup \text{co}NP \subset PSPACE \subset EXPTIME \subset NEXPTIME \cap \text{co}NEXPTIME$

$NEXPTIME \cup \text{co}NEXPTIME \subset EXPSPACE$

thm 15 : Si $g = o(f)$ et f calculable en temps $\mathcal{O}(f(n))$, $\mathrm{Time}(g(n)) \subsetneq \mathrm{Time}(f(n))$. Si f est calculable en espace $\mathcal{O}(f(n))$, $\mathrm{Space}(g(n)) \subsetneq \mathrm{Space}(f(n))$

cor 16 : $P \subsetneq \mathrm{EXPTIME}$, $\mathrm{PSPACE} \subsetneq \mathrm{EXPSPACE}$.

def 17: On dit que A se réduit à B ssi $\exists f$ calculable telle que $\forall w$, w est solution de $A \Leftrightarrow f(w)$ est solution de B



def 18: Soit C une classe, B un problème. B est dit
- C-difficile ssi $\forall A \in C$, A se réduit à B via f vérifiant des propriétés spécifiques à C
- C-complet ssi $B \in C$ et B est C-difficile

II P et NP

1) Caractérisations

def 19: Un vérificateur pour un langage L est une MT déterministe M qui accepte des entrées $\langle w, c \rangle$ telle que $L = \{w, \exists c \ \langle w, c \rangle \in L(M)\}$.

prop 20: $L \in NP \Leftrightarrow \exists$ un vérificateur en temps polynomial en $|w|$ pour L.

def 21: On parle de P-difficulté avec des réductions en espace logarithmique
- On parle de NP-difficulté avec des réductions en temps polynomial

prop 22: L est NP-complet $\Leftrightarrow \overline{L}$ est co-NP complet

2) Problèmes P et P-complets

- Savoir si un entier est premier est P
- Si G est une grammaire algébrique sous forme de Chomsky, savoir si $w \in L(G)$ est P.
- Etant donné un circuit logique, le problème de savoir si sa valeur de sortie est 0 ou 1 est P-complet

3) Problèmes NP-complets en logique

thm (Cook) 23: Le problème consistant à déterminer si une formule du calcul propositionnel est satisfiable (SAT) est NP-complet

cor 24: Si on se restreint à des formules sous FNC où chaque clause a moins de k littéraux (k-SAT) et si $k \geq 3$, k-SAT est NP-complet

rmq 25: 2-SAT et HORNSAT sont P.

cor 26: Savoir si une formule est une tautologie est co-NP-complet.

- Si C est un ensemble de clauses et $k < |C|$, le problème "$\exists$? valuation qui satisfait au moins k clauses" est NP-complet
- Etant données deux expressions régulières E et E' sans *, savoir si $L(E) = L(E')$ est co-NP-complet.

4) Problèmes NP sur les graphes.

- Etant donné un graphe G et $k \in \mathbb{N}$, "$\exists$? une clique de taille k dans G", "$\exists$? un stable de taille k dans G" et "$\exists$? une couverture de sommets de taille k" sont NP-complets.
- Le problème du chemin hamiltonien : "$\exists$? un cycle passant une et une seule fois par chaque sommet de G" est NP-complet
- Le problème du voyageur de commerce est NP-complet (même dans le cas euclidien).
- Déterminer le nombre chromatique d'un graphe est NP complet

5) Autres problèmes

- Si $S \subset \mathbb{N}$ est fini et $k \in \mathbb{N}$, "$\exists$? $S' \subset S$ de somme k" est NP-complet
- Le problème de la programmation linéaire en nombres entiers est NP-complet.

III L, NL, co-NL

On suppose que chaque MT a une bande d'entrée où elle ne peut pas écrire et une bande de sortie où elle ne peut aller qu'à droite. En ne comptant pas l'espace pris par ces deux bandes, on peut définir $L = Space(\log n)$ et $NL = NSpace(\log n)$.

prop 27: $L \subset NL \subset P$, $L \subset co\text{-}NL \subset P$.

def 28: On parle de NL-complétude en utilisant des réductions en espace logarithmique

ex 29:
- 2-SAT est NL-complet
- l'accessibilité dans un graphe orienté est NL-complet et co-NL-complet

cor 30: $NL = co\text{-}NL$

rmq 31: le théorème de Savitch est valable avec $s(n) \geq \log n$.

IV Au delà de NP

1) PSPACE

def 32: Pour la PSPACE-difficulté, on utilise des réductions en temps polynomial.

- M une MT déterministe, x un mot. Le problème "M accepte-t-elle x sans sortir des $|x|$ premiers caractères du ruban" est PSPACE complet
- La satisfiabilité d'une formule booléenne quantifiée est un problème PSPACE-complet.
- L'universalité d'un automate fini et l'équivalence d'expressions régulières sont PSPACE-complets.
- Le jeu de Go est PSPACE-difficile et devient PSPACE-complet si on borne le nombre de tours.

2) EXPTIME, NEXPTIME et co NEXPTIME

def 33:
- Les réductions EXPTIME sont en espace polynomial
- Les réductions NEXPTIME sont en temps exponentiel

prop 34: $L \in NEXPTIME \iff \exists$ un vérificateur de L en temps exponentiel

- La satisfiabilité de formules du premier ordre prénexes sans symboles de fonction et de la forme:
  - $\exists^* \forall \exists^*$ est EXPTIME-complet
  - $\exists^* \forall \forall \exists^*$ et $\exists^* \forall^*$ sont NEXPTIME-complets.
- Savoir si le carré $n \times n$ est pavable avec les tuiles d'un ensemble T est NEXPTIME-complet
- L'équivalence d'expressions régulières avec 2 et sans $*$ est co-NEXPTIME-complet

3) EXPSPACE

def 35: Les réductions se font en temps exponentiel

- L'équivalence d'expressions régulières avec 2 et $*$ est EXPSPACE-complet.