INFO-F408: Computability & complexity

Rémy Detobel

4 décembre, 2017

1 Chapitre 8 : SPACE Complexity

Pour une machine de Turing, fonction : $f : \mathbb{N} \to \mathbb{N}$.
Taille maximum d'une case scannée par une machine de Turing sur chaque input de taille n.

Machine de Turing non déterministe : $*$ + maximise le calcule de chaque branche

$$\begin{aligned} \mathsf{SPACE}(f(n)) = \{L|L \text{ un langage décidable par une machine} \\ \text{de turing déterministe en } O(f(n)) - space\} \end{aligned}$$

$$\begin{aligned} \mathsf{NSPACE}(f(n)) = \{L|L \text{ un langage décidable par une machine} \\ \text{de turing \textbf{non} déterministe en } O(f(n)) - space\} \end{aligned}$$

Exemple : $SAT \in SPACE(n)$

1.1 Théorème de Savitch

Théorème 8.5

Théorème

Pour toute fonction $f : \mathbb{N} \to \mathbb{N}$ telle que $\forall n \in \mathbb{N} : f(n) \geqslant n$:
$NSPACE(f(n)) \subseteq SPACE(f^2(n))$, où $f^2(n) = (f(n))^2$, à ne pas confondre avec $f(f(n))$.

Preuve du théorème de Savitch :
Machine de turing non déterministe N décidable pour A en $f(n)space$
Algorithme Canyield sur l'input : c_1, c_2 (configurations), t (nombre d'étapes).

Question Peut-on aller de la configuration c_1 à la configuration c_2 avec un maximum de t étapes.
Supposons sans perte de généralité, qu'il n'existe qu'une seule configuration acceptante (supprimer le contenu du tape et déplacer la tête sur la première case).

Pour une constante d :

$$CANYIELD(c_{start}, c_{accept}, 2^{df(n)}) \text{ simule N}$$

1. Si $t = 1$, est-ce que N peut atteindre c_2 depuis c_1 en une étape. (cfr "legal windows")
2. Dans les autres cas, pour chaque configuration c_m de N en utilisant $space\ f(n)$:
   (a) exécuter $CANYIELD(c_1, c_m, t/2)$
   (b) exécuter $CANYIELD(c_m, c_2, t/2)$
   (c) Accepter si les deux acceptent
3. Si pas accepté, alors on rejette

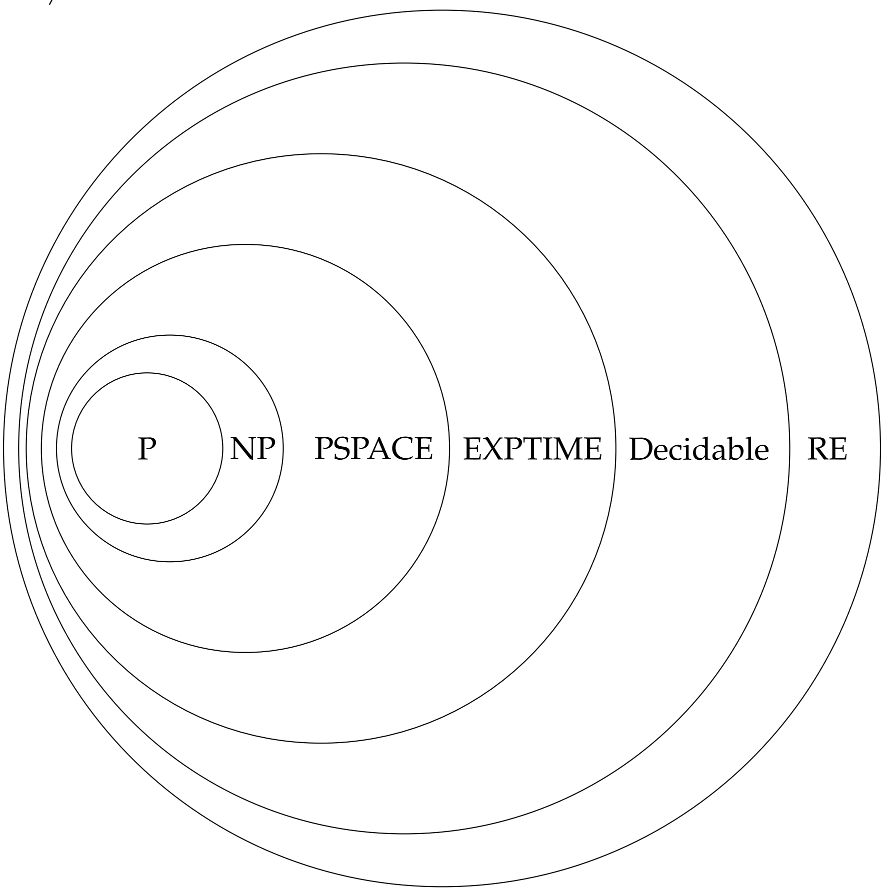
Remarque : t ne peut pas être impaire car il est égal à une puissance de 2 (à savoir $2^{df(n))}$)

Le nombre maximum d'appel récursif est égal à $\log_2 t$, donc $\log_2 2^{df(n)} = O(f(n))$
L'espace requis pour chaque appel récursif est plus au moins égal à $c_m \leqslant f(n))$
$\Rightarrow O(f^2(n))$

1.2 PSPACE

$$\text{PSPACE} = \cup_k \text{SPACE}(n^k)$$

On ne sait pas si NP est strictement inclut dans $PSPACE$ ou pas. Par contre on est sur que $P \neq EXPTime$



Cfr : Complexité de Zoo

$$\text{NPSPACE} = \cup_k \text{NSPACE}(n^k)$$

Par le théorème de Savitch, on peut écrire : $PSPACE = NPSPACE$.

Complémentarité de $PSPACE$:
Un langage B est PSPACE-complet si et seulement si :

1. $B \in PSPACE$
2. $\forall A \in PSPACE, A \leqslant_p B$ ("PSPACE-Hard")

NB : Une instance $\phi(x)$ (qui est une formule booléen) du problème SAT $\Leftrightarrow$ est-ce que cette fonction est valide : $\exists x : \phi(x)$?

TRUE QUANTIFIED BOOLEAN FORMULE (TQBF)

Exemple : $\exists x | \forall y : \exists z | \forall t : \phi(x, y, z, t)$

$TQBF = \{\langle\phi\rangle | \phi$ est vrai et complètement quantifié comme étant une formule booléen$\}$

$TQBF \in NP$, la réponse n'est pas définie clairement, mais on peut facilement montrer par un exemple qu'il est dans NP.

Théorème

TQBF est PSPACE Complet

1. $TQBF \in PSPACE$
2. TQBF est PSPACE-Dur ($\forall A \in PSPACE, A \leqslant_p TQBF$).
   Pour M décidable sur A dans n^k space, nous ajoutons un input w à une formule booléenne totalement quantifié (Fully quantified Boolean formula) qui est vrai si et seulement si w est accepté par M.
   Rappelons nous l'encodage de la configuration de "string" en variable booléens dans le théorème de Cook-Levin : $X_{i,s}$ = "Symbole S est trouvé à la position i"[a].

a. Chaque configuration peut être encodé avec $O(n^k)$ variables booléen

$\phi_{c_1,c_2,t} \Leftrightarrow$ M peut aller de la configuration c_1 à la configuration c_2 avec au maximum t étapes.

Pour $t = 1$: cfr "legal windows"

Échauffement : $\phi_{c_1,c_2,t} = \exists m_1 \left[\phi_{c_1,m_1,t/2} \wedge \phi_{m_1,c_2,t/2}\right]$ Faire ceci revient à séparer en deux le problème et donc cela devient exponentiel.

Meilleure idée : $\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3, c_4) \in \left\{(c_1, m_1), (m_1, c_2)\right\}$
$[\phi_{c_3,c_4,t/2}]$

NB : $\forall x \in \{y, z\}$ peut être remplacé par $\forall x \Big[(x = y \vee x = z) \rightarrow ...\Big]$

Observation :

- A chaque niveau de la récursion, on rajoute une partie de taille $O(f(n)) = O(n^k)$
- Le nombre de niveau est égal à $\log_2 t$

$$\phi_{c_{start},c_{accept},2^{(df(n)=n^k)}} \text{ a une taille de } O(f^2(k)) = O(n^{2k})$$