

INFO-F408: Computability & complexity

Rémy Detobel

4 décembre, 2017

1 Chapitre 8 : SPACE Complexity

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

Machine de turing non déterministe : * + maximise le calcul de chaque branche

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

$$\text{NSPACE}(f(n)) = \{L \mid L \text{ un langage décidable par une machine de turing } \mathbf{non} \text{ déterministe en } O(f(n)) - \text{space}\}$$

Exemple : $\text{SAT} \in \text{SPACE}(n)$

1.1 Théorème de Savitch

Théorème 8.5

Théorème

$$\forall f(n) \geq n \\ \text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$$

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 que sans perte de généralité, il n'existe qu'une seule configuration acceptante (supprimer le contenu du tape et déplacer la tête sur la première case).

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

1. Si $t = 1$, est-ce que N peut attendre 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 $\text{CANYIELD}(c_1, c_m, t/2)$
 - (b) exécuter $\text{CANYIELD}(c_m, c_2, t/2)$
 - (c) Accepter si les deux acceptent
3. Si pas accepté, alors on rejette

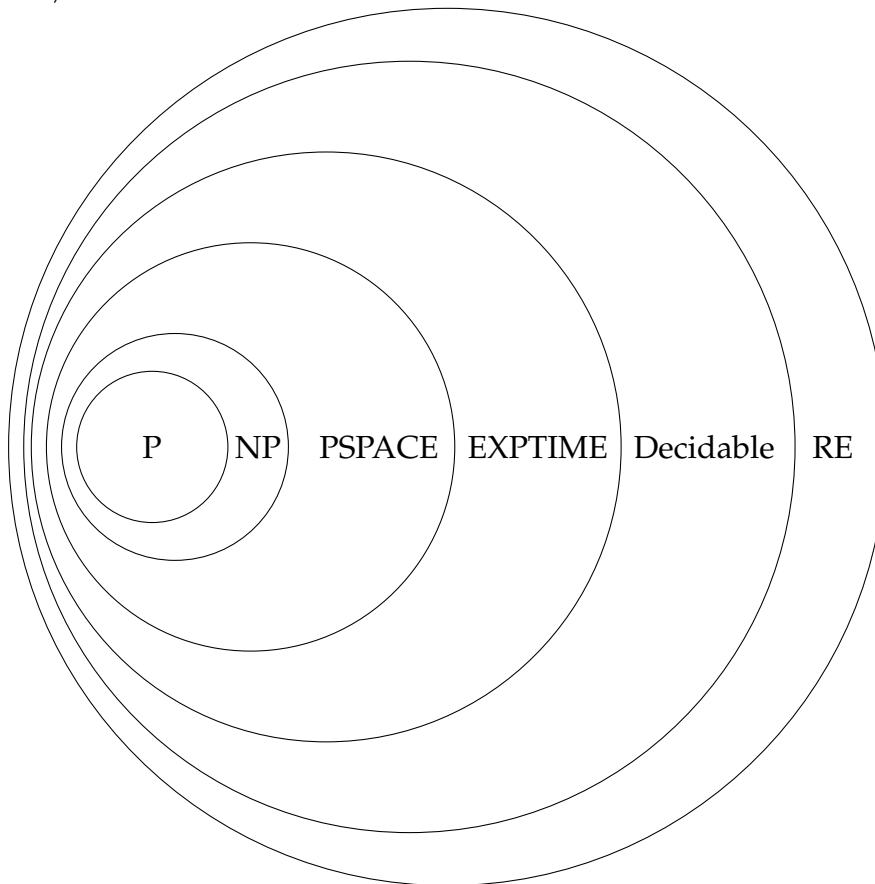
Remarque : t ne peut pas être impaire car il est égal à une puissance de 2 (à savoir $2^{df(n)}$)

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 $c_m \leq f(n)$
 $\Rightarrow O(f^2(n))$

1.2 PSPACE

$$PSPACE = \cup_k SPACE(n^k)$$

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



Zoo

Cfr : Complexité de

$$NPSPACE = \cup_k NSPACE(n^k)$$

En parant du théorème de Savitch, on peut écrire : $\Rightarrow PSPACE = NPSPACE$.

Complémentarité de PSPACE :

Un langage B est PSPACE complet si et seulement si :

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

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

TRUE QUANTIFIED BOOLEAN FORMULE (TQBF)

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

$$\text{TQBF} = \left\{ \langle \phi \rangle \mid \phi \text{ est vrai et complètement quantifié} \right. \\ \left. \text{comme étant une formule booléenne} \right\}$$

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

Théorème

TQBF est PSPACE Complet

1. TQBF \in PSPACE

2. TQBF est PSPACE-Dure ($\forall A \in \text{PSPACE}, A \leq_p \text{TQBF}$).

Pour M décidable sur A dans n^k space, nous ajoutons un input w a une formule booléenne totalement quantifiée (Fully quantified Boolean formula) qui si est vrai si et seulement si w est accepté par M.

Rappelons nous l'encodage de la configuration de "string" en variable booléennes 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 booléennes

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

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

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

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

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

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 $\log_2 t$

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