Complexité en espace

22 mars 2023 18:23

- Une MT M est en espace $O(f(n))$ si elle utilise tjrs au plus $c \cdot f(n)$ cellules de ruban, c = constante.

- Une MT non-dét. M est en espace $O(f(n))$ si $\forall$ séq. de transitions acceptante, l'espace est $O(f(n))$.

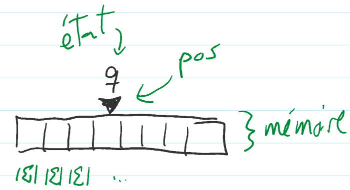
- Espace $\leq$ Temps

- Temps $\leq 2^{d \cdot \text{Espace}}$ d = constante (si M arrête)

Thm : si une MT M est en espace $O(f(n))$, alors M est en temps $O(2^{d\, f(n)})$ (si M arrête tjrs), où d = constante.

Preuve : si M n'a pas de boucle ∞, alors M ne rencontre jamais $2\times$ la même config.

Donc temps de $M \leq$ nb de cfg possibles   cfg



$$
\begin{aligned}
&= \#\text{ états possibles} \times \quad |Q| \\
&\quad \#\text{ pos possibles} \times \quad c \cdot f(n) \text{ (temps)} \\
&\quad \#\text{ bandes mém. possibles} \quad |\Sigma|^{c f(n)} \\
&= |Q| \times c \cdot f(n) \times |\Sigma|^{c f(n)} \\
&\leq 2^{a} \cdot 2^{c \cdot f(n)} \cdot 2^{b \cdot c f(n)} \qquad a, b, c \in O(1) \\
&= 2^{a + c f(n) + bc\, f(n)} \\
&= 2^{d\, f(n)} \qquad d \in O(1) \qquad \blacksquare
\end{aligned}
$$

$\text{DSPACE}(f(n)) = \{ L : \exists$ M.T. M qui décide L en espace $O(f(n)) \}$

$\text{NSPACE}(f(n)) = \{ L : \exists$ MT non-dét M dont le langage accepté est L et qui est en espace $O(f(n)) . \}$

$$\text{PSPACE} = \bigcup_{k=1}^{\infty} \text{DSPACE}(n^k)$$

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

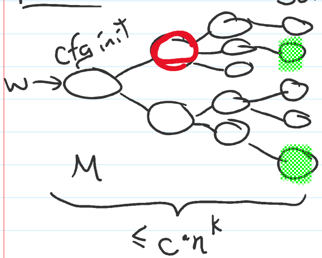
$$NPSPACE = \bigcup_{k=1}^{\infty} NSPACE(n^k)$$

Proposition : $P \subseteq PSPACE$

"Preuve": Espace $\leq$ Temps

Proposition: $NP \subseteq PSPACE$

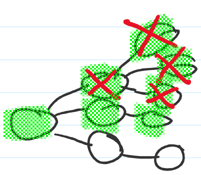
Preuve: Soit $L \in NP$. $\exists$ MT non-dét. M qui accepte $w \in L$ en temps $O(n^k)$. Donc $\exists$ constante c telle que ce temps est $\leq c \cdot n^k$ (k = constante).



Pour décider L en espace polynomial:

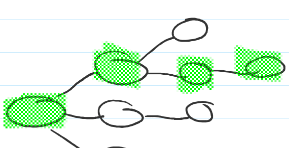
simulerM (w, cfgcur, nbsteps)

```
    si nbsteps > c·n^k        (*)
        return false
    si cfgcur est acceptante
        return true

    pour chaque cfg pouvant suivre cfgcur
    selon M
        si simulerM(w, cfg, nbsteps+1)
            return true
        fin si
    fin pour

    return false
```

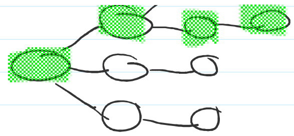


Appel init: simulerM(w, cfginit, 1)

- Un appel à simulerM prend un espace $O(n^k)$ (sans compter les récursions), car une config de M ne prendra jamais plus que $c \cdot n^k$ espace, et on n'a qu'à stocker 2 configs: cfgcur et cfg.
- Au total, l'espace requis par l'exécution sera proportionnel à la profondeur de l'arbre de récursion, fois l'espace requis par appel. La profondeur est $\leq c \cdot n^k$ (par $*$) et l'espace par appel est $\leq c \cdot n^k$.



$\Rightarrow$ Espace total: $\leq$ (profondeur) $\times$ (espace appel)



$\Rightarrow$ Espace total: $\leq$ (profondeur) $\times$ (espace appel)

$\leq \quad c \cdot n^k \times c \cdot n^k$

$= \quad O(n^{2k})$ ∎

Preuve alt:

Thm: SAT $\in$ PSPACE

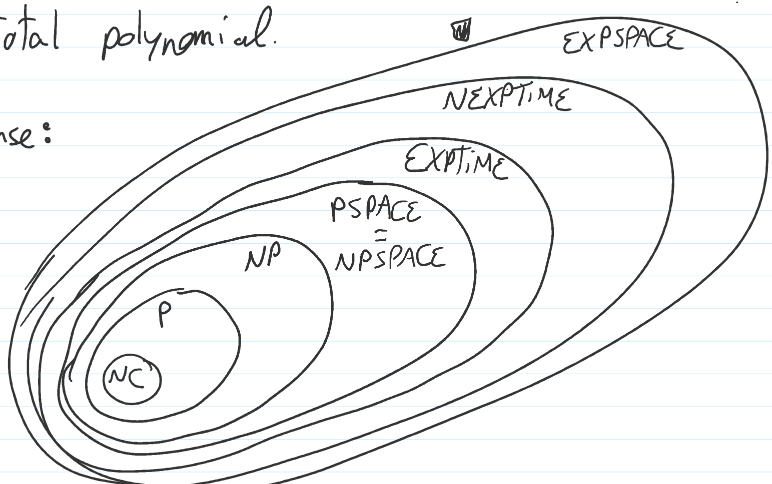
Preuve: soit φ une instance de SAT,
avec variables $x_1, x_2, \ldots, x_n$

pour chaque assignation possible des x:
- si l'assignation satisfait φ
  - return true

return false

Chaque assignation peut être enumérée en espace $O(n)$, et chacune peut être testée sur φ en temps, et donc espace, $O(n^k)$.
$\Rightarrow$ Espace total polynomial. ∎

Ce qu'on pense:



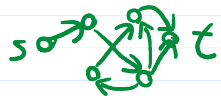
Ce qu'on sait: P $\neq$ EXPTIME (Time Hierarchy Thm) PSPACE $\neq$ EXPSPACE (Space Hierarchy Thm)

[PSPACE = NPSPACE]

Théorème de Savitch: $NSPACE(f(n)) \subseteq DSPACE(f(n)^2)$
(pour $f(n) \in \Omega(\log n)$)

Pour prouver Savitch, on a besoin de:

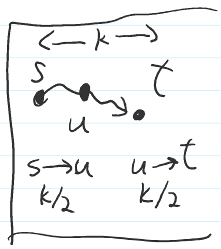
<u>Lemme</u>: Soit DPATH = $\{\langle G, s, t, k\rangle :$ G est un graphe orienté et $\exists$ chemin de s à t de longueur au plus $k\}$



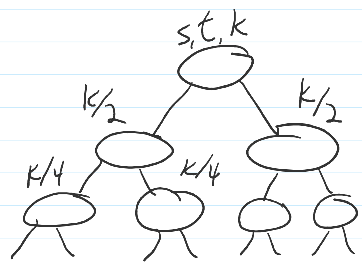
$k = 4$

Si G est encodé par une fct en espace $O(\log |V(G)|)$ qui, sur entrée $u,v \in V(G)$, retourne si $(u,v) \in E(G)$ ou pas, alors on peut décider DPATH en espace $O((\log |V(G)|)^2)$

<u>Preuve</u>: Voici un algo:




```
              dpath(G, s, t, k)
O(log n)        si k == 0, return (s == t)
O(log n)        si k == 1, return ((s,t) ∈ E(G) ou s == t)
                // brancher sur le point milieu u du chemin
                // de s à t (tester tous les u possibles)
O(log n)        pour u ∈ V(G)   // chaque u ∈ V(G) = entier entre 1 et n
O(log n)        |  si dpath(G, s, u, ⌊k/2⌋) et dpath(G, u, t, ⌈k/2⌉)
                |     └ return true
                |  fin si
                fin pour
                return false
```



Cet algo explore un arbre de récursion où

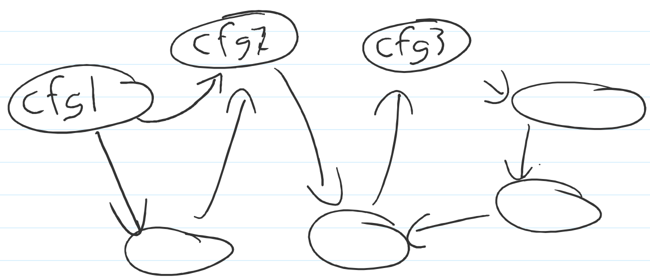
- l'espace d'un appel est $O(\log n)$ car chaque ligne peut s'implémenter en espace logarithmique.
- la profondeur de l'arbre de récursion est $O(\log n)$ car k est divisé par 2 à chaque niveau de profondeur.

$O(\log n)$ car k est divisé par 2 à chaque niveau de profondeur.

L'espace requis par l'algo est proportionnel à profondeur × espace appel $\in O(\log n \cdot \log n)$
$= O((\log n)^2)$. ∎

__Thm de Savitch:__ $NSPACE(f(n)) \subseteq DSPACE(f(n)^2)$

$L \in NSPACE(f(n))$: graphe de config



Soit $L \in NSPACE(f(n))$. Alors $\exists$ MT non-dét. M dont le langage accepté est L et qui est en espace $O(f(n))$. On a déjà établi que le # de configs possibles sur un espace $O(f(n))$ est $O(2^{d \cdot f(n)})$.

Soit G_M le graphe où
- $V(G_M)$ = l'ensemble des configs possibles
- $(c_1, c_2) \in E(G_M)$ si M permet d'aller de c_1 à c_2.
  - on peut concevoir une fonction en espace $O(f(n))$ qui détermine si $(c_1, c_2) \in E(G)$. $\boxed{f(n) \in O(\log 2^{d f(n)})}$

On note que $w \in L \iff \exists$ chemin de cfg init vers une config c_i acceptante. De plus, ce chemin a une longueur $\leq O(2^{d \cdot f(n)})$.

espace $O(f(n))$ L'algo pour décider L de façon déterministe : sur entrée w
pour chaque cfg c_i acceptante

esp. L'algo pour décider L de façon déterministe: sur entrée w

$O(f(n))$ pour chaque cfg C_i acceptante

si $dpath(G_M, cfginit, C_i, 2^{d \cdot f(n)})$

accepter w

rejeter w

Espace: stocker les C_i : $O(f(n))$

stocker $\underbrace{G_M, cfginit, C_i}_{O(f(n))}$, $\underbrace{2^{d \cdot f(n)}}_{\log(2^{d \cdot f(n)}) = d \cdot f(n) \in O(f(n))}$

Espace requis par un appel à dpath:

$$O((\log(|V(G_M)|))^2)$$
$$= O((\log(2^{d \cdot f(n)}))^2)$$
$$= O((d \cdot f(n))^2) = O((f(n))^2)$$

$\Rightarrow$ On décide L en espace $(f(n))^2$ déterministe

$\Rightarrow L \in DSPACE((f(n))^2)$.

$\Rightarrow NSPACE(f(n)) \subseteq DSPACE((f(n))^2)$. ∎

Corollaire: PSPACE = NPSPACE

Preuve: • PSPACE $\subseteq$ NPSPACE car $L \in$ PSPACE $\Rightarrow L \in$ NPSPACE.

• NPSPACE $\subseteq$ PSPACE:

soit $L \in$ NPSPACE. Alors $\exists$ MT non-dét. pour L en espace $O(n^k)$, $k \in \mathbb{N}$.

Soit $L \in NPSPACE$. Alors $\exists$ MT nondét. pour L en espace $O(n^k)$, $k \in \mathbb{N}$.
Par Savitch, L peut être décidé en espace $O((n^k)^2)$ $= O(n^{2k}) \Rightarrow L \in PSPACE$. $\blacksquare$