Théorème de SAVITCH

Antoine DEQUAY

21 septembre 2022

Notes

- Prof : Nicolas MARKEY.
- Leçon : 913, 915, 926.
- Références :
 - Computational Complexity, PAPADIMITRIOU.

Définition 1 *Problème REACH.* Soit $G = (S, A)$ un graphe et $(x, y) \in S^2$. Le problème REACH correspond à savoir s'il existe ou non un chemin entre x et y dans G .

Remarque On remarque en particulier que si un tel chemin existe, alors il en existe un qui relie x et y et qui passe au maximum une fois par chaque sommet de G .

Théorème 2 (Théorème de SAVITCH) $\text{REACH} \in \mathbf{SPACE}(\log^2(n))$.

Preuve. Soit $G = (S, A)$ un graphe, on note $n = \#S$. On se donne $(x, y) \in S^2$ et $i \in \mathbb{N}$. Pour prouver le théorème, on pose le problème $\text{PATH}(x, y, i)$, pour comme suit : $\text{PATH}(x, y, i)$ renvoie True si il existe un chemin de x à y en au plus 2^i arêtes, False sinon.

Par la remarque, on se rend compte que $\text{REACH}(x, y) = \text{PATH}(x, y, \lceil \log(n) \rceil)$.

On va se baser sur l'idée suivante pour mettre en place un algorithme de calcul de PATH , via le paradigme diviser pour régner : il existe un chemin entre x et y de longueur au plus 2^{i+1} si et seulement si il existe $z \in S$, tel qu'il existe un chemin entre x et z de longueur au plus 2^i et un chemin entre z et y de longueur au plus 2^i . C'est à dire :

$$\text{PATH}(x, y, i + 1) \iff \exists z \in S, \text{PATH}(x, z, i) \wedge \text{PATH}(z, y, i) \quad (1)$$

Construisons une machine de TURING déterministe qui résoud $\text{PATH}(x, y, i)$ via ce schéma et qui soit dans $\text{SPACE}(\log^2(n))$. Soit donc $\mathcal{M}$ une machine du TURING à deux rubans de travail. On suppose que M prends sur son entrée la matrice d'adjacence de G , dont tous les sommets sont numérotés de 1 à n , et codés en binaire, ainsi que x, y et i .

Tout au long de l'exécution de M , on imposera que le premier ruban de travail ne contienne que des triplets (u, v, i) , avec $(u, v) \in S^2$ et $i \in \llbracket 0, \lceil \log(n) \rceil \rrbracket$, u, v et i codés en binaire. Le second ruban servira aux calculs intermédiaires.

On commence par recopier au début du premier ruban le triplet (x, y, i) .

Si $i = 0$, on peut vérifier si x et y sont connectés dans G via un chemin de longueur au plus égale à $2^i = 1$ en vérifiant si $x = y$ et sinon si x et y sont connectés via la matrice d'adjacence écrite sur le ruban d'entrée.

Si $i \in \mathbb{N}^*$, on génère un à un l'ensemble de $z \in S$ (en réutilisant, à chaque nouveau z , l'espace de calcul utilisé dans le second ruban de travail pour le z précédemment utilisé). Pour un z généré, on écrit dans le premier ruban (à la suite de ce qui y est déjà écrit) le triplet $(x, z, i - 1)$ et on résoud $\text{PATH}(x, z, i - 1)$. Si la réponse est négative, on efface le triplet et on recommence avec le z suivant. Si la réponse est positive, on efface le triplet, on écrit à sa place le triplet $(z, y, i - 1)$ (on récupère la valeur de y grâce au premier triplet du premier ruban de travail). On travaille alors à la décision de $\text{PATH}(z, y, i - 1)$. Si c'est négatif, on efface le triplet et on recommence du début avec un nouveau z . Sinon, on valide $\text{PATH}(x, y, i)$ (on détecte qu'on est dans la seconde phase de l'algorithme grâce au premier triplet du premier ruban de travail).

La correction et la terminaison de cet algorithme sont claires. Comme on l'a vu plus haut, $\mathcal{M}$ résout donc $\text{REACH}(x, y)$ via $\text{PATH}(x, y, \lceil \log(n) \rceil)$. Ainsi, le premier ruban de travail possède au maximum $\lceil \log(n) \rceil$ triplets, tous de taille au plus $3\log(n)$. Le second ruban est toujours de taille en $O(\log(n))$, car il ne sert qu'à incrémenter des nombres un par un. Cela complète la preuve. □

Corollaire 3 Si $f(n) \geq \log(n)$, $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Preuve. Soit $\mathcal{M}$ une machine de TURING associée a un problème de $\text{NSPACE}(f(n))$. On considère G le graphe de configuration de $\mathcal{M}$ pour une entrée x de $\mathcal{M}$, avec $\#x = n$. Soit y l'état final de M . On résoud $\text{REACH}(x, y)$ sur le graphe G , qui contient au plus $c^{f(n)}$ noeuds pour une certaine constante c . Ainsi, un $O(f^2(n))$ suffit pour résoudre le problème via une machine déterministe, d'où $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$. □

Théorème 4 Comme $\text{SPACE}(f(n)) \subseteq \text{NSPACE}(f(n))$, on a : $\text{NSP}/\text{EXPPACE} = \text{P}/\text{EXPSPACE}$.