Théorème d'Immerman-Szelepcsenyi

Réfs : Sipser : Introduction to the theory of computation
Perifel : Complexité algorithmique

Prop : PATH est NL-complet

Démo
- PATH ∈ NL car il suffit que la machine non-déterministe se souvienne du sommet sur lequel elle est puis choisisse au hasard un de ses voisin, et ce au plus |S| fois.

- Pour prouver la complétude, on définit d'abord ce qu'est une configuration d'une machine M sur l'entrée w : c'est la donnée de l'état, du ruban de travail, et des positions des têtes de lecture.
Le résultat clé est que si M tourne en espace $f(n)$ et que w est une entrée de taille n, le nombre de configurations différentes de M sur w est $n 2^{O(f(n))}$.
(ce résultat donne notamment les inclusions $SPACE(f(n)) \subseteq DTIME(2^{O(f(n))})$)

- Soit M une machine qui décide A en $O(\log n)$ espace. Etant donné une entrée w, on construit $\langle G, s, t \rangle$ en espace log tq G est un graphe orienté contenant un chemin de s vers t ssi M accepte w.
Les noeuds de G sont les configurations de M sur w. Etant donné une config c_1 et c_2 de M, la paire (c_1, c_2) est une arrête de G ssi c_2 est une configuration accessible depuis c_1 (en 1 pas). Le noeud s est la configuration initiale de M et la machine est modifiée pour avoir une seule configuration acceptante (après effacement total par exemple) que l'on associe au noeud t.

- Pour générer le graphe

Th : $NL = co\text{-}NL$!

Démo On montre que $\overline{PATH} \in NL$, et par complétude de PATH on aura l'égalité

* **Lemme :** Si G est un graphe orienté et s un sommet, on note S_i l'ensemble des sommets de G à distance $\leq i$ de s. Il existe une MT non-dét fonctionnant en espace log sur l'entrée $(G, s, i, |S_i|)$ tq si la machine accepte, $|S_{i+1}|$ est inscrit sur le ruban.

Dém :
- On teste de manière non-dét en espace log si un sommet u est à distance au plus i de s : Test 1(s, u, i)
- On va maintenant se servir de $|S_i|$ pour vérifier qu'on a bien atteint tous les sommets de S_i :

```
Test 2 (s, u, i, |Si|) :
    c ← 0, b ← false;
    Pour tout sommet v FAIRE
        Tester si v ∈ Si par l'algo Test 1
        Si le calcul est acceptant
            c ← c+1 ;
            Si v = u Alors b ← true;
    Si c = |Si| Alors accepter et renvoyer b
    Sinon Rejeter;
```

- On calcule $|S_{i+1}|$ à l'aide de Test 2 en comptant tous les sommets de S_i et les voisins de sommets de S_i

□

* Pour compléter la preuve, il suffit de calculer successivement, en partant de $|S_0| = 1$, les valeurs de $|S_1|, \dots, |S_n|$, où n est le nombre de sommets de G.

* Puis pour tester un sommet t, on utilise Test 2 $(s, t, n, |S_n|)$ puisque si un chemin existe entre s et t, il est de taille au plus n.

□