TD 11 – Complexité en espace

1 NL = coNL

Nous allons ici montrer un cas particulier du théorème d'Immerman-Szelepcsényi, sachant que le cas général s'en déduit sans trop de difficultés.

Rappelons que le langage

$$\mathrm{PATH} = \{\langle G, s, t\rangle \mid G \text{ est un graphe orienté contenant un chemin du sommet } s \text{ au sommet } t\}$$

est NL-complet. Pour montrer que NL = coNL, nous allons montrer que $\overline{\mathrm{PATH}} \in \mathsf{NL}$, en suivant les étapes ci-dessous.

Exercice 1. Montrer qu'il existe une MTN $\mathcal{R}$ qui termine toujours et telle que, étant donné un graphe orienté $G = (V, A)$, deux sommets $s, t \in V$ et un entier $l \in \mathbb{N}, l < |V|$, fonctionne en espace logarithmique en $|\langle G\rangle|$ sur l'entrée $\langle G, s, t, l\rangle$ et a au moins un chemin acceptant sur cette entrée si et seulement s'il existe un chemin de s à t de longueur au plus l.

Solution 1. Étant donné un graphe orienté $G = (V, A)$, deux sommets $s, t \in V$ et un entier $l \in \mathbb{N}, l < |V|$, la MTN $\mathcal{R}$ se comporte selon l'algorithme suivant sur l'entrée $\langle G, s, t, l\rangle$:

si $s = t$ **alors**
  accepter
fin si
$x \leftarrow s$
$i \leftarrow 0$
tant que $i < l$ **faire**
  $i \leftarrow i + 1$
  choisir $y \in V$
  si $(x, y) \notin A$ **alors**
    rejeter
  sinon
    $x \leftarrow y$
    si $x = t$ **alors**
      accepter
    fin si
  fin si
fin tant que
rejeter .

Étant donné un graphe orienté $G = (V, A)$ et un sommet $s \in V$, pour tout $l \in \mathbb{N}, l < |V|$, on dénote par $S_l(G, s)$ l'ensemble des sommets de G vers lesquels il existe un chemin de longueur au plus l depuis s. On dénote par $S(G, s)$ l'ensemble des sommets de G vers lesquels il existe un chemin depuis s.

Exercice 2. Montrer qu'il existe une MTN $\mathcal{R}'$ qui termine toujours et telle que, étant donné un graphe orienté $G = (V, A)$, deux sommets $s, t \in V$, un entier $l \in \mathbb{N}, l < |V| - 1$ et $S_l(G, s)$, fonctionne en espace logarithmique en $|\langle G\rangle|$ sur l'entrée $\langle G, s, t, l, |S_l(G, s)|\rangle$ et vérifie :

— qu'elle a au moins un chemin acceptant sur cette entrée ;
— que tout chemin acceptant termine avec 1 inscrit sur la bande de sortie s'il existe un chemin de s à t dans G de longueur au plus $l + 1$, 0 sinon.

Indication : D'abord supposer que $\mathcal{R}$ est déterministe, puis introduire un mécanisme de comptage pour vérifier que $\mathcal{R}$ n'a bien donné que des « bonnes réponses » de manière non déterministe.

Solution 2. Étant donné un graphe orienté $G = (V, A)$, deux sommets $s, t \in V$, un entier $l \in \mathbb{N}, l < |V| - 1$ et $S_l(G, s)$, la MTN $\mathcal{R}'$ se comporte selon l'algorithme suivant sur l'entrée $\langle G, s, t, l, |S_l(G, s)|\rangle$:

$acc \leftarrow 0$
$c \leftarrow 0$
pour $v \in V$ **faire**
  si $\mathcal{R}(\langle G, s, v, l\rangle)$ accepte **alors**
    $c \leftarrow c + 1$
    si $v = t \vee (v, t) \in A$ **alors**
      $acc \leftarrow 1$
    fin si
  fin si
fin pour
si $c < |S_l(G, s)|$ **alors**
  rejeter
fin si
accepter et retourner acc .

Exercice 3. Montrer ensuite qu'il existe une MTN $\mathcal{C}$ qui termine toujours et telle que, étant donné un graphe orienté $G = (V, A)$, un sommet $s \in V$, un entier $l \in \mathbb{N}, l < |V| - 1$ et $S_l(G, s)$, fonctionne en espace logarithmique en $|\langle G\rangle|$ sur l'entrée $\langle G, s, l, |S_l(G, s)|\rangle$ et vérifie :

— qu'elle a au moins un chemin acceptant sur cette entrée ;
— que tout chemin acceptant termine avec $\langle\, |S_{l+1}(G, s)|\rangle$ inscrit sur la bande de sortie.

Solution 3. Étant donné un graphe orienté $G = (V, A)$, un sommet $s \in V$, un entier $l \in \mathbb{N}, l < |V| - 1$ et $S_l(G, s)$, la MTN $\mathcal{C}$ se comporte selon l'algorithme suivant sur l'entrée $\langle G, s, l, |S_l(G, s)|\rangle$:

$r \leftarrow 0$
pour $v \in V$ **faire**
  si $\mathcal{R}'(\langle G, s, v, l, |S_l(G, s)|\rangle)$ accepte **alors**
    si valeur retournée $= 1$ **alors**
      $r \leftarrow r + 1$
    fin si
  sinon
    rejeter
  fin si
fin pour
accepter et retourner r .

Exercice 4. En déduire qu'il existe une MTN $\mathcal{C}'$ qui termine toujours et telle que, étant donné un graphe orienté $G = (V, A)$ et un sommet $s \in V$, fonctionne en espace logarithmique en $|\langle G\rangle|$ sur l'entrée $\langle G, s\rangle$ et vérifie :

— qu'elle a au moins un chemin acceptant sur cette entrée ;
— que tout chemin acceptant termine avec $\langle\, |S(G, s)|\rangle$ inscrit sur la bande de sortie.

Solution 4. Étant donné un graphe orienté $G = (V, A)$ et un sommet $s \in V$, la MTN $\mathcal{C}'$ se comporte selon l'algorithme suivant sur l'entrée $\langle G, s\rangle$:

$r \leftarrow 1$
$l \leftarrow 0$
tant que $l < |V| - 1$ **faire**
  si $\mathcal{C}(\langle G, s, l, r\rangle)$ accepte **alors**
    $r \leftarrow$ valeur retournée
  sinon
    rejeter
  fin si
  $l \leftarrow l + 1$

fin tant que
accepter et retourner r .

Exercice 5. Montrer, en utilisant $\mathcal{C}'$, que $\overline{\text{PATH}} \in \mathsf{NL}$.

Solution 5. Étant donné un graphe orienté $G = (V, A)$ et deux sommets $s, t \in V$, on considère le graphe orienté $G' = (V, A')$ où $A' = A \cup \{(t, v) \mid v \in V \setminus \{t\}\}$. Il est possible de calculer $\langle G' \rangle$ étant donné $\langle G \rangle$ en espace logarithmique avec une MT $\mathcal{T}$.

Pour montrer que $\overline{\text{PATH}} \in \mathsf{NL}$, on construit une MTN $\mathcal{M}$ telle que si l'entrée ne correspond pas à un encodage valide d'un triplet (G, s, t) avec $G = (V, A)$ un graphe orienté et $s, t \in V$ deux sommets, elle accepte, sinon, étant donné un graphe non orienté $G = (V, A)$ et deux sommets $s, t \in V$, se comporte selon l'algorithme suivant sur l'entrée $\langle G, s, t \rangle$:

si $\mathcal{C}'(\langle \mathcal{T}(\langle G \rangle), s, t \rangle)$ accepte **alors**
  si valeur retournée $< |V|$ **alors**
    accepter
  sinon
    rejeter
  fin si
sinon
  rejeter
fin si

Puisque la composition de deux MTN en espace logarithmique peut se faire en espace logarithmique, il s'ensuit que $\mathcal{M}$ fonctionne en espace logarithmique. Par conséquent, on a $\overline{\text{PATH}} \in \mathsf{NL}$.

Exercice 6. En conclure que $\mathsf{NL} = \mathsf{coNL}$.

Solution 6. Puisque PATH est NL-complet, on a que $\overline{\text{PATH}}$ est coNL-complet. Par conséquent, puisque $\overline{\text{PATH}} \in \mathsf{NL}$, que tout langage de coNL se réduit en espace logarithmique à $\overline{\text{PATH}}$ et que NL est clôt par réduction en espace logarithmique, il s'ensuit que $\mathsf{coNL} \subseteq \mathsf{NL}$.

Soit maintenant $L \in \mathsf{NL}$. On a donc que $\bar{L} \in \mathsf{coNL}$ et donc que $\bar{L} \in \mathsf{NL}$ par ce que nous venons de montrer. Ainsi, $L \in \mathsf{coNL}$. On en conclut que $\mathsf{NL} \subseteq \mathsf{coNL}$ et donc que $\mathsf{NL} = \mathsf{coNL}$.

2 NL-complétude

Exercice 7.

1. Montrer que le langage
$$\text{L}_{\text{DFA}} = \{\langle \mathcal{A}, w \rangle \mid \mathcal{A} \text{ est un AFD sur un alphabet } \Sigma \text{ acceptant } w \in \Sigma^*\}$$
est dans L.
2. Montrer que le langage
$$\text{L}_{\text{NFA}} = \{\langle \mathcal{A}, w \rangle \mid \mathcal{A} \text{ est un AFN sur un alphabet } \Sigma \text{ acceptant } w \in \Sigma^*\}$$
est NL-complet.

Solution 7.

1. Étant donné un AFD $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ et un mot $w \in \Sigma^*$, vérifier si $\mathcal{A}$ accepte w revient simplement, si l'on voit $\mathcal{A}$ comme un graphe, à tester s'il existe un chemin étiqueté par w de q_0 à un état de F. Cela peut se faire avec une MT en espace logarithmique puisque pour chaque état, il n'y a qu'un seul choix de successeur possible en utilisant un arc étiqueté par une certaine lettre $a \in \Sigma$.

   Par conséquent, $\text{L}_{\text{DFA}} \in \mathsf{L}$.

2. Étant donné un AFN $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ et un mot $w \in \Sigma^*$, vérifier si $\mathcal{A}$ accepte w revient, tout aussi simplement, si l'on voit $\mathcal{A}$ comme un graphe, à tester s'il existe un chemin étiqueté par w de q_0 à un état de F, ce qui peut se faire avec une MTN en espace logarithmique. Il s'ensuit que $\mathrm{L}_{\mathrm{NFA}} \in \mathsf{NL}$.

   Pour montrer que $\mathrm{L}_{\mathrm{NFA}}$ est NL-difficile, on donne une réduction en espace logarithmique de PATH à $\mathrm{L}_{\mathrm{NFA}}$ que nous décrivons à présent. À tout mot w qui ne correspond pas à un encodage valide d'un triplet (G, s, t) avec $G = (V, A)$ un graphe orienté et $s, t \in V$ deux sommets, on associe un mot qui n'est trivialement pas dans $\mathrm{L}_{\mathrm{NFA}}$. Autrement, à $\langle G, s, t\rangle$ où $G = (V, A)$ est un graphe orienté et $s, t \in V$ sont deux sommets, on associe $\langle \mathcal{A}, 1^{|V|-1}\rangle$ où $\mathcal{A}$ est l'AFN $(V, \{1\}, \delta, s, \{t\})$ avec $\delta \colon V \times \{1, \varepsilon\} \to \mathfrak{P}(V)$ telle que pour tout $v \in V$,

$$\delta(v, \varepsilon) = \emptyset$$

   et

$$\delta(v, 1) = \begin{cases} \{u \in V \mid (t, u) \in A\} \cup \{t\} & \text{si } v = t \\ \{u \in V \mid (v, u) \in A\} & \text{sinon} \end{cases} .$$

   On peut montrer que cette réduction se fait en espace logarithmique et que, donc, $\mathrm{L}_{\mathrm{NFA}}$ est NL-difficile.

   On en conclut que $\mathrm{L}_{\mathrm{NFA}}$ est NL-complet.

Exercice 8. Montrer que 2 - SAT est NL-complet.

Solution 8. En utilisant la même astuce utilisée pour montrer que 2 - SAT $\in \mathsf{P}$, on peut montrer que $\overline{\text{2-SAT}} \in \mathsf{NL}$, et donc en déduire que 2 - SAT $\in \mathsf{NL}$.

Montrons maintenant que $\overline{\text{2-SAT}}$ est NL-difficile en donnant une réduction en espace logarithmique de PATH à $\overline{\text{2-SAT}}$ que nous décrivons maintenant. À tout mot w qui ne correspond pas à un encodage valide d'un triplet (G, s, t) avec $G = (V, A)$ un graphe orienté et $s, t \in V$ deux sommets, on associe un mot qui n'est trivialement pas dans $\overline{\text{2-SAT}}$. Autrement, à $\langle G, s, t\rangle$ où $G = (V, A)$ est un graphe orienté et $s, t \in V$ sont deux sommets, on associe la formule CNF φ sur l'ensemble de variables V telle que $\varphi = s \wedge \bigwedge_{(v,u)\in A} (\neg v \vee u) \wedge \neg t$, qui n'est pas satisfaisable si et seulement s'il existe un chemin de s à t dans G. On peut montrer que cette réduction se fait en espace logarithmique et que, donc, $\overline{\text{2-SAT}}$ est NL-difficile. Cela implique que 2 - SAT est coNL-difficile, et donc NL-difficile.

On en conclut que 2 - SAT est NL-difficile.

Un *semi-groupe* est la donnée $(S, \star)$ d'un ensemble S et d'une loi de composition interne $\star \colon S \times S \to S$ tels que $\star$ est associative : $\forall x, y, z \in S, (x \star y) \star z = x \star (y \star z)$.

Étant donné deux semi-groupes $(S, \star)$ et $(T, \diamond)$, on dit que $(T, \diamond)$ est un sous-semi-groupe de $(S, \star)$ lorsque $T \subseteq S$ et $\diamond = \star|_T$.

Enfin, pour un semi-groupe $(S, \star)$ et un sous-ensemble E donnés, on appellera le *sous-semi-groupe de* $(S, \star)$ *engendré par* E le plus petit sous-semi-groupe de $(S, \star)$ contenant E, relativement à l'inclusion.

Exercice 9. Montrer que le langage

$$\mathrm{SGEN} = \left\{ \langle S, \star, E, x\rangle \;\middle|\; \begin{array}{l} (S, \star) \text{ est un semi-groupe fini, } E \subseteq S \text{ et } x \in S \text{ appartient au} \\ \text{sous-semi-groupe de } (S, \star) \text{ engendré par } E \end{array} \right\}$$

est NL-complet.

Solution 9. Montrer que SGEN $\in \mathsf{NL}$ n'est pas très compliqué, car décider si $\langle S, \star, E, x\rangle \in$ SGEN pour $(S, \star)$ un semi-groupe fini, $E \subseteq S$ et $x \in S$ revient a décider si $\langle G, s, x\rangle \in$ PATH, où $G = (V, A)$ est un graphe orienté tel que $V = S \cup \{s\}$, où s est un élément n'apparaissant pas dans S, et

$$A = \{(s, e) \mid e \in E\} \cup \{(y, z) \in S^2 \mid \exists e \in E, y \star e = z\} .$$

Pour montrer que SGEN est NL-difficile, on donne une réduction en espace logarithmique de PATH à SGEN que nous décrivons à présent. À tout mot w qui ne correspond pas à un encodage valide d'un triplet (G, s, t) avec $G = (V, A)$ un graphe orienté et $s, t \in V$ deux sommets, on associe un mot qui n'est trivialement pas dans SGEN. Autrement, à $\langle G, s, t\rangle$ où $G = (V, A)$ est un graphe orienté et $s, t \in V$ sont deux sommets, on associe $\langle S, \star, E, (s, t)\rangle$ où $S = V^2 \cup \{\bot\}$ avec $\bot$ un élément n'apparaissant pas dans V^2, où $E = A \cup \{(s, s)\}$ et $\star \colon S^2 \to S^2$ est telle que :

— $\bot \star \bot = \bot$;

— pour tout $e \in S^2$, $e \star \bot = \bot \star e = \bot$;

— pour tous $(a, b), (c, d) \in S^2$, $(a, b) \star (c, d) = \begin{cases} (a, d) & \text{si } b = c \\ \bot & \text{sinon} \end{cases}$.

On peut montrer que cette réduction se fait en espace logarithmique et que donc, SGEN est NL-difficile.

On en conclut que SGEN est NL-complet.

3 PSPACE-complétude

Soit $G = (V, A)$ un graphe orienté et $s \in V$. Le *jeu de géographie* sur (G, s) est un jeu à deux joueurs fonctionnant de la manière suivante. Au début de la partie, un marqueur est placé sur le sommet s. Les joueurs jouent tour à tour, le joueur 1 commençant en premier ; à chaque tour, le joueur dont c'est le tour de jouer déplace le marqueur placé sur un sommet x sur un autre sommet y tel que $(x, y) \in A$ et que le marqueur n'ait jamais été placé sur y auparavant ; le premier joueur qui ne peut plus déplacer le marqueur perd, ce qui termine la partie.

Pour le joueur 1, une *stratégie* est une fonction qui à toute suite finie de choix légaux de sommets $(s, x_1, y_1, \ldots, x_l, y_l)$ $(l \in \mathbb{N})$ sur lesquels est successivement placé le marqueur, sans que cette suite n'aboutisse à la fin de la partie, associe un nouveau sommet x_{l+1} tel que $(s, x_1, y_1, \ldots, x_l, y_l, x_{l+1})$ soit une suite finie de choix légaux de sommets sur lesquels est successivement placé le marqueur. Le joueur 1 a une *stratégie gagnante* lorsqu'il existe une stratégie qui lui permet de gagner toute partie (c'est-à-dire, quels que soient les choix du joueur 2).

Exercice 10. Montrer que le langage

$$\text{GEOGRAPHY} = \left\{ \langle G, s\rangle \;\middle|\; \begin{array}{l} G \text{ est un graphe orienté et } s \text{ un de ses sommets tels que le joueur 1 a} \\ \text{une stratégie gagnante pour le jeu de géographie sur } (G, s) \end{array} \right\}$$

est PSPACE-complet.

Solution 10. Montrer que GEOGRAPHY $\in \mathsf{PSPACE}$ se fait en construisant une MT mettant en œuvre un algorithme de « backtracking ».

Pour montrer que GEOGRAPHY est PSPACE-difficile, on donne une réduction en temps polynomial de TQBF à GEOGRAPHY décrite dans la suite. À tout mot w qui ne correspond pas à un encodage valide d'une formule booléenne quantifiée, on associe un mot qui n'est trivialement pas dans GEOGRAPHY. Autrement, soit une formule booléenne quantifiée φ, que l'on supposera, sans perte de généralité, être de la forme

$$\exists x_1 \forall x_2 \cdots \exists x_{n-1} \forall x_n (C_1 \wedge C_2 \wedge \cdots \wedge C_m)$$

avec $n \in \mathbb{N}_{>0}$ pair, $m \in \mathbb{N}_{>0}$ et les $C_1, C_2, \ldots, C_m$ étant des clauses. À $\langle\varphi\rangle$ on associera $\langle G, s\rangle$ où $G = (V, A)$ est un graphe orienté avec $V = \bigcup_{i \in [\![1,n]\!]} \{x_i, \neg x_i, y_i, z_i\} \cup \{C_j \mid j \in [\![1, m]\!]\}$ et

$$A = \begin{array}{l} \displaystyle\bigcup_{i \in [\![1,n]\!]} \{(y_i, x_i), (y_i, \neg x_i), (x_i, z_i), (\neg x_i, z_i)\} \cup \{(z_i, y_{i+1}) \mid i \in [\![1, n-1]\!]\} \cup \{(z_n, C_j) \mid j \in [\![1, m]\!]\} \cup \\ \{(C_j, \neg x_i) \mid x_i \text{ apparaît dans } C_j\} \cup \{(C_j, x_i) \mid \neg x_i \text{ apparaît dans } C_j\} \end{array},$$

et $s = y_1$. On peut vérifier que φ est vraie si et seulement si le joueur 1 a une stratégie gagnante pour le jeu de géographie sur (G, s). Cette réduction se fait en espace polynomial, d'où GEOGRAPHY est PSPACE-difficile.

On en conclut que GEOGRAPHY est PSPACE-complet.