Complexité : TD 3

Chargé de TD : Adrien Koutsos

Exercice 1 : Planification

On note $I = \{1, \dots, n\}$. Un problème de planification est donné par :
- n variables booléennes $\{x_i\}_{i \in I}$;
- m opérations où chaque opération est définie par une condition de la forme $\bigwedge_{i \in I'} x_i = \alpha_i$ avec $I' \subseteq I$ et une mise à jour de la forme $\{x_i \leftarrow \beta_i\}_{i \in I''}$ avec $I'' \subseteq I$.
  Voici un exemple d'opération : Si $x_1 = \mathtt{V} \wedge x_3 = \mathtt{F}$ Alors $x_1 \leftarrow \mathtt{F}; x_2 \leftarrow \mathtt{V}$
- une configuration initiale s_{init} et une configuration finale s_{fin} où une configuration est un assignement de valeurs (i.e. une valuation) aux variables.
- Une opération est applicable à une configuration si la condition de l'opération s'évalue à $\mathtt{V}$. Son application consiste à effectuer ses mises à jour pour obtenir la nouvelle configuration. Par exemple l'opération précédente est applicable à la configuration $(\mathtt{V}, \mathtt{F}, \mathtt{F})$ et conduit à la configuration $(\mathtt{F}, \mathtt{V}, \mathtt{F})$.

Le problème consiste à déterminer s'il existe une suite d'applications des opérations (avec éventuellement plusieurs applications d'une même opération) qui conduise de la configuration initiale à la configuration finale.

Question 1. Montrez que le problème de planification est dans PSPACE.

Question 2. Montrez que le problème de planification est PSPACE-difficile. Pour cela, on montrera une réduction LOGSPACE du problème ci-dessous vers le problème de planification :
Donnée : Une machine $\mathcal{M}$ calculant en espace n^k et un mot w.
Question : $\mathcal{M}$ accepte w.
(Hint : pour simplifier, on supposera que $\mathcal{M}$ a une seule bande, qui sert de bande d'entrée, de travaille et de sortie.)

Exercice 2 : Vacuité d'un Automate Déterministe Concurrent

Un automate déterministe concurrent $\mathcal{A}$ est donné par un ensemble d'automates déterministes $\{\mathcal{A}_i\}_{i \leq n}$ appelés composants. Un état de l'automate déterministe concurrent est un tuple $(s_1, \dots, s_n)$ composé d'un état par composant. Lorsqu'une lettre a est lue, les automates $\mathcal{A}_i$ qui ont une transition issue de s_i étiquetée par la lettre effectuent simultanément leur transition tandis que les autres conservent leur état. Pour qu'une lettre puisse être lue, au moins un automate doit effectuer une transition. Un mot est reconnu s'il conduit à un tuple d'états terminaux. Le problème de la vacuité consiste à savoir s'il existe au moins un mot accepté par l'automate.

Question 3. Montrez que le problème de la vacuité des automates déterministes concurrents est dans PSPACE.

Question 4. Montrez que le problème de la vacuité est PSPACE-difficile.

Exercice 3 : Chemin dans un Graphe Pondéré à Poids Relatifs

On considère des graphes pondérés de la forme $G = (V, E, p)$ où les arêtes de $E \subseteq V \times V$ sont orientées et portent chacune un poids, un entier *relatifs* donné par $p : E \to \mathbb{Z}$.

On commence par définir ou rappeler quelques notions et notations qui seront utiles dans la suite de l'énoncé et dans vos solutions : Pour une arête $e = (u, v) \in E$, on note ${}^\bullet e$ pour u et $e^\bullet$ pour v. Un *chemin de longueur ℓ dans G* est un mot $\rho = e_1 \cdots e_\ell \in E^+$ composé de $\ell > 0$ arêtes de E et tel que ${e_{i-1}}^\bullet = {}^\bullet e_i$ pour tout $i = 2, \ldots, \ell$. Si $\rho = e_1 \cdots e_\ell$ est un chemin, les notations ${}^\bullet\rho$ et $\rho^\bullet$ désignent ${}^\bullet e_1$ et ${e_\ell}^\bullet$ respectivement. Le poids $p(\rho)$ d'un chemin est la somme $\sum_{i=1}^{\ell} p(e_i)$ des poids de ses arêtes.

Un chemin $\rho = e_1 \ldots e_\ell \in E^+$ est un *cycle* si ${e_l}^\bullet = {}^\bullet e_1$ et le cycle est *élémentaire* si les sommets ${}^\bullet e_1, \ldots, {}^\bullet e_\ell$ sont tous distincts.

Le problème WEIGHTEDPATH a comme input un graphe pondéré G, deux sommets $u, v \in E$, un poids $a \in \mathbb{N}$. Il s'agit de décider s'il existe dans G un chemin allant de u à v et de poids total a.

Question 5. On dit qu'un chemin ρ est *factorisé en cycles* si ρ est écrit sous la forme

$$\rho = \rho_0 \sigma_1^{k_1} \rho_1 \sigma_2^{k_2} \cdots \rho_{r-1} \sigma_r^{k_r} \rho_r$$

telle que les facteurs $\sigma_1, \ldots, \sigma_r$ sont des cycles élémentaires, les entiers $k_1, \ldots, k_r$ sont non nuls et les facteurs $\rho_0, \ldots, \rho_r$ n'ont aucun facteur qui soit un cycle. (La notation w^k avec $k \in \mathbb{N}$ dénote la concaténation de k copies de w, avec $w^0 = \epsilon$ et $w^{k+1} = w^k \cdot w$).

Montrez que tout chemin admet une factorisation en cycles.

Question 6. Montrez que si G admet un chemin de poids a allant de s à t alors il existe en particulier un tel chemin avec une factorisation en cycles $\rho_0 \sigma_1^{k_1} \rho_1 \sigma_2^{k_2} \cdots \rho_{r-1} \sigma_r^{k_r} \rho_r$ telle que les σ_i's aient tous des poids $p(\sigma_i)$ différents.

Question 7. Pour $G = (V, E, p)$ avec $p : E \to \mathbb{Z}$, on notera k le nombre $|V|$ de sommets, m le nombre $|E|$ d'arêtes, et $P = \max_{e \in E} |p(e)|$ le plus grand poids (en valeur absolue). Ainsi la donnée du graphe utilise un espace mémoire en $O(k + m\lceil \log_2(P) \rceil)$.

Donnez un polynôme à quatre variables $Q(x_1, x_2, x_3, x_4)$ tel que pour toute instance $\langle G, u, v, a \rangle$, si G a un chemin de poids a reliant u à v alors il existe un tel chemin de longueur bornée par $Q(k, m, P, a)$.

Question 8. Montrer que le problème WEIGHTEDPATH est NP-complet.

Exercice 4 : Géographie

Un problème de géographie est donné par :

- Un graphe orienté $G = (V, E)$ avec un sommet initial v_0 ;
- Deux joueurs Alice et Bob qui jouent à tour de rôle en commençant par Alice ;
- Une configuration est donnée par l'ensemble des sommets déjà visités V', le sommet courant $v \in V'$, le joueur qui doit jouer P ;
- Lorsque P joue, il choisit un sommet non encore visité (i.e. $v' \notin V'$) et accessible depuis v (i.e. $(v, v') \in E$). La nouvelle configuration est donnée par le sous-ensemble $V \cup \{v'\}$, le sommet courant v' et le joueur $P' \neq P$. S'il n'existe pas de sommet v' vérifiant ces conditions alors P' gagne.

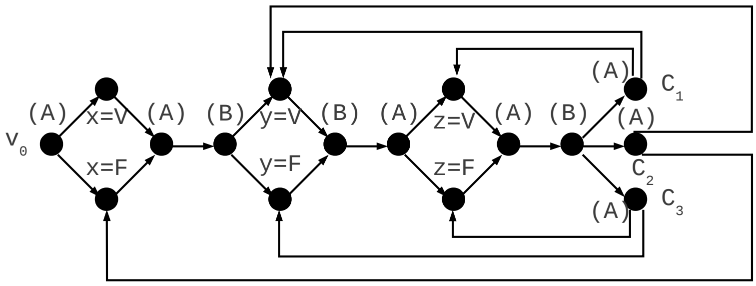

FIGURE 1 – Réduction de $QSAT$ vers Géographie

Le problème consiste à déterminer si Alice a une stratégie gagnante dans la configuration initiale $(\{v_0\}, v_0, Alice)$. On rappelle que pour ce type de jeu, dans une configuration quelconque l'un des deux joueurs a forcément une configuration gagnante.

Question 9. Montrez que le problème de géographie est dans PSPACE en proposant un algorithme récursif qui prend en entrée une configuration quelconque et qui renvoie `V` si le joueur courant a une stratégie gagnante. Vous devrez détailler votre analyse de complexité spatiale.

Question 10. Montrez que le problème de géographie est PSPACE-difficile en réduisant le problème de $QSAT$ au problème de géographie t.q. la formule est vraie ssi Alice gagne. Un exemple de réduction est représenté sur la figure 1 pour la formule :

$$\exists x \ \forall y \ \exists z \ C_1 \wedge C_2 \wedge C_3$$

avec $C_1 = \neg y \vee \neg z$, $C_2 = x \vee \neg y$, $C_3 = y \vee z$ Nous avons indiqué entre parenthèses lorsque cela était possible le joueur associé à un sommet quelque soit le déroulement du jeu. Notez aussi qu'un sommet étiqueté par une clause a pour successeurs les sommets étiquetés par les négations de ses littéraux. Il s'agit pour vous de généraliser et de justifier la réduction.