Université Bordeaux – Master Sciences et Technologies, Informatique. Session automne 2015

DS Automates et Complexité, 2 novembre 2015, 16h15 – 18h15

Documents autorisés : notes de cours et de TD.
La notation attachera une grande importance à la clarté et à la concision des justifications.
Le barème est indicatif.

Exercice 1 — 5 points. Répondez aux questions suivantes en **justifiant brièvement** vos réponses :

a) Existe-t-il des problèmes à la fois semi-décidables et indécidables ?

b) On considère le langage L des codes des machines de Turing qui acceptent si l'entrée représente un nombre pair. Le langage L est-il décidable ou indécidable ?

c) On considère la variante DNF-SAT du problème de satisfaisabilité où les formules données en entrée sont en *forme normale disjonctive*, c'est-à-dire, sont disjonctions de clauses conjonctives. Par exemple, la formule suivante est en forme normale disjonctive : $(x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y) \vee (\neg y \wedge y \wedge t \wedge \neg z)$. Le problème DNF-SAT est-il dans la classe de complexité **P** ou est-il **NP**-complet ?

d) Est-il vrai que tout problème dans **P** se réduit au langage universel $\text{UNIV} = \{\langle M, w\rangle \mid M \text{ accepte } w\}$?

e) On propose de définir la classe des problèmes **P**-complets de la façon suivante. Un problème est **P**-complet si il satisfait les deux conditions suivantes :
   — Il est dans **P**.
   — Tout problème de **P** se réduit polynomialement à lui.

   Cette définition est mauvaise. Expliquez pourquoi. ■

Solution.

a) Oui, on a vu en cours et TD plusieurs tels problèmes : par exemple, le problème de correspondance de Post, le problème Univ de la question d), ou le problème de déterminer si une machine de Turing accepte le mot vide.
   Chacun de ces problèmes est indécidable. Mais pour chacun d'eux, on peut écrire une machine de Turing qui s'arrête sur toute instance positive en acceptant, et soit rejette soit boucle sur les autres instances. Par exemple, pour le problème de correspondance de Post, il suffit d'énumérer les suites d'indices et de tester pour chacune si elle donne une solution.

b) Le langage L est indécidable, c'est une application directe du théorème de Rice. Il s'agit de savoir si $L(M)$ est égal à $(0+1)^*0$, ce qui est une propriété non triviale.

c) Il est très facile de tester en temps polynomial si une formule DNF est satisfaisable. Pour qu'elle le soit, il faut et il suffit qu'une des clauses conjonctives le soit, par définition de la disjonction $\vee$. Une telle clause est satisfaisable si et seulement si elle ne contient pas une variable et sa négation, par définition de la conjonction $\wedge$, ce qui se teste en temps linéaire. Le problème peut donc être résolu en temps linéaire, donc il est dans **P**.

d) Oui, et même tout problème *décidable* se réduit à Univ. En effet, si P est un problème décidable, il existe une machine de Turing M_P qui s'arrête toujours et répond OUI sur les instances positives de P et NON sur les instances négatives. La réduction de P à Univ est très simple : étant donnée une entrée x du problème P, on associe le couple $\langle M_P, x\rangle$ qui est bien une entrée de Univ. On a alors

$$x \text{ est une instance positive de } P \iff M_P \text{ accepte } x \iff \langle M_P, x\rangle \in \textsc{Univ}.$$

e) Cette définition est mauvaise car tout problème A de **P** se réduit polynomialement à tout problème non trivial B de **P**. Donc, avec cette définition, tout problème non trivial B de **P** serait **P**-complet. La définition serait donc équivalente à "être dans **P** et non trivial", et n'aurait donc aucun intérêt.
   Pour montrer l'affirmation que tout problème A de **P** se réduit polynomialement à tout problème non trivial B de **P**, soient A et B deux tels problèmes. On veut une réduction polynomiale f telle que pour tout x, x est une instance positive de A si et seulement si $f(x)$ est une instance positive de B.

Comme B n'est pas trivial, il a une instance positive p et une instance négative n. La réduction f est définie ainsi :

— si x est une instance positive de A, alors $f(x) = p$.

— si x est une instance négative de A, alors $f(x) = n$.

Cette réduction est bien polynomiale, car pour calculer f, il suffit de tester si x est une instance positive ou négative de A, ce qui se fait en utilisant un algorithme polynomial pour A, qui existe puisque $A \in \mathbf{P}$.

Note : dans cette preuve, on n'a même pas utilisé le fait que B est dans $\mathbf{P}$, seulement le fait qu'il est non trivial. On a donc montré que tout problème de $\mathbf{P}$ se réduit polynomialement à tout problème non trivial.

Exercice 2 — 3 points. On considère une variante des machines de Turing déterministes avec une seule bande dont la tête de lecture ne peut se déplacer *que vers la droite*, c'est-à-dire, ses transitions sont toujours de la forme $\delta(p, a) = (q, b, \triangleright)$ pour n'importe quelle paire p, a.

Montrer que le problème de l'arrêt $\text{HALT} = \{\langle M, w\rangle \mid M \text{ s'arrête sur l'entrée } w\}$, restreint à ces machines, est décidable. ■

Solution. On va montrer que si M (une machine qui va toujours à droite) ne s'arrête pas après $|w| + |Q|$ pas de calcul, alors elle ne s'arrêtera jamais.

En admettant temporairement ce résultat, cela nous donne un algorithme pour tester si M s'arrête sur w : on simule M pendant (au plus) $|w| + |Q|$ pas de calcul sur l'entrée w, et

— si lors de cette simulation, M s'est arrêtée, alors elle s'arrête évidemment sur w,

— sinon, grâce à l'affirmation admise temporairement, M ne s'arrêtera jamais sur w.

Il reste à montrer l'affirmation. Supposons que M ne s'arrête pas pendant les $|w|+|Q|$ premiers pas de calcul de M sur w et montrons que M ne s'arrêtera jamais sur w. Comme M va toujours à droite, après avoir effectué les $|w|$ premiers pas de calcul, ses têtes se trouveront toutes sur des cases blanches (pour la tête de lecture, sur la première case blanche qui suit w), et ensuite, ne rencontreront plus que des cases blanches. La machine effectue ensuite sans s'arrêter encore $|Q|$ pas de calcul, elle rencontre donc $|Q| + 1$ états intermédiaires le long de ces Q pas de calcul, en ayant toujours comme lettre sous sa tête de lecture le symbole "blanc" (puisqu'elle continue à aller à droite). D'après le "principe des tiroirs", un état q sur cette portion du calcul est répété deux fois. Comme les lettres sous les têtes de lecture sont des "blancs" dans les deux cas où M est dans l'état q, la machine M, qui est déterministe, va devoir répéter à l'infini la portion de calcul entre les deux occurrences de q, et donc le calcul boucle.

Exercice 3 — 6 points. Dans cet exercice, on considère des graphes non orientés. Étant donné un graphe G, un *chemin Hamiltonien* pour G est un chemin dans G qui passe une et une seule fois par chacun de ses sommets. De plus, on appelle *circuit Hamiltonien* pour G, un chemin Hamiltonien fermé, c'est-à-dire qu'il existe une arrête entre entre le dernier sommet du chemin et le premier.

On appelle Chemin Hamiltonien et Circuit Hamiltonien les problèmes associés : ils ont pour entrée un graphe non orienté G et demandent s'il existe un chemin Hamiltonien pour G et un circuit Hamiltonien pour G, respectivement.

On rappelle aussi que le problème Sous-Graphe prend deux graphes non orientés G, H en entrée et demande si H est un *sous-graphe* de G (H est un sous-graphe de G s'il peut être obtenu en enlevant quelques sommets et quelques arêtes de G).

Donner des réductions en temps polynomial

a) de Circuit Hamiltonien à Sous-Graphe,

b) de Chemin Hamiltonien à Circuit Hamiltonien,

c) de Circuit Hamiltonien à Chemin Hamiltonien. ■

Solution.

a) Soit G un graphe. On veut construire en temps polynomial à partir de G deux graphes G_1, G_2 tels que G a un circuit Hamiltonien si et seulement si G_2 est un sous-graphe de G_1. Il suffit de choisir $G_1 = G$ et G_2 le cycle de longueur n, où n est le nombre de sommets de G. La réduction est clairement polynomiale.

b) Soit G un graphe. On veut construire en temps polynomial à partir de G un graphe G' tel que

$$G \text{ a un chemin Hamiltonien ssi } G' \text{ a un circuit Hamiltonien.}$$

Il suffit de prendre pour G' le graphe G auquel on ajoute un nouveau sommet s relié à tous les autres sommets de G. La construction de G' est clairement polynomiale, montrons que G' satisfait l'équivalence voulue.

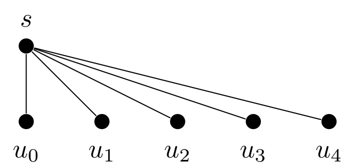
Si G a un chemin Hamiltonien s_1—s_2—$\cdots$—s_n alors dans G' on a bien le circuit s—s_1—s_2—$\cdots$—s_n—s. Inversement si G' a un circuit Hamiltonien, il doit passer par s par définition. On peut donc l'écrire s—s_1—s_2—$\cdots$—s_n—s où $s_1, \ldots, s_n$ sont les autres sommets de G', c'est-à-dire les sommets de G. Par définition de G', on a donc dans le graphe G le chemin s_1—s_2—$\cdots$—s_n, qui passe bien par tous les sommets de G et jamais deux fois par le même, puisque le circuit s—s_1—s_2—$\cdots$—s_n—s de G' est Hamiltonien. On a donc trouvé un chemin Hamiltonien dans G.

Attention : dans cette question, on ne peut pas supposer connaître les extrémités d'un chemin Hamiltonien pour construire le graphe G', pour la simple raison que G n'a peut-être pas de chemin Hamiltonien.

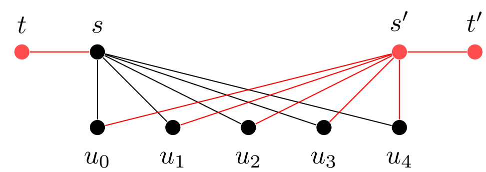
c) Soit G un graphe. On veut construire en temps polynomial à partir de G un graphe G' tel que

G a un circuit Hamiltonien ssi G' a un chemin Hamiltonien.

On choisit un sommet s de G (n'importe lequel), et pour obtenir G', on ajoute à G trois sommets s', t, t', ainsi que les arêtes $\{s, t\}$, $\{s', t'\}$, et pour chaque arête $\{s, u\}$ dans G, on ajoute l'arête $\{s', u\}$. Graphiquement, si par exemple le sommet s est relié à 5 autres sommets de G comme sur la figure suivante,



on obtient G' en modifiant localement G de la façon suivante (les sommets et arêtes ajoutées sont en rouge) :



et en conservant le reste du graphe G. Cette transformation peut clairement être calculée en temps polynomial.

Supposons que G a un circuit Hamiltonien, qu'on peut écrire à partir du sommet s :

$$s—s_1—s_2—\cdots—s_n—s.$$

Alors G' a un chemin Hamiltonien

$$t—s—s_1—s_2—\cdots—s_n—s'—t'.$$

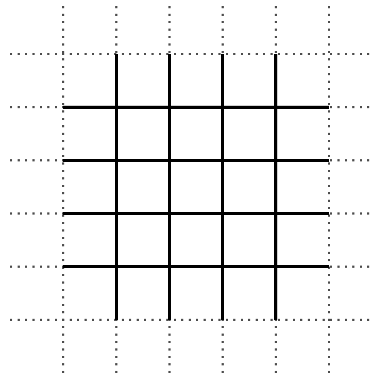
Réciproquement, si G' a un chemin Hamiltonien, celui-ci doit nécessairement partir d'un des sommets t, t' parce qu'ils sont de degré 1, et arriver à l'autre. Un tel chemin est donc de la forme

$$t—s—s_1—s_2—\cdots—s_n—s'—t'.$$

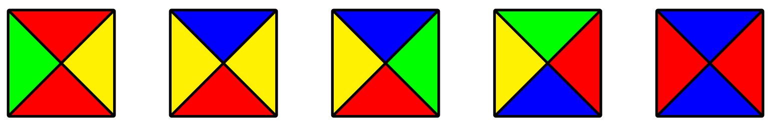
Mais dans ce cas, par construction de G', on a le circuit Hamiltonien suivant dans G :

$$s—s_1—s_2—\cdots—s_n—s.$$

Exercice 4 — 6 points. Dans cet exercice on considère deux problèmes de pavage du plan entier **infini** :

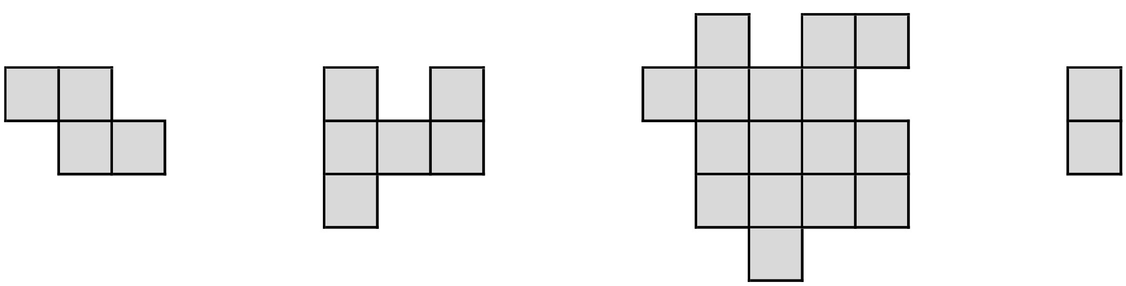


Le premier problème est une variante du problème de pavage avec des dominos de Wang, appelée Tiling. L'entrée de ce problème est un ensemble fini D de dominos de Wang, colorés sur leur côtés Nord, Est, Sud et Ouest (il n'y a pas de domino de départ imposé dans cette variante). Une entrée est par exemple :



Le problème Tiling consiste à déterminer si on peut paver le plan infini entier en utilisant uniquement des dominos de l'entrée en nombre arbitraire, mais sans changer leur orientation et en respectant les contraintes de couleur pour des dominos adjacents. Sur l'exemple, la réponse est positive, on peut même utiliser uniquement le dernier domino. On sait que ce problème est indécidable (ce qui a été vu en TD peut se généraliser à cette variante).

On appelle le second problème Tetris. Une entrée du problème Tetris est un ensemble fini P de *polyominos* de formes différentes et sans couleur. Un polyomino est une forme connexe obtenue en réunissant des dominos de même taille (*dominos unitaires*) par leurs côtés. Un exemple d'entrée est l'ensemble de polyominos suivants :



Étant donnée une entrée P, le problème Tetris consiste à déterminer si on peut couvrir, entièrement et sans chevauchements, le plan infini entier avec des polyominoes dans P. Pour former une couverture, on peut utiliser plusieurs fois le même polyomino, mais sans le faire tourner. Sur l'exemple, la réponse est encore positive, en utilisant par exemple uniquement le dernier polyomino.

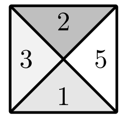
1) Le problème Tiling est indécidable. On veut montrer que Tetris est aussi indécidable. Pour cela, faut-il réduire Tiling à Tetris ou Tetris à Tiling ? Justifier votre réponse.
2) Décrire de façon intuitive
   a) une réduction de Tiling à Tetris,
   b) une réduction de Tetris à Tiling.

■

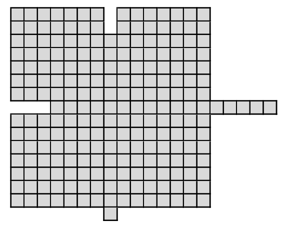
Solution.

1) Il faut réduire Tiling à Tetris, voir le cours pour la justification.

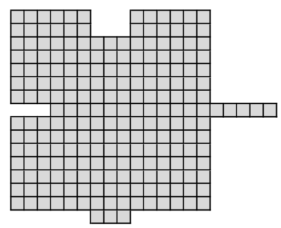
2a) Il s'agit de fabriquer un jeu de polyominos à partir d'un jeu de dominos de Wang, de telle sorte qu'on peut paver le plan avec le jeu de domino de Wang si et seulement si on peut recouvrir le plan sans chevauchement avec les polyominos construits. Une idée simple est de faire des tuiles carrées avec des encoches dont la longueur code les couleurs. Un polyomino sera donc de la forme d'un carré de taille $2n+3$ où n est le nombre de couleurs, à ceci près que le carré a 2 encoches en creux et 2 en relief. Chaque couleur $i = \{1, \ldots, n\}$ est codée par une encoche de longueur i. Les côtés Est et Sud ont des encoches en relief, destinées à s'emboîter dans les encoches en creux des côtés Ouest et Nord d'autres pièces. Par exemple, avec ce codage, le domino



est codé par

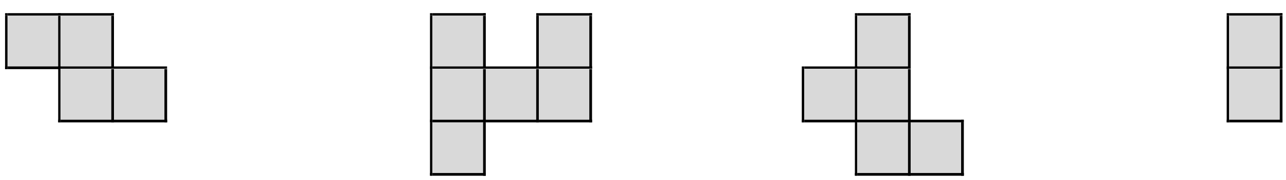


Cette réduction fonctionne pour le problème posé dans l'énoncé. On peut l'améliorer pour imposer l'orientation des polyominos, en faisant des encoches Nord-Sud d'épaisseur différente de celle des encoches Est-Ouest :



Cela montre que la variation du problème TETRIS dans laquelle on peut faire tourner les polyominos est également indécidable.

2b) Inversement, pour construire un jeu de dominos de Wang à partir d'un jeu de polyominos en assurant que soit les deux jeux peuvent paver le plan, soit aucun des deux ne le peut, on associe à chaque polyomino autant de tuiles de Wang qu'il comporte de carrés. La couleur des tuiles est choisie pour forcer les polyominos à être recomposés, avec une bordure de couleur uniforme. Par exemple, avec le jeu de polyominos suivant :



on construit le jeu de dominos suivant. Ce qui est important de remarquer est que chaque frontière entre 2 carrés élémentaires correspond à une couleur associée à la frontière de façon unique, c'est-à-dire partagée uniquement par les côtés des deux dominos de part et d'autre de la frontière. Une fois un domino placé pour paver le plan, on doit donc placer ses voisins de façon à reformer en entier le polyomino dont le domino provient. D'autre part, le contour de chacun des polyominos ainsi formés n'a pas de contrainte autre que d'être ajusté avec une bordure extérieure d'une autre reconstitution d'un polyomino.