UNIVERSITE JOSEPH FOURIER

**Master d'informatique 1ère année**

**Calculabilité et Complexité** **Année 2013-2014**

**Examen**

# Règles générales

– Durée : 3 heures. Seul document autorisé : une feuille A4 recto-verso.
– Toutes les réponses doivent être justifiées.
– Toutes les questions valent autant.
– Il est inutile et même contre-indiqué de recopier l'énoncé sur la copie d'examen.

# Énoncé

---

*Machines de Turing Non Déterministes*

---

Soit La MT Z suivante : $\begin{bmatrix} q_0 & s_1 & s_3 & q_1 \\ q_0 & s_1 & s_4 & q_1 \\ q_0 & s_2 & s_3 & q_1 \\ q_1 & s_3 & R & q_0 \\ q_1 & s_4 & s_5 & q_2 \\ q_0 & B & s_6 & q_2 \end{bmatrix}$

1. Donner explicitement l'ensemble E de ses états et l'ensemble S de ses symboles.
2. Expliquer pourquoi cette MT est non déterministe.
3. Donner explicitement ses différentes exécutions à partir de la configuration intiale $q_0s_1s_2s_2B$. Pour chacune de ces exécutions, vous préciserez chaque pas de calcul.
4. On utilise cette MT pour résoudre un problème de décision dont les instances sont codées par des mots quelconques construits à partir des symboles $s_1$ et $s_2$ et dont le résultat est soit 'OUI' (codé par le symbole $s_5$) soit 'NON' (codé par le symbole $s_6$).
   (a) En utilisant la question précédente, montrer que l'instance codée par le mot $s_1s_2s_2$ est une instance positive de ce problème, et donner le nombre de pas de calculs effectués par la MT pour décider du résultat.
   (b) Donner un exemple du codage d'une instance négative de ce problème, et préciser le nombre de pas de calculs effectués par la MT pour décider du résultat.
   (c) Rappeler la définition de la fonction de complexité d'une MT non déterministe qui résout un un problème de décision.
   (d) Donner la complexité (en ordre de grandeur) de la MT Z en fonction de la taille $n$ du mot d'entrée.
5. Dans cette question on va coder en logique les exécutions de la MT Z à partir de la configuration initiale $q_0s_1s_2$.
   On convient que :
   – les symboles sont numérotés de 0 à 6 (le symbole $B$ est le symbole $s_0$ et a donc le numéro 0),

- les cases de la bande sont numérotées de 0 à $M$ (où $M$ est une constante entière donnée) à partir de la case pointée par la tête de lecture dans la configuration initiale, qui a donc le numéro 0,
- les configurations sont numérotées de 0 à $P$ (où $P$ est une constante entière donnée) : la configuration initiale a le numéro 0 ; la configuration numéro 1 sera la configuration obtenue à partir de la configuration initiale en effectuant le pas de calcul correspondant à l'application du premier quadruplet du programme de Z, etc ...

On utilisera les variables propositionnelles suivantes :

- $S_{jct}$, qui vaut Vrai si le symbole $s_j$ est dans la case numéro $c$ lorsque Z est dans la configuration numéro $t$,
- $T_{ct}$, qui vaut Vrai si la tête de lecture pointe sur la case numéro $c$ lorsque Z est dans la configuration numéro $t$,
- $E_{it}$, qui vaut Vrai si Z est dans l'état $q_i$ quand elle est dans la configuration numéro $t$.

En utilisant ces variables propositionnelles, avec les indices adéquats :

(a) Ecrire la formule $F_{init}$ qui exprime que dans la configuration initiale, l'état est $q_0$, et que $s_1$ est le symbole contenu dans la case numéro 0, et que $s_2$ est le symbole contenu dans la case numéro 1, et que toutes les cases numérotées de 2 à $M$ contiennent le symbole $s_0$ (c'est-à-dire $B$).

(b) Ecrire la formule $F^1_{univ}$ qui exprime que dans la configuration initiale, seule la case numéro 0 est pointée par la tête de lecture (c'est-à-dire toutes les autres cases, numérotées de 1 à M, ne sont pas pointées par la tête de lecture).

(c) Ecrire la formule $F^2_{univ}$ qui exprime que dans la configuration initiale, la machine Z n'est pas dans un autre état que l'état $q_0$.

(d) Ecrire la formule $F^3_{univ}$ qui exprime que dans la configuration initiale, chaque case (numérotée de 0 à M) contient un symbole et un seul.

(e) Ecrire la formule $F_{pas}$ qui traduit l'application du premier quadruplet de Z à la configuration initiale, et qui fait passer Z dans la configuration numéro 1.

---

## *Le problème de la couverture de sommets d'un graphe est NP-complet*

---

**Préambule :** on rappelle les notions de base suivantes sur les graphes et la logique propositionnelle, suffisantes pour la bonne compréhension de ce problème.

**Graphes**

- Un graphe $G$ est un couple $(S,A)$ où $S$ est un ensemble fini de sommets et $A \subseteq S \times S$ est l'ensemble de ses arcs.
- Deux sommets $x$ et $y$ d'un graphe $G$=$(S,A)$ seront dits *adjacents* **ssi** $(x, y) \in A$ ou $(y, x) \in A$ (i.e., s'il existe un arc entre les deux sommets). L'adjacence est donc une relation symétrique.
- Une **couverture de sommets** d'un graphe $G$= $(S,A)$ est un sous-ensemble de sommets $S' \subseteq S$ tel que tout arc du graphe a au moins un de ses sommets qui appartient à $S'$ ( pour tout arc $(u, v) \in A$, on a $u \in S'$ ou $v \in S'$).
- Le **problème de couverture de graphes** est le problème de décision qui consiste à déterminer, étant donnés un graphe $G$ et un entier $j$, si il existe une couverture de sommets du graphe $G$ contenant $j$ sommets.

**Logique propositionnelle**

- Le problème 3-SAT est le problème de décision qui consiste à tester la satisfaisabilité d'une conjonction de clauses comportant chacune exactement 3 littéraux (différents deux à deux).
- Une instance de 3-SAT est donc une formule $E_1 \wedge \ldots \wedge E_i \ldots \wedge E_k$ construite à partir d'un ensemble $V = \{p_1, \ldots, p_v\}$ de variables propositionnelles, telle que chaque $E_i$ est une clause de la forme $x_{i1} \vee x_{i2} \vee x_{i3}$, où $x_{ij}$ est une variable propositionnelle $p$ de $V$ ou la négation $\neg p'$

d'une variable propositionnelle $p'$ de $V$, et telle que les 3 littéraux $x_{i1}$, $x_{i2}$, $x_{i3}$ sont différents deux à deux.

- La taille d'une clause est le nombre des littéraux qui la constituent (on considère des clauses sans répétition de littéraux). La taille d'une conjonction de clauses est la somme des tailles des clauses présentes dans la conjonction.
  La taille d'une instance $E_1 \wedge \ldots \wedge E_i \ldots \wedge E_k$ de 3-SAT comportant $k$ clauses est donc $3 \times k$.
- Une interprétation $I$ d'un ensemble $V = \{p_1, \ldots, p_v\}$ de variables propositionnelles est une fonction qui associe à chaque variable propositionnelle $p_i$ la valeur $Vrai$ ou $Faux$.
  - L'évaluation d'une conjonction de clauses dans une interprétation $I$ de ses variables propositionnelles renvoie $Vrai$ si *toutes* les clauses sont évaluées à $Vrai$ dans $I$, et $Faux$ sinon.
  - L'évaluation d'une clause dans une interprétation $I$ renvoie $Vrai$ si *au moins* un de ses littéraux est évalué à $Vrai$ dans $I$ et $Faux$ sinon.
  - Un littéral positif $p_i$ est évalué à $Vrai$ dans $I$ si $I(p_i) = Vrai$, à $Faux$ sinon.
  - Un littéral négatif $\neg p_i$ est évalué à $Faux$ dans $I$ si $I(p_i) = Vrai$, à $Vrai$ sinon.
- Une conjonction de clauses est satisfaisable s'il existe une interprétation de ses variables propositionnelles dans laquelle toutes les clauses sont évaluées à $Vrai$.
  **Exemple :** La conjonction de clauses $(\neg p_1 \vee p_2 \vee p_3) \wedge (p_1 \vee p_2 \vee \neg p_3)$ est une instance de 3-SAT de taille 6.
  Elle est évaluée à $Vrai$ dans l'interprétation $I$ définie par : $I(p_1) = I(p_2) = I(p_3) = Faux$
  Elle est évaluée à $Faux$ dans l'interprétation $I'$ : $I'(p_1) = I'(p_2) = Faux, I'(p_3) = Vrai$.
  Elle est satisfaisable ($I$ suffit pour le prouver).

- **Rappel :** 3-SAT est un problème NP-complet.

**Problème**

1. Donner un exemple de graphe à 4 sommets qui a une couverture de sommets de taille 2. Donner ensuite un exemple d'un graphe à 4 sommets pour lequel il n'existe pas de couverture de sommets de taille 2.
2. On code la relation d'adjacence associée à un graphe $G$ de $n$ sommets (que l'on identifie par leurs numéros de 1 à $n$) par un tableau T de taille $n \times n$ tel que : $T[l, c] = 1$ si les sommets de numéros $l$ et $c$ sont adjacents (cf. préambule), et $T[l, c] = 0$ sinon.
   Compléter le programme pseudo-Pascal suivant qui vérifie si un ensemble de $j$ sommets (représenté par un tableau $C$) est une couverture de sommets d'un graphe $G$ (représenté par tableau $T$ codant sa relation d'adjacence). Un ensemble de $j$ sommets est codé par un tableau $C$ de dimension 1 dont les $j$ premières cases contiennent les numéros de ces sommets (sans répétition et dans un ordre quelconque). Le programme devra renvoyer comme résultat une variable ayant la valeur 1 si $C$ représente une couverture de sommets du graphe, et 0 sinon.

```
l:= 1 ; c:=1 ; couverture := 1 ;
while l ≤ n do
   while c ≤ n do
      if T[l,c] = 1 then
         ...
         ...
write(couverture);
```

3. Donner l'ordre de grandeur de la complexité de ce programme en fonction du nombre $n$ de sommets du graphe.

4. Rappeler la définition de la classe des problèmes NP, et montrer que le problème de couverture d'un graphe est dans NP.

5. Soit f la transformation qui, à partir d'une instance $E_1 \wedge \ldots \wedge E_i \ldots \wedge E_k$ de 3-SAT, construit une instance $(G,j)$ du problème de couverture de graphe comme suit :
   – L'entier $j$ est $v + 2k$, où $v$ est le nombre de variables propositionnelles de la formule.
   – L'ensemble S des sommets du graphe $G$ est constitué de :
     - autant de paires de sommets étiquetés respectivement par $p_i$ et $\neg p_i$ qu'il y a de variables propositionnelles dans la formule,
     - autant de triplets de sommets étiquetés respectivement par $x_{i1}$, $x_{i2}$, $x_{i3}$ qu'il y a de clauses $x_{i1} \vee x_{i2} \vee x_{i3}$ dans la formule.
     Si $v$ est le nombre de variables propositionnelles de la formule, il y a donc $2v + 3k$ sommets dans le graphe.
   – L'ensemble A des arcs du graphe $G$ est constitué de :
     - l'arc $(p_i, \neg p_i)$ pour chaque paire de sommets $p_i, \neg p_i$,
     - les arcs $(x_{i1}, x_{i2})$, $(x_{i2}, x_{i3})$, $(x_{i1}, x_{i3})$ pour chaque triplet de sommets $x_{i1}$, $x_{i2}$, $x_{i3}$,
     - un arc entre chaque sommet $x_{ij}$ et le sommet $p$ ou $\neg p$ représentant le littéral correspondant.
     Le nombre d'arcs du graphe est donc : $v + 3k + 3k = v + 6k$.

   Dessiner le graphe $G_1$ obtenu à partir de la formule $F$ : $(\neg p_1 \vee p_2 \vee p_3) \wedge (p_1 \vee p_2 \vee \neg p_3)$. Donner également la valeur de l'entier $j_1$ associé, c'est-à-dire, tel que $(G_1, j_1)$ est le résultat produit par $f(F)$.

6. Dans le cas général, montrer que la transformation $f$ est de complexité polynomiale en fonction de la taille de la formule à laquelle on l'applique.

7. Montrer que si l'instance $E_1 \wedge \ldots \wedge E_i \ldots \wedge E_k$ de 3-SAT est satisfaisable, le graphe obtenu par la transformation f a une couverture de $j$ sommets (où $j = v + 2k$, $v$ étant le nombre de variables propositionnelles de la formule).
   *Indication* : construire cette couverture à partir d'une interprétation $I$ qui rend vraie chaque clause $E_i$ ; considérer l'ensemble constitué des sommets $p_i$ tels que $I(p_i) =$ vrai, des sommets $\neg p_j$ tels que $I(p_j) =$ faux, et de deux sommets bien choisis en fonction de $I$ parmi chaque groupe de trois sommets $x_{i1}$, $x_{i2}$, $x_{i3}$.

8. A partir d'une couverture de $v + 2k$ sommets du graphe obtenu par la transformation f à partir d'une instance $E_1 \wedge \ldots \wedge E_i \ldots \wedge E_k$ de 3-SAT, construire une interprétation $I$ qui rend vraie la formule.
   *Indication* : noter d'abord (en le justifiant) que la couverture doit contenir au moins un sommet parmi chaque paire $p_i$, $\neg p_i$ ; montrer ensuite que l'interprétation $I$ telle $I(p_i) =$ vrai si $p_i$ est dans la couverture, et $I(p_i)=$ faux si $\neg p_i$ est dans la couverture, rend vraies toutes les clauses $E_i$ de la formule.

9. Rappeler la définition d'une réduction polynomiale d'un problème de décision $B$ vers un problème de décision $B'$, ainsi que la définition d'un problème NP-complet.

10. Montrer comment le résultat de certaines des questions précédentes (que vous préciserez) permet de déduire que le problème de couverture de graphes est NP-complet.