Nom : .... Prénom : .... Matricule : ....

Introduction à la théorie de l'informatique

Examen écrit du vendredi 11 janvier 2013

Durée : 3 heures 1/2 maximum.

Lisez les consignes avant de commencer l'examen :

- Répondez à chaque question sur une feuille séparée, sur laquelle figurent votre nom et votre section.
- Exposez votre raisonnement de manière claire et complète si vous voulez avoir le maximum de points. Il n'est pas suffisant de donner simplement la solution finale.
- Soyez bref et concis, mais précis. Les propositions incorrectes ou hors-sujet vous feront perdre des points.
- Pour chaque démonstration, précisez la technique utilisée (absurde, induction, etc.) et pour chaque technique respectez le mise en forme préconisée dans le cours.
- Vous avez le droit d'utiliser **uniquement** les transparents du cours théorique.
- Vous ne pouvez communiquer avec personne excepté les examinateurs pendant l'examen. Toute personne surprise à tricher recevra une note de 0/20 pour l'examen et s'expose à des sanctions.
- Cet examen comporte 6 questions. Vérifiez que vous avez bien les 6 questions sur vos feuilles d'énoncé !
- Sauf indications contraires, tous les graphes sont simples (non multigraphes) et non-orientés. Ils ne contiennent donc aucune boucle (arêtes dont les deux extrémités arrivent au même sommet) et ont au plus une arête joignant les deux mêmes sommets.

Bon travail !

1. L'algorithme suivant implémente le calcul du quotient et du reste de la division entière de y par z :

```
DIVIDE(y, z)
1  r = y ;q = 0 ;w = z
2  while w ≤ y
3        w = 2 · w
4  while w > z
5        q = 2 · q ; w = ⌊w/2⌋ ;
6        if w ≤ r
7              r = r − w ;q = q + 1
8  return (q, r)
```

   (a) En considérant les lignes 1 à 3 comme une initialisation, modéliser cet algorithme par une machine d'état. Définissez précisément l'ensemble d'états, l'état initial et les transitions de la machine.

   (b) En utilisant le théorème d'invariant, montrez que l'algorithme est partiellement correct pour les précondition et postcondition suivantes :
   Pre $=$ "$y, z \in \mathbb{N}$"
   Post $=$ "$y = qz + r$ et $r < z$".
   Suggestion : utilisez comme invariant le prédicat : "$q \cdot w + r = y$ et $r < w$"

   (c) Montrez que l'algorithme se termine toujours après au plus $\log_2(y) + 1$ itérations.

2. Soit un langage d'expressions arithmétiques avec variables défini par la syntaxe BNF suivante :

$$E \in Exp := \mathtt{n}|\mathtt{x}|(E + E)|(E \times E)$$

   (a) Définissez les deux fonctions suivantes récursivement sur base de la syntaxe des expressions :

      i. `simplify` : $Exp \to Exp$ prenant en argument une expression et renvoyant une expression simplifiée où une somme de deux termes identiques est remplacée par 2 fois ce terme. Par exemple,

$$\begin{aligned}&\mathtt{simplify}((1 + (((3 \times x) + (y + y)) + ((3 \times x) + (y + y)))))\\ &= (1 + (2 \times ((3 \times x) + (2 \times y))))\end{aligned}$$

      ii. `nvarop` : $Exp \to \mathbb{N}$ prenant en argument une expression et renvoyant le nombre de variables et d'opérateurs $+$ ou $\times$ dans l'expression. Par exemple :

$$\mathtt{nvarop}((1 + (((3 \times x) + (y + y)) + ((3 \times x) + (y + y))))) = 14.$$

   (b) Montrez que pour toute expression $E \in Exp$, on a

$$\mathtt{nvarop}(\mathtt{simplify}(E)) \leq \mathtt{nvarop}(E)$$

   (`simplify` simplifie bien les expressions).
   Il n'est pas nécessaire de traiter la multiplication.

(c) Soit la sémantique opérationnelle suivante pour Exp :

$$\frac{}{\langle x, s\rangle \to \langle \mathtt{n}, s\rangle}\ \texttt{(W-EXP.NUM)} \qquad \frac{\langle E_1, s\rangle \to \langle E_1', s'\rangle}{\langle (E_1 + E_2), s\rangle \to \langle (E_1' + E_2, s'\rangle}\ \texttt{(W-EXP.LEFT)}$$

$$\frac{}{\langle (\mathtt{n}_1 + \mathtt{n}_2), s\rangle \to \langle \mathtt{n}_3, s\rangle}\ \texttt{(W-EXP.ADD)} \qquad \frac{\langle E_2, s\rangle \to \langle E_2', s'\rangle}{\langle (\mathtt{n} + E_2), s\rangle \to \langle (\mathtt{n} + E_2', s'\rangle}\ \texttt{(W-EXP.RIGHT)}$$

Montrez que pour tout $E \in Exp$,

$$(\mathtt{nvarop}(E) = k) \Rightarrow \exists \mathtt{n} : \langle E, s\rangle \to^k \langle \mathtt{n}, s'\rangle.$$

(`nvarop` mesure le nombre de transitions nécessaires à l'évaluation d'une expression)
On supposera que toutes les variables sont initialisées à 0 en mémoire. La multiplication ne doit pas être traitée.

3. Soit un graphe $G = (V, E)$. Un ensemble de sommets $A \subseteq V$ est *indépendant* s'il n'y a pas d'arêtes dans G entre les sommets de A. Soit A et $B \subseteq V$ deux ensembles indépendants de taille maximale. Montrez qu'il existe une correspondance pour les sommets de $A \setminus B$ dans le graphe biparti défini par les sommets de $A \setminus B$ et de $B \setminus A$.

4. Pour chacune des paires de fonctions de n suivantes, précisez toutes les notations parmi celles ci-dessous qui permettent de comparer la première fonction à la seconde :

$$=, O, \Omega, \Theta, o, w, \sim$$

$$\begin{array}{rcl}
\ln(n^3) & ? & \ln(n) \\
n^{\log_2(3)} & ? & 4^{\log_2(n)} \\
T(1) = 2, T(n) = 2T(n/4) + n \text{ pour } n > 1 & ? & n \\
[x^n]\frac{x}{2(1-\frac{x}{2})^2(1-x)} & ? & n
\end{array}$$

Justifiez vos réponses.

5. Soit la récurrence

$$\begin{array}{rcl}
u_0 & = & 1, \\
u_n & = & \displaystyle\sum_{k=0}^{n-1} 2^k k u_{n-1-k}.
\end{array}$$

Calculez une solution analytique pour $U(z) = \sum_{n=0}^{+\infty} u_n z^n$.
Suggestion : servez vous d'une solution analytique pour la série $\sum_{n=0}^{+\infty} n 2^n z^n$.

6. Un programme informatique considère un chaîne constituée de chiffres décimaux comme un mot de passe valide si elle contient un nombre pair de chiffres 0. Par exemple "1230407869" et "12345" sont des mots de passe valides mais "120987045608" et "012879" ne le sont pas. On considérera que la chaîne vide est un mot de passe valide (de longueur 0).
   (a) Exprimez l'ensemble de mots de passe valides sur base de sous-ensembles élémentaires de caractères à l'aide des opérateurs $+$, $\times$ et seq.
   (b) En déduire directement une solution analytique pour la fonction génératrice correspondant au choix des mots de passe.
   (c) Montrez à partir de cette fonction que le nombre de mots de passe de longueur n vaut $\frac{8^n+10^n}{2}$.
   Suggestion : vous pouvez partir de la solution.
   (d) Comment se modifie la fonction génératrice si on impose en plus que tout chiffre 2 soit toujours suivi d'un autre chiffre 2 ? Il n'est pas demandé de déterminer le nombre de mots de passe de longueur n dans ce nouveau cas.