

## Examen - 17 mai 2017

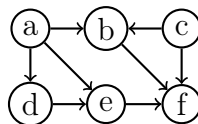
L'examen dure 2 heures. Toutes les réponses devront être clairement justifiées. Les seuls documents autorisés sont une page A4 manuscrite recto-verso.

**Exercice 1** *Graphes, ordre topologique (6 points)*

On se donne un graphe orienté sur un ensemble fini de sommets  $A$ . On cherche à ordonner les sommets de  $A$  de manière totale et compatible avec la relation associée au graphe. C'est-à-dire que l'on veut construire une bijection  $\sigma$  de  $A$  dans  $[1..n]$  (ce qui revient à numéroter les sommets du graphe) telle que s'il y a une arête de  $a$  vers  $b$  dans le graphe alors  $\sigma(a) < \sigma(b)$ . On dira dans la suite que  $\sigma$  est une **numérotation compatible** avec la relation du graphe.

Si on voit les sommets comme des tâches à réaliser (par exemple compiler un fichier) et si une arête entre  $a$  et  $b$  dans le graphe représente le fait que la tâche  $a$  doit être réalisée avant la tâche  $b$ , alors  $\sigma$  va nous donner une manière de réaliser les tâches l'une après l'autre sans risque de blocage.

1. Soit le graphe

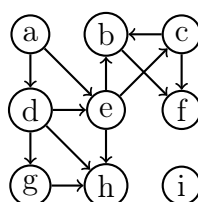


Proposer deux numérotations distinctes des sommets, compatibles avec la relation du graphe.

2. Montrer que si un graphe  $G$  contient un cycle alors il n'existe pas de numérotation  $\sigma$  compatible avec la relation du graphe.
3. Montrer que si un graphe a un nombre fini de sommets et que tous ces sommets ont un degré sortant strictement positif alors ce graphe a forcément un cycle. Donner un exemple de graphe infini pour lequel cette propriété est fausse.

**Rappel :** Le degré sortant du sommet  $x$  est le nombre d'arêtes du graphe qui ont  $x$  pour origine.

4. Soit un graphe  $G$  et  $a$  un sommet de degré sortant 0. Soit  $G'$  le graphe obtenu à partir de  $G$  en retirant le sommet  $a$  et toutes les arêtes d'extrémité  $a$ . Montrer que s'il existe une numérotation compatible pour  $G'$  alors on peut aussi construire une numérotation compatible pour  $G$ .
5. Déduire des questions précédentes une preuve par récurrence sur le nombre de sommets du fait que si  $G$  est un graphe orienté avec  $n$  sommets et sans cycle alors il existe une numérotation compatible.
6. La preuve précédente donne également un algorithme pour construire une telle numérotation. Appliquer cet algorithme au graphe suivant, donner la numérotation trouvée :



**Exercice 2** *Ensembles d'entiers, mots binaires (8 points)*

Dans cet exercice, on s'intéresse à des ensembles d'entiers compris entre 1 et  $n$ . On note  $I_n$  l'ensemble des entiers compris entre 1 et  $n$  (par convention  $I_0$  est l'ensemble vide) et  $P_n = \wp(I_n)$  l'ensemble des sous-ensembles de  $I_n$ .

On représente un ensemble d'entiers, élément de  $P_n$ , par un mot de longueur  $n$  ne contenant que des 0 ou des 1, appelé un *mot binaire*. On note  $M_n$  l'ensemble des mots binaires de longueur  $n$ .

— Le mot vide  $\epsilon$  représente l'ensemble vide, qui est le seul élément de  $P_0$ .

— Le mot  $a_n a_{n-1} \dots a_1$  représente l'ensemble des entiers  $i$  pour lesquels  $a_i = 1$ .

Si on prend  $n = 3$ , alors 000 représente l'ensemble vide, 100 représente le singleton  $\{3\}$ , 001 représente le singleton  $\{1\}$  et 101 représente l'ensemble  $\{1; 3\}$ .

Les constructions sur les mots binaires de longueur  $n$  peuvent se faire de manière récursive. En effet si  $n > 0$ , le mot  $m = a_n a_{n-1} \dots a_1$  de  $M_n$  se décompose en  $a_n$  et en un mot  $m' = a_{n-1} \dots a_1 \in M_{n-1}$ . Le mot  $m'$  représente un ensemble  $X$  d'entiers entre 1 et  $n - 1$ , le mot  $m$  représente le même ensemble vu comme un élément de  $P_n$  si  $a_n = 0$  et l'ensemble  $\{n\} \cup X$  si  $a_n = 1$ .

1. Pour  $n = 6$ , dire à quel ensemble correspond le mot 101011 et donner la représentation par un mot binaire de l'ensemble  $\{1; 4\}$ .
2. Définir de manière récursive une fonction  $\text{val}_n$  de  $M_n \rightarrow P_n$  qui à un mot binaire associe l'ensemble qu'il représente.

(Indication : On pourra définir  $\text{val}_0(\epsilon)$ , et  $\text{val}_{n+1}(0m)$  ainsi que  $\text{val}_{n+1}(1m)$  en fonction de  $n$  et de  $\text{val}_n(m)$ )

3. Définir de manière récursive sur  $n$  une fonction  $\text{inter}$  qui à deux mots binaires  $m_1$  et  $m_2$  de même longueur associe un mot binaire qui représente l'intersection des ensembles représentés par  $m_1$  et  $m_2$ . C'est-à-dire que l'on doit avoir  $\text{val}_n(\text{inter}(m_1, m_2)) = \text{val}_n(m_1) \cap \text{val}_n(m_2)$ .
4. On note  $0_n$  le mot de longueur  $n$  formé uniquement de 0, le mot  $0_0$  est vide et donc correspond à  $\epsilon$ .

Montrer par récurrence sur  $n$  que pour tout  $m \in M_n$  on a  $\text{inter}(m, 0_n) = 0_n$ .

5. On définit par des règles d'inférence ci-dessous une relation  $\text{incl}(m_1, m_2)$  sur les mots de même longueur qui représente le fait que l'ensemble représenté par  $m_1$  est inclus dans l'ensemble représenté par  $m_2$  :

$$\frac{}{\text{incl}(\epsilon, \epsilon)}(1) \quad \frac{\text{incl}(m_1, m_2)}{\text{incl}(am_1, am_2)}(2) \quad \frac{\text{incl}(m_1, m_2)}{\text{incl}(0m_1, 1m_2)}(3)$$

- (a) Construire une preuve de  $\text{incl}(0011, 1011)$  en indiquant à chaque étape quelle règle d'inférence est utilisée.
- (b) Montrer par récurrence sur le mot  $m$  que pour tout  $m \in M_n$  on a  $\text{incl}(m, m)$ .
- (c) Énoncer le principe d'induction associé à la relation  $\text{incl}$ .
- (d) En utilisant ce principe, montrer que pour tout  $m_1$  et  $m_2$  dans  $M_n$ , on a  $\text{incl}(m_1, m_2) \Rightarrow \text{inter}(m_1, m_2) = m_1$ .

**Exercice 3** *Tas binaire (7 points)*

On se donne un ensemble  $E$  et un ordre total  $\leq$  sur les éléments de  $E$ . On note  $\overline{E}$  l'ensemble  $E \cup \infty$ . L'ordre  $\leq$  est étendu à  $\overline{E}$  en fixant que pour tout  $x$  dans  $E$ ,  $x \leq \infty$ .

Un tas binaire est une structure de données ayant la forme d'un arbre binaire et qui permet de stocker une collection finie d'objets de  $E$ . Les tas binaires offrent des algorithmes efficaces pour les trois actions suivantes :

- L'ajout d'un élément
- La fusion de deux tas
- L'extraction du minimum

Nous représenterons les arbres comme des termes sur la signature contenant une constante  $F$  et un symbole  $N$  d'arité trois. Un terme  $F$  représente une feuille, tandis qu'un terme  $N(a_g, x, a_d)$  représente un nœud portant un élément  $x$  et ayant respectivement pour sous-arbres gauche et droit  $a_g$  et  $a_d$ .

La fonction `racine` définie ci-dessous renvoie la racine de l'arbre passé en paramètre, ou  $\infty$  dans le cas d'une feuille.

$$\text{racine}(F) = \infty \qquad \text{racine}(N(a_g, x, a_d)) = x$$

Les règles d'inférence suivantes définissent la relation sur les arbres "être un tas binaire".

$$\frac{}{\text{tas}(F)} \qquad \frac{\text{tas}(a_g) \quad x \leq \text{racine}(a_g) \quad x \leq \text{racine}(a_d) \quad \text{tas}(a_d)}{\text{tas}(N(a_g, x, a_d))}$$

1. Montrer que, pour tout élément  $x$ , l'arbre singleton  $N(F, x, F)$  est un tas binaire.
2. Dessiner les arbres suivants, et indiquer lesquels sont des tas binaires (on pourra se passer de dessiner les feuilles).
  - A.  $N(N(N(F, 5, F), 3, N(F, 9, F)), 1, N(F, 6, N(F, 8, F)))$
  - B.  $N(N(N(F, 1, F), 3, N(F, 5, F)), 6, N(F, 8, N(F, 9, F)))$
  - C.  $N(N(F, 9, F), 1, N(N(F, 5, F), 3, N(F, 6, N(F, 8, F))))$
3. Définir par des règles d'inférence une relation `estDans` telle que `estDans(x, a)` est vraie si l'élément  $x$  apparaît dans l'arbre  $a$ .
4. Énoncer le principe d'induction sur les tas binaires.
5. Rappeler la définition du minimum d'un ensemble, et démontrer que dans tout tas binaire  $a$ , `racine(a)` est le minimum de l'ensemble des éléments apparaissant dans  $a$ .

On définit par des équations récursives sur la structure des arbres une fonction d'union :

$$\begin{aligned} \text{union}(F, F) &= F \\ \text{union}(N(a_g, x, a_d), F) &= N(a_g, x, a_d) \\ \text{union}(F, N(a_g, x, a_d)) &= N(a_g, x, a_d) \\ \text{union}(N(a_g, x, a_d), N(b_g, y, b_d)) &= \begin{cases} N(a_d, x, \text{union}(a_g, N(b_g, y, b_d))) & \text{si } x \leq y \\ N(b_d, y, \text{union}(b_g, N(a_g, x, a_d))) & \text{sinon} \end{cases} \end{aligned}$$

L'ajout d'un élément  $x$  à un tas binaire est alors défini par :

$$\text{ajoute}(x, a) = \text{union}(a, N(F, x, F))$$

6. Dessiner les arbres `ajoute(4, A)` et `ajoute(7, ajoute(4, A))`, où  $A$  est tel que défini à la question 2.
7. Définir un ordre  $\prec$  sur les paires d'arbres, qui est bien fondé et qui vérifie de plus les contraintes suivantes, pour tous  $a_g, x, a_d, b_g, y, b_d$  :
  - $(a_g, N(b_g, y, b_d)) \prec (N(a_g, x, a_d), N(b_g, y, b_d))$
  - $(b_g, N(a_g, x, a_d)) \prec (N(a_g, x, a_d), N(b_g, y, b_d))$
 En déduire que l'application récursive des équations définissant `union` finit nécessairement par s'arrêter.
8. Démontrer grâce au principe d'induction bien fondée que, si  $a$  et  $b$  sont deux tas binaires, alors `union(a, b)` est un tas binaire.