Utbm mt42 Examen final Automne 2008

Nb : Les exercices seront rédigés sur des feuilles séparées.

Exercice 1 *Fonctions booléennes*

Nb: On change de feuille !

B désigne, comme d'ordinaire, l'ensemble $\{0,1\}$ doté de sa structure naturelle d'algèbre de Boole $(B,+,.)$.

1.1 *Etude d'une fonction booléenne*

On considère la fonction f de trois variables booléennes définie par :

$$\begin{array}{rccl} f: & B^3 & \to & B \\ & (x,y,z) & \mapsto & f(x,y,z) = \bar{x}\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z} \end{array}$$

a) Fournir une représentation dans l'espace qui fasse apparaître les points couverts par f.

b) Déterminer géométriquement l'ensemble des monômes premiers de f. Démontrer que ces monômes sont premiers.

Nb : On notera $P(f)$ l'ensemble des monômes premiers de la fonction f.

c) En déduire une écriture optimale de f.

1.2 *Etude du complément de f*

On définit le complément de la fonction f par :

$$\forall (x,y,z) \in B^3 \quad g(x,y,z) = 1 - f(x,y,z).$$

a) Ecrire $g(x,y,z)$ comme somme de monômes de degré 3.

b) Déterminer géométriquement l'ensemble $P(g)$ des monômes premiers de la fonction g.

c) *Question ouverte:*

Etudier la possibilité de déterminer directement $P(g)$ à partir de $P(f)$.

Exercice 2 *Représentation matricielle de relations binaires*

Nb: On change de feuille !

On définit l'ensemble E_k de cardinal $k = 4$ par :

$$E = \{e_1, e_2, e_3, e_4\}.$$

2.1 On définit sur E_k la relation binaire r par son écriture ensembliste :

$$r = \{(e_1,e_4), (e_2,e_3), (e_3,e_3), (e_3,e_2), (e_4,e_1)\}.$$

Déterminer la matrice R de $\mathcal{M}_4(\{0,1\})$ associée à r; R est carrée de côté 4, à coefficients dans l'algèbre de Boole $\{0,1\}$.

2.2 *Etude des puissances de r*

a) Déterminer la relation $r^2 = r \circ r$ par sa matrice associée. En déduire son écriture ensembliste.

b) Déterminer la relation $r^3 = r^2 \circ r$. En déduire son écriture ensembliste.

c) Déterminer la relation $r^4 = r^3 \circ r$. En déduire son écriture ensembliste.

d) Proposer une propriété $P(n)$ qui fournit la valeur de r^n pour tout n de $\mathbb{N}$. Démontrer $P(n)$ par récurrence.

Exercice 3 *Etude de complexité*

Nb: On change de feuille !

On suppose avoir étudié un algorithme $algo(n)$ qui travaille sur des données de taille n, pour n élément de $\mathbb{N}^*$. On admet avoir démontré que pour tout n de $\mathbb{N}^*$, le temps d'exécution $T(n)$ de $algo(n)$ vérifie :

$$T(n+1) = 2T(n) + b \text{ avec } T(1) = a,$$

où a et b sont des constantes réelles positives connues. On se propose de donner une écriture explicite de $T(n)$ pour étudier la complexité de $algo(n)$.

3.1 *Développement d'outils algébriques*

Soit une suite quelconque (u_n) vérifiant la relation (L) donnée par :

$$(L) \qquad \forall n \in \mathbb{N}^* \quad u_{n+1} = 2u_n + b.$$

On cherche à caractériser (u_n).

a) On considère d'abord l'ensemble de toutes les suites (v_n) qui vérifient la relation (H) :

$$(H) \qquad \forall n \in \mathbb{N}^* \quad v_{n+1} = 2v_n.$$

Montrer que toute suite (v_n) vérifiant (H) est définie par :

$$\forall n \in \mathbb{N}^* \quad v_n = v_1 2^{n-1}.$$

b) Montrer qu'il existe une suite (w_n) constante, c'est-à-dire définie par : $\forall n \in \mathbb{N}^* \quad w_n = \alpha$ $(\alpha \in \mathbb{R})$, vérifiant la relation (L).

Nb : On donnera α en fonction de b.

c) Démontrer que toute suite (u_n) vérifiant la relation (L) est la somme d'une suite (v_n) vérifiant (H) et de la suite (w_n) particulière solution de la relation (L) trouvée au b).

d) En déduire que toute suite (u_n) vérifiant la relation (L) est caractérisée par :

$$\forall n \in \mathbb{N}^* \quad u_n = v_1 2^{n-1} - b,$$

où v_1 est un réel quelconque.

3.2 *Utilisation pour la détermination de la suite* $(T(n))$

a) Montrer que la suite $(T(n))$ vérifie la relation (L). En déduire que :

$$\forall n \in \mathbb{N}^* \quad T(n) = v_1 2^{n-1} - b,$$

où b est connu et v_1 à déterminer.

b) Montrer que v_1 peut être déterminée par la connaissance de $T(1) = a$. Exprimer $v_1$en fonction de a et de b.

c) Donner l'expression de $T(n)$ pour $n \geq 1$, en fonction de a, b et n.

d) *Application numérique*

Les temps sont exprimés en secondes. On suppose que $T(1) = 10^{-3}$ et que $T(2) = 5 * 10^{-3}$.

Donner une valeur approchée de $T(n)$ pour n élément de $\{10, 100, 1000\}$. Que pensez-vous de la complexité de $algo(n)$?