Utbm mt42 **Examen final** **Automne 2011**

Nb : Les exercices seront absolument rédigés sur des feuilles séparées.

Exercice 1 *Logique*

Nb : On change de feuille !

On rappelle la définition de l'équivalence logique.

Définition 1

Deux formules logiques F_1 et F_2 sont dites équivalentes, et on note $F_1 \equiv F_2$, si $F_1 \models F_2$ et $F_2 \models F_1$, où $\models$ dénote la conséquence logique.

1.1 Parmi les équivalences logiques suivantes, laquelle est vraie, laquelle est fausse ? Justifier.

- $p \wedge q \to r \equiv p \to (q \to r)$
- $(p \wedge \neg q) \to r \equiv p \vee q \vee r$

1.2 Montrez que si $F_1 \equiv F_2$, alors F_1 est valide si et seulement si F_2 est valide.

1.3 *Etude des formes normales conjonctives (FNC)*

On fournit la définition de monôme disjonctif comme suit.

Définition 2

Un monôme disjonctif est une disjonction de littéraux, un littéral étant soit une variable (x), soit une variable complémentée $(\neg x)$. Par exemple, $p \vee \neg q \vee r$ et $\neg p \vee q$ sont des monômes disjonctifs.

Une formule logique F est dite en forme normale conjonctive (FNC) si elle s'écrit comme une conjonction de monômes disjonctifs. Par exemple, la formule :

$$(\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee r)$$

est en FNC.

N.B : L'intérêt de cette forme d'écriture est qu'il est très facile de savoir si une formule en FNC est valide, en examinant ses monômes disjonctifs.

a) En justifiant votre réponse, donner une condition nécessaire et suffisante pour qu'un monôme disjonctif soit valide.

b) En justifiant votre réponse, en déduire une condition nécessaire et suffisante pour qu'une formule en FNC soit valide.

c) *Décision de validité*

c1) *Problématique*

Vu qu'il est très facile de décider si une formule en FNC est valide, nous pouvons concevoir une méthode de décision de la validité, pour une formule F_1 quelconque :

- Trouver une formule F_2 en FNC, telle que $F_1 \equiv F_2$.
- Décider de la validité de F_2.
  D'après la question 1.2 nous savons que cela décide de la validité de F_1.
  Le problème est de pouvoir trouver, pour toute F_1, une formule F_2 en FNC, telle que $F_1 \equiv F_2$.

c2) *Question*

Montrer que pour toute formule F_1 il existe au moins une formule F_2 en FNC telle que $F_1 \equiv F_2$

Suggestion : On pourra raisonner à partir de la table de vérité.

.../...

Exercice 2 *Fonctions booléennes*

Nb : On change de feuille !

Soit $\mathbb{B} = \{0, 1\}$, $\mathbb{F}_3$ l'ensemble des fonctions booléennes de $\mathbb{B}^3$ dans $\mathbb{B}$; on note $\overline{x}$ le complémentaire de x. Soit f la fonction booléenne de $\mathbb{F}_3$ définie par:

$$f\,[(x, y, z)] = \overline{y}\ z + \overline{x}\ y\ \overline{z} + \overline{y}\ \overline{z}$$

2.1 Représenter les monômes de la fonction f.

2.2 Donner la liste de tous les monômes inférieurs à f et les classer selon leurs degrés.

2.3 Déterminer géométriquement la liste des monômes premiers de f; rappeler la règle utilisée.

2.4 Démontrer que les monômes identifiés comme premiers dans 2.3, le sont effectivement.

2.5 Montrer que f est la somme de ses monômes premiers. Cette écriture est-elle optimale au sens de la simplification de l'écriture ?

Exercice 3 *Complexité*

Nb : On change de feuille ! Tout résultat intermédiaire pourra être admis afin de poursuivre la résolution de l'exercice.

3.1 *Cadre de l'étude*

Soit n un élément de $\mathbb{N}^*$. On considère un algorithme $algo(n \to res)$ qui à partir de données de taille n fournit le résultat res. On suppose que l'analyse de sa complexité a permis d'établir que le temps d'exécution $T(n)$ de $algo(\ )$ vérifie la relation suivante :

$$\forall n \in \mathbb{N}^* \quad T(n+1) = 2T(n) + 1 \text{ et } T(1) \text{ donné.}$$

Pour la suite de l'exercice on notera, sauf indication contraire, $T(n) = u_n$, en vue de simplifier les écritures.

A titre d'exemples, ce type de situation se rencontrera dans l'algorithme des tours de Hanoï ou dans l'algorithme de recherche des longueurs maximales d'un parcours préfixe dans un arbre binaire de profondeur inférieure ou égale à n.

3.2 *Recherche d'une écriture explicite de* (u_n) vérifiant l'équation : $(L) : \forall n \in \mathbb{N}^* \quad u_{n+1} - 2u_n = 1$

a) On suppose que la suite (u_n) vérifie l'équation : (L).

- Quels seront les calculs nécessaires pour déterminer par exemple u_{1000}?
- En déduire qu'il est crucial de fournir une écriture explicite de u_n en fonction de n.

b) Montrer qu'il existe une suite constante (w_n) vérifiant l'équation (L). Fournir la valeur de cette constante.

c) On considère désormais une suite quelconque (u_n) vérifiant l'équation (L).

- Montrer que la suite $(v_n) = (u_n) - (w_n)$ définie pour tout n de $\mathbb{N}^*$par: $v_n = u_n - w_n$, est solution de l'équation (H) donnée par:

$$(H) : \forall n \in \mathbb{N}^* \quad v_{n+1} - 2v_n = 0$$

- En déduire que pour connaitre l'ensemble des solutions (u_n) de $(L) : \forall n \in \mathbb{N}^* \quad u_{n+1} - 2u_n = 1$, il suffit de connaître l'ensemble des solutions de (H).

d) *Détermination de l'ensemble des suites solutions de* (H)

Démontrer que l'ensemble des suites (v_n) solutions de (H) est l'ensemble des suites géométriques de raison $q = 2$.

e) En déduire que le temps d'exécution $T(n)$ de l'algorithme $algo(\ )$ est donné par:

$$T(n) = a2^{n-1} - 1.$$

Fournir a en fonction de $T(1)$.