Représentation des fonctions booléennes

Épreuve pratique d'algorithmique et de programmation

Juillet 2003

Ce problème est consacré à l'étude de deux représentations des fonctions booléennes de N variables (booléennes) : les tables de vérité et les diagrammes de décisions binaires.

Soit $\mathbb{B} = \{0, 1\}$ l'ensemble des booléens. L'ensemble $\mathbb{B}$ est munis de l'opération unaire $\neg$ ($\neg 0 = 1$, $\neg 1 = 0$) et de l'opération binaire $\vee$ ($0 \vee 0 = 0$, $1 \vee 0 = 1$, $0 \vee 1 = 1$, $1 \vee 1 = 0$). Les opérations $\wedge$, $\Rightarrow$, $\Leftrightarrow$ et ite (if-then-else) sont définies à partir des opérations sur $\mathbb{B}$ de la manière suivante : $x \wedge y = \neg(\neg x \vee \neg y)$, $x \Rightarrow y = \neg x \vee y$, $x \Leftrightarrow y = (x \Rightarrow y) \wedge (y \Rightarrow x)$ et $\mathrm{ite}(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$.

Nous appelons fonctions booléennes à N variables toutes applications de $\mathbb{B}^N$ dans $\mathbb{B}$. Les opérations sur $\mathbb{B}$ s'étendent naturellement à des opérations sur les fonctions booléennes : $(\neg f)(x_1, \cdots x_N) = \neg f(x_1, \cdots x_N)$ et $(f \vee g)(x_1, \cdots x_N) = f(x_1, \cdots x_N) \vee g(x_1, \cdots x_N)$. On peut montrer que toute fonction booléenne peut être construite à partir des fonctions de projections π_i ($\pi_i(x_1, \cdots x_N) = x_i$) et des opérations $\neg$, $\vee$. Par abus de langage, nous identifierons les fonctions π_i par les variables booléennes x_i.

Soit f une fonction booléenne, soit x_i une variable. On appelle cofacteur gauche et cofacteur droit de la fonction f pour la variable x_i, les fonctions booléennes $f^-_{x_i}$, $f^+_{x_i}$: $f^-_{x_i}(x_1, \cdots, x_i, \cdots x_N) = f(x_1, \cdots, 0, \cdots x_N)$ et $f^+_{x_i}(x_1, \cdots, x_i, \cdots x_N) = f(x_1, \cdots, 1, \cdots x_N)$. Noter que les valeurs d'un cofacteur ne dépendent plus de la variable x_i. La fonction f et ces cofacteurs pour une variable x_i sont liées par l'identité : $f = \mathrm{ite}(x_i, f^+_{x_i}, f^-_{x_i})$.

Nous introduisons trois nouvelles opérations sur les fonctions booléennes :
- L'opération de substitution d'une variable x_i : $f(x_i/g) = \mathrm{ite}(g, f^+_{x_i}, f^-_{x_i})$,
- La quantification universelle : $(\forall x_i : f) = f^+_{x_i} \wedge f^-_{x_i}$,
- La quantification existentielle $(\forall x_i : f) = f^+_{x_i} \vee f^-_{x_i}$.

Le préambule élabore les données qui seront appliquées aux algorithmes du problème. Il est cependant très fortement conseillé de tester au préalable vos algorithmes sur des données plus petites (e. g. "des fonctions à 2, 3 variables"). Les parties 1, 2 et 3 sont dépendantes et doivent être réalisées dans l'ordre.

Préambule

Soit N un entier positif. Soit (u_i) la suite récurrente définit par $u_{i+1} = (101 \cdot u_i + 463) \bmod (9973)$ et u_0 est égal au numéro inscrit sur votre table d'examen. On définit le tableau de booléens F de taille 2^N par $F[i] = 1$ si $u_i < 2720$ sinon $F[i] = 0$ pour $0 \leq i < 2^N$. On définit de manière analogue les tableaux G et H en prenant comme valeur initiale x_0 "votre numéro plus 50" et "votre numéro plus 100".

Question 1 *Donner le nombre de valeurs* 1 *contenues dans les tableaux F, G, H pour $N = 6$.*

1 Tables de vérité

Soit L un tableau de taille de 2^N. Pour tout entier positif u strictement inférieur à 2^N, notons $u_N \cdots u_1$ son codage binaire où u_1 est le bit de poids faible (ou de parité) de l'entier u. Le tableau L identifie la fonction booléenne l par $l(u_1, \cdots, u_N) = L[u]$. Par exemple, un tableau de taille 4 définit une fonction des deux variables x_1 et x_2 : le tableau $[0, 1, 0, 1]$ représente la fonction x_1, $[0, 0, 1, 1]$ représente x_2 et $[0, 1, 1, 1]$ représente $x_1 \vee x_2$.

Dans cette partie, nous posons $N = 6$, les tableaux F, G, H du préambule sont considérés comme des fonctions booléennes à N variables (6 variables). Nous nous proposons de coder les opérations sur les fonctions booléennes et de les appliquer à des expressions construites à partir des fonctions F, G et H.

Question 2 *Donner une description concise des algorithmes de calcul des opérations "$\neg$", "$\wedge$", "ite" . Calculer le nombre de valeurs* 1 *des fonctions $\neg F$, $\neg G$, $\neg H$, $F \wedge G$, $ite(F, G, H)$. Que remarquez-vous ?*

Question 3 *Donner une description concise des algorithmes de calcul des cofacteurs d'une fonction par rapport à une variable x. Calculer le nombre de valeurs* 1 *des fonctions F_x^-, F_x^+ pour $x = x_1, \cdots x_N$.*

Question 4 *Donner une description concise des algorithmes de calcul de l'opération de substitution et de l'évaluation des quantificateurs "$\exists$", "$\forall$". Calculer le nombre de valeurs* 1 *des fonctions $F(x/G)$, $G(x/F)$, $\exists x(F)$, $\neg\forall x(\neg F)$ pour $x = x_1, \cdots x_N$. Que remarquez-vous ?*

2 Diagramme de décisions binaires

Dans cette partie, nous nous proposons de coder les opérations sur les fonctions booléennes sur une autre structure, appelée "diagramme de décisions binaires". Nous appliquerons nos algoritmes aux même expressions que la partie 1 (e.g. $N = 6$).

Soit MAX $= 10000$. Soit UniqueTable un tableau de triplets d'entiers positifs de taille MAX. Une entrée d du tableau est dite non définie si UniqueTable$[d] = (0, 0, 0)$. Quand d est une entrée définie, nous utiliserons les notations var(d), co$^+(d)$ et co$^-(d)$ pour référencer les trois champs du triplet UniqueTable$[d]$ (e.g. UniqueTable$[d]$ $= ($var$(d),$ co$^+(d),$ co$^-(d)))$, Imposons que les entrées 0 et 1 soient toujours définies par UniqueTable$[0] = (0, 0, 1)$ et UniqueTable$[1] = (0, 1, 0)$. De plus, tout entrée définie supérieur à 2 doit vérifiée les règles de cohérence suivante :

- co$^+(d)$ et co$^-(d)$ sont des entrées définies,
- co$^+(d) \neq$ co$^-(d)$,
- var$(d) >$ var(co$^+(d)$) et var$(d) >$ var(co$^-(d)$)
- La valeur UniqueTable$[d]$ de l'entrée d est unique (e. g. UniqueTable$[d]$ = UniqueTable$[d'] \Rightarrow d = d'$).

Nous définissons 3 opérations assurant la bonne gestion de la table UniqueTable

- Une opération d'initialisation : UniqueTable[0] $= (0, 0, 1)$, UniqueTable[1] $= (0, 1, 0)$ et UniqueTable[d] $= (0, 0, 0)$ pour $d \geq 2$,
- Une opération renvoyant le nombre d'entrées définies,
- Une opération d'ajout d'un triplet (x, u, v) retournant l'entrée. Cette opération doit respecter les règles suivantes :
  - si $u = v$, ne pas modifier la table UniqueTable et retourner la valeur u,
  - si il existe un indice d tel que UniqueTable[d] $= (x, u, v)$, alors ne pas modifier la table et retourner d,
  - sinon, trouver un indice d non initialisé, réaliser l'affectation UniqueTable[d] $= (x, u, v)$ et retourner d.

Question 5 *Programmer les 3 opérations de gestion de la table UniqueTable. Donner une description concise des structures de données et des algorithmes.*

Toute entrée définie d représente une fonction booléenne notée $\mathrm{bdd}(d)$. Formellement, $\mathrm{bdd}(d)$ est définie inductivement par :
- $\mathrm{bdd}(0)$ et $\mathrm{bdd}(1)$ sont les fonctions constantes 0 et 1,
- si $d \geq 2$ alors $\mathrm{bdd}(d) = \mathrm{ite}(x_{\mathrm{var}(d)}, \mathrm{bdd}(\mathrm{co}^+(d)), \mathrm{bdd}(\mathrm{co}^-(d)))$.

Noter que $\mathrm{co}^+(d)$ et $\mathrm{co}^-(d)$ donnent les entrées représentant les cofacteurs de la fontion booléenne $\mathrm{bdd}(d)$ pour la variable d'indice $\mathrm{var}(d)$. Par exemple, supposons que UniqueTable[2] $= (1, 1, 0)$, UniqueTable[3] $= (2, 2, 0)$, UniqueTable[4] $= (2, 1, 2)$, UniqueTable[5] $= (1, 0, 1)$, alors $\mathrm{bdd}(2) = x_1$, $\mathrm{bdd}(3) = x_1 \wedge x_2$, $\mathrm{bdd}(4) = x_2 \vee x_3$ et $\mathrm{bdd}(5) = \neg x_1$.

Supposons que l'on désire définir la fonction f à 3 variables booléennes, de table de vérité $[0, 1, 1, 0, 1, 0, 1, 0]$ dans la table UniqueTable, il suffit de remarquer qu'en découpant la table par des tableaux de taille 1, 2, 4 et 8, on obtient les tables de vérité des fonctions à 0, 1, 2 puis 3 variables à ajouter dans la table :
- Le découpage en tableaux de de taille 1 redonne la liste des 8 fonctions $0, 1, 1, 0, 1, 0, 1, 0$. Notons que les constantes sont déjà définies dans la table UniqueTable.
- Le découpage en tableaux de taille 2 donne une liste de 4 fonctions à une variable : $x_1, \neg x_1, \neg x_1, \neg x_1$ ou exprimer en utilisant uniquement l'opérateur ite : $\mathrm{ite}(x_1, 1, 0), \mathrm{ite}(x_1, 0, 1), \mathrm{ite}(x_1, 0, 1), \mathrm{ite}(x_1, 0, 1)$. Il suffit donc d'ajouter dans la table les triplets $(1, 1, 0)$ et $(1, 0, 1)$. Imaginons pour notre propos qu'après les opérations d'ajout UniqueTable[2] $= (1, 1, 0)$ et UniqueTable[3] $= (1, 1, 0)$, notre liste de 4 fonctions s'exprime sur la table UniqueTable par la suite d'entrées $2, 3, 3, 3$.
- Le découpage en tableaux de taille 4 donne 2 fonctions à 2 variables : $(x_2 \wedge \neg x_1) \vee (\neg x_2 \wedge x_1), \neg x_1$ ou exprimer avec l'opérateur ite : $\mathrm{ite}(x_2, \mathrm{ite}(x_1, 0, 1), \mathrm{ite}(x_1, 1, 0))$, $\mathrm{ite}(x_2, \mathrm{ite}(x_1, 0, 1), \mathrm{ite}(x_1, 0, 1))$. Il suffit d'ajouter à notre table les triplets $(2, 3, 2)$ et $(2, 3, 3)$. En supposant que UniqueTable[4] $= (2, 3, 2)$, notre liste de 2 fonctions s'exprime par $4, 3$. En effet l'ajout du triplet $(2, 3, 3)$ ne modifie pas la table et retourne la valeur 3. Cette règle se justifie par l'identé $\mathrm{bdd}(3) = \mathrm{ite}(x_2, \mathrm{bdd}(3), \mathrm{bdd}(3))$.
- Finalement, en ajoutant le triplet $(3, 3, 4)$, la valeur retournée est la représentation de la fonction f dans la table UniqueTable.

Question 6 *Programmer une fonction d'ajout d'une fonction booléenne à N variables exprimer par sa table de vérité dans la table UniqueTable. Donner une description concise des structures de données et des algorithmes. Evaluer votre fonction sur les tables de vérité des fonctions F, G, H (pour $N = 6$). On donnera comme résultat la taille de la table UniqueTable (c-a-d le nombre d'entrées définies de UniqueTable)* *avant et après chaque appel de la fonction. Juste avant le premier appel, la table devra être initialisée et aura 2 pour taille.*

Soit d une entrée définie, on pose $\Phi(d)$ l'ensemble des entrées utilisées pour définir la fonction $\mathrm{bdd}(d)$. Formellement $\Phi(d)$ est défini inductivement par ;
– si $d = 0$ ou $d = 1$ alors $\Phi(d) = \{d\}$,
– sinon $\Phi(d) = \{d\} \cup \Phi(\mathrm{co}^+(d)) \cup \Phi(\mathrm{co}^-(d))$.

Nous appelons taille d'une entrée d définie, le cardinal de l'ensemble $\Phi(d)$. Notons que la taille des entrées 0 et 1 est toujours égale à 1 et que la taille de la fonction f de table de vérité $[0, 1, 1, 0, 1, 0, 1, 0]$ est 6.

Question 7 *Programmer une fonction de calcul de la taille d'une entrée définie. Donner une description concise des structures de données et des algorithmes. Appliquer votre algorithme aux entrées des fonctions F, G, H pour $N = 6$ et donner les résulats de ces appels.*

L'implémentation de l'opérateur ite sur notre structure repose sur l'identé suivante

$$\mathrm{ite}(f, g, h) = \mathrm{ite}(x, \mathrm{ite}(f_x^+, g_x^+, h_x^+), \mathrm{ite}(f_x^-, g_x^-, h_x^-)) \tag{1}$$

où f, g, h sont des fonctions booléennes et x une variable booléenne. Le calcul du $\mathrm{ite}(d_1, d_2, d_3)$ de trois entrées est obtenu en appliquant récursivement les règles suivantes :
– si $d_1 = 1$, retourner d_2,
– si $d_1 = 0$, retourner d_3,
– sinon, effectuer les quatre étapes suivantes

1. Calculer k égal au maximum de l'ensemble $\{\mathrm{var}(d_1), \mathrm{var}(d_2), \mathrm{var}(d_3)\}$.
2. Calculer les cofacteurs des entrées d_i pour la variable d'incide k. Les entrées de ces cofacteurs, notées $\mathrm{co}_k^+(d_i)$, $\mathrm{co}_k^-(d_i)$, sont obtenues directement par les formules : si $\mathrm{var}(d_i) = k$ alors $\mathrm{co}_k^+(d_i) = \mathrm{co}^+(d_i)$ et $\mathrm{co}_k^-(d_i) = \mathrm{co}^-(d_i)$; si $\mathrm{var}(d_i) < k$ alors $\mathrm{co}_k^+(d_i) = \mathrm{co}_k^-(d_i) = d_i$.
3. Calculer par des appels récursifs les entrées $u = \mathrm{ite}(\mathrm{co}_k^+(d_1), \mathrm{co}_k^+(d_2), \mathrm{co}_k^+(d_3))$ et $v = \mathrm{ite}(\mathrm{co}_k^-(d_1), \mathrm{co}_k^-(d_2), \mathrm{co}_k^-(d_3))$.
4. Ajouter le triplet (k, u, v) à la table UniqueTable et retourner le résultat.

Question 8 *Programmer une fonction récursive de calcul de l'opérateur ite. Donner une description concise des structures de données et des algorithmes. Appliquer votre algorithme aux formules $\neg F$, $\neg G$, $\neg H$, $F \wedge G$, ite(F, G, H) et donner la taille des résultats. Que remarquez-vous ?*

L'implémentation du calcul des cofacteurs, notées $\mathrm{co}_k^+(d)$, $\mathrm{co}_k^-(d)$, repose sur les mêmes principes que celui de l'opérateur ite : si $\mathrm{var}(d) \leq k$, le résultat est immédiat ; si $\mathrm{var}(d) > k$, il suffit d'appliquer récursivement les identités suivantes :

$$\mathrm{co}_k^+(d) = \mathrm{ite}(\mathrm{var}(d), \mathrm{co}_k^+(\mathrm{co}^+(d)), \mathrm{co}_k^+(\mathrm{co}^-(d))) \tag{2}$$

$$\mathrm{co}_k^-(d) = \mathrm{ite}(\mathrm{var}(d), \mathrm{co}_k^-(\mathrm{co}^+(d)), \mathrm{co}_k^-(\mathrm{co}^-(d))) \tag{3}$$

Question 9 *Programmer des fonctions récursives du calcul des cofacteurs. Donner une description concise des structures de données et des algorithmes. Appliquer votre algorithme aux formules* F_x^-, F_x^+ *pour* $x = x_1, \cdots x_N$ *et donner la taille des résultats.*

Question 10 *Donner une description concise des algorithmes de calcul de l'opération de substitution et de l'évaluation des quantificateurs "*$\exists$*", "*$\forall$*". Donner la taille des entrées des fonctions* $F(x/G)$, $G(x/F)$, $\exists x(F)$, $\neg\forall x(\neg F)$ *pour* $x = x_1, \cdots x_N$. *Que remarquez-vous ?*

3 Décodage d'un digicode

Attention : nous appliquerons les résultats de cette partie, aux fonctions F, G et H pour $N = 3$. Soit E un ensemble non vide de mots de longueur N sur l'alphabet $\mathbb{B}$.

Nous dirons qu'un mot w décode l'ensemble E si tout mot de M est un sous-mot de w. Par exemple, le mot 10011 décode l'ensemble $\{00, 01, 10, 11\}$, le mot 000111 décode $\{000, 111\}$.

Représentons les ensembles E de mots de longueur N par une fonction booléenne f à N variables : $u_1 \cdots u_N \in E$ si $f(u_1, \cdots, u_N) = 1$. Soit $w_1 \cdots w_M$ un mot de longueur M. On peut montrer que w décode l'ensemble E ssi

$$\forall x_1 \cdots \forall x_N : f(x_1, \cdots x_N) \Rightarrow \bigvee_{k=0}^{M-N} \left(\bigwedge_{i=1}^{N} (x_i \Leftrightarrow w_{i+k}) \right) \tag{4}$$

La formule 4 est une fonction booléenne à M variables (les variables $w_1, \cdots w_M$). Notons Π_M la fonction ainsi définie. Tester si il existe un mot de longueur M décodant l'ensemble E revient à vérifier si Π_M n'est pas la fonction nulle. Ce problème peut être résolu en utilisant les outils de la partie 2. En particulier, tester si une fonction est non nulle se résume à vérifier que l'entrée de la fonction n'est pas 0.

Question 11 *Concevoir un programme calculant la longueur du plus petit mot décodant une fonction booléenne à* N *variables. Donner une description concise des structures de données et des algorithmes. Appliquer votre algorithme à la fonction* 1 *et aux fonctions* F, G *et* H *pour* $N = 3$. *Donner la valeur des résultats. Vous prendrez soin d'initialiser la table d'unicité aussi souvent que possible pour éviter les problèmes d'engorgement de la table. (Indication : utiliser la variables* x_{i+N} *pour représenter une variable* w_i*)*

Question 12 *Concevoir un programme calculant un mot de taille minimal décodant une fonction booléenne à* N *variables. Donner une description concise des structures de données et des algorithmes. Appliquer votre algorithme aux fonctions de la question 11 et donner la valeur des résultats.*