Module I41, Licence Informatique 2

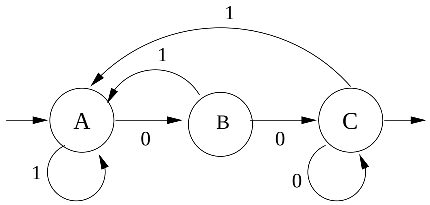
27 juillet 2006, correction

Le sujet est composé d'un problème et deux exercices indépendants à traiter en moins de deux heures. Tous les documents sont autorisés. Vous êtes invités à remettre une copie claire, concise, sans rature ni surcharge. Il est par ailleurs inutile de recopier l'énoncé... Le barème est donné à titre indicatif, la note finale tiendra compte de la présentation générale de la copie.

EXERCICES

[1] Automates [4 pts]

Considérons l'automate $\mathcal{A}$ sur l'alphabet $A = \{0, 1\}$:



(a) Les mots ε, 1100, 1001, 1010, 1111 et 1000 sont-il reconnus par l'automate $\mathcal{A}$?

(b) Donner l'expression la plus simple du langage $00^* + \varepsilon$.

(c) Déterminer le langage $\mathcal{L}(\mathcal{A})$ par la méthode des équations de langage.

(d) Si les mots de A^* sont la représentation en base 2 des entiers naturels, quelle est la propriété arithmétique des mots reconnus par l'automate ?

[a] absolument trivial.
[b] $00^* + \varepsilon = 0^*$.
[c] Equations des résiduels:

$$X_A = 1.X_A + 0.X_B$$
$$X_B = 1.X_A + 0.X_C$$
$$X_C = 1.X_A + 0.X_C + \epsilon$$

Pour mémoire la solution de $X = UX + V$ est U^*V dès que $\epsilon \notin U$. Les troisième équation donne

$$X_C = 0^*(1X_A + \epsilon) = 0^*1X_A + 0^*$$

La seconde donne

$$X_B = 1.X_A + 00^*1.X_A + 00^*$$

et donc

$$X_A = 1.X_A + 00^*1X_A + 000^* = 0^*1X_A + 000^* = (0^*1)^*000^*$$

[d] Les représentations des multiple de 4 (sauf 0) qui se terminent donc par 00.

[2] Logique de Hoare [2 pts]

Soient p et q deux propositions logiques. On suppose que p et q sont deux invariants de l'instruction itérative :

tant que b **faire** S **ftq**

Montrer que $(p \wedge q)$ est invariant de cette boucle.

Par définition,

$$p \wedge b \overset{S}{\Longrightarrow} p$$
$$q \wedge b \overset{S}{\Longrightarrow} q$$

Comme $p \wedge q \wedge b \Longrightarrow p$ et que $p \wedge q \wedge b \Longrightarrow q$, La règle du "ET" permet d'affirmer que :

```
Algorithme PGCD-Additif( a,b : entier)
données
  début
    tant que (0 < b) ∧ (0 < a) faire
          si (a < b) alors
                    b ← b − a
               sinon
                    a ← a − b
          finsi
    fintq
    retourner(a + b)
  fin
```

Figure 1: Algorithme du PGCD additif.

$$p \wedge q \wedge b \overset{S}{\Longrightarrow} p \wedge q$$

c'est-à-dire que $p \wedge q$ est un bien invariant de boucle.

PROBLEME

Soient a et b deux entiers quelconques. On rappelle que le PGCD de a et de b est le plus grand entier d qui divise à la fois a et b. On sait qu'il existe un algorithme efficace pour le déterminer. L'objectif de ce problème concerne le calcul de PGCD en mode additif, pour des implantations sur des machines privées des opérations de réduction modulaire.

[3] Implantation [3 pts]

(a) Donner une implantation itérative en langage C de l'algorithme de la figure (1).

(b) Donner une implantation récursive en langage C de l'algorithme de la figure (1).

[a]

```
int pgcd-additif(int a, int b)
{
 while ( (a>0) && (b>0) ) {
    if ( a > b ) a = a - b;
    else b = b - a;
 }
 return a+b;
}
```

[b]

```
int pgcd-rec(int a, int b)
{
 if ( ( a == 0 ) || ( b == 0) ) return a+b;
 if ( a > b ) return pgcd-rec( a - b, b);
 return pgcd-rec( b - a, a);
}
```

[4] Arithmétique [2 pts]

Soient a et b deux entiers positifs ou nul. Par définition, $\textsc{pgcd}(a,b) = \textsc{pgcd}(b,a)$.

(a) Montrer que si $a > 0$ alors $\textsc{pgcd}(a,0) = a$.

(b) Montrer que si $a > b$ alors $\textsc{pgcd}(a,b) = \textsc{pgcd}(b,a-b)$.

[a] 0 est divisible par tous les nombres positifs, et a est le plus grand diviseur de a.

[b] La distributivité de la multiplication sur l'addition montre que si d divise a et $b-a$ alors d divise a et b. La distributivité de la multiplication sur la soustraction montre que si d divise a et b alors d divise a et $b-a$. L'ensemble des diviseurs communs à a et b est donc identique à ceux de a et $b-a$.

[5] Preuve partielle [4 pts]

Soient a et b deux entiers positifs ou nul.Par définition, $\text{PGCD}(a,b) = \text{PGCD}(b,a)$.

(a) Montrer $a \geq 0$ est un invariant de boucle de l'algorithme du PGCDadditif.

(b) Est-ce que $a > 0$ est un autre invariant ?

(c) Soit γ un entier. Montrer que $\text{PGCD}(a,b) = \gamma$ invariant de boucle.

(d) Soient α et β les valeurs initiales des paramètres a et b. Faire une preuve partielle pour établir que le résultat de l'algorithme du PGCD additif est bien $\text{PGCD}(\alpha,\beta)$.

[a] Il suffit de remarquer que :

$$(a \geq 0) \wedge (a > b) \Longrightarrow (a - b > 0) \xrightarrow{a \leftarrow b-a} (a > 0)$$

et que

$$(a \geq 0) \xrightarrow{b \leftarrow b-a} (a > 0)$$

pour appliquer la règle de l'alternative et conclure.

[b] Non, $a > 0$ n'est pas un invariant.

[c] La formule (b) du [4] montre que $\text{PGCD}(a,b) = \gamma$ est un invariant de boucle. On sort de la boucle du programme (1) avec $\text{PGCD}(a,b) = \text{PGCD}(\alpha,\beta)) \wedge \neg((a > 0) \wedge (b > 0))$. On remarque que $b \geq 0$ est un invariant. La formule ci-dessous est vérifiée en fin sortie de boucle:

$$\text{PGCD}(a,b) = \text{PGCD}(\alpha,\beta)) \wedge \neg((a > 0) \wedge (b > 0)) \wedge (a \geq 0) \wedge (b \geq 0).$$

Comme,

$$\neg((a > 0) \wedge (b > 0)) \wedge (a \geq 0) \wedge (b \geq 0) \iff ((a = 0) \vee (b = 0)).$$

La propriété [a] de [4] permet de conclure.

[6] Preuve d'arrêt [3 pts]

(a) Proposer un variant de boucle pertinent. PGCD additif.

(b) Faire une preuve d'arrêt.

[a] L'une des variables a ou b décroit, et donc $a + b$ est un candidat pertinent.

[b] Les invariant $a \geq 0$ et $b \geq 0$ montrent que pour $a + b$ assez petit, au pire ≤ 0, l'une des variables a ou b est nulle.

[7] Temps de calcul [3 pts]

Soient $\alpha > 0$ et β deux entiers naturels, on note $i(\alpha,\beta)$ le nombre d'itérations de l'algorithme du PGCD additif pour initialisé par $a = \alpha$ et $b = \beta$.

(a) On suppose $\alpha > \beta$, et on note q et ρ les quotient et reste de la division euclidienne de α par β. Exprimer $i(\alpha,\beta)$ en fonction de q et $i(\beta,\rho)$.

(b) Soient q_1, q_2,... les quotients successifs, obtenus par l'algorithme d'Euclide pour calculer le PGCD des entiers α et β. Établir

$$i(\alpha,\beta) = \sum_{i=1}^{n} q_i.$$

(c) Déterminer les valeurs $0 < \beta < 144$ qui maximisent $i(144,\beta)$.

(d) Déterminer les valeurs $0 < \beta < 144$ qui minimisent $i(144,\beta)$.

[a] Les premières itérations réalisent la division euclidienne de α par β, soit :

$$i(\alpha,\beta) = q + (\beta,\rho).$$

[b] Pour tout entier $\gamma > 0$, on a $i(\gamma,0) = 0$, et la formule [a] conduit à :

$$i(\alpha,\beta) = \sum_{i=1}^{n} q_i + i(\rho_n, 0)$$

avec $i(\rho_n, 0) = 0$.

[c] La question (b) montre que $i(144, \beta) \leq 144$ avec égalité si et seulement si $\beta = 1$.

[d] La question n'était pas facile ! Le lien entre les nombres de Fibonacci et l'algorithme d'Euclide suggère $i(144, 89) = 11$ mais le minimum $i(144, 72) = 2 \ldots$