Exercice 1 : Égalité de tableaux avec sentinelle

On peut manipuler un tableau sans en connaître la taille à condition de pouvoir détecter la fin du tableau.[1] Un moyen consiste à mettre une valeur spéciale, appelée *sentinelle*, dans la case qui suit la dernière case utile du tableau. Il faut choisir une sentinelle différente des valeurs possibles des autres cases du tableau. Dans cet exercice on considère des tableaux contenant des entiers dans $\mathbb{N}$ et on prend la valeur **-1** comme sentinelle.

Exemple de tableaux A et B, à sentinelle, égaux :

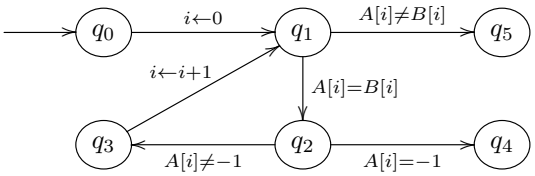
i	0	1	2	3	4	5	6	7	8	...
$A[i]$	5	3	4	7	**-1**	2	1	0	8	...
$B[i]$	5	3	4	7	**-1**	2	9	...		

Le but de cet exercice est de montrer la correction partielle du programme suivant, censé tester l'égalité de 2 tableaux A et B avec sentinelle. Dans cet exercice on considèrera que chaque instruction `return` ne fait pas d'affectation mais amène à un état de sortie différent. On obtient un automate à 6 états (de q_0 à q_5) avec deux états de sortie q_4 et q_5.

égalité de tableaux

```
int i ;
i:=0 ;
while(A[i]==B[i]){
  if (A[i]==-1){ return true;
     }
  i := i+1 ;
}
return false;
```

SOLUTION



ψ_4 *On souhaite démontrer qu'à l'état de sortie q_4 de l'algorithme on a*

$$\psi_4 : A[i] = -1 \wedge \forall k \in [0..i[,\ A[k] = B[k]$$

ψ_2 *D'après la transition test $q_2 \to q_4$ on doit choisir ψ_2 telle que* $\psi_2 \ \wedge \underbrace{A[i] = -1}_{test} \implies \psi_4$.

On choisit de prendre

$$\psi_2 : \forall k \in [0..i[,\ A[k] = B[k]$$

ψ_1 *D'après la transition test $q_1 \to q_2$ on doit choisir ψ_1 telle que* $\psi_1 \ \wedge \underbrace{A[i] = B[i]}_{test} \implies \psi_2$.

On réécrit ψ_2 sous une forme équivalente pour faire apparaître les termes $A[i]$ et $B[i]$.

$$\begin{array}{rl} \psi_2 : & \forall k \in [0..i[,\ A[k] = B[k] \\ \equiv & \forall k \in [0..i-1], A[k] = B[k] \wedge \underbrace{A[i] = B[i]}_{test} \end{array}$$

On choisit de prendre

$$\psi_1 : \forall k \in [0..i-1[,\ A[k] = B[k]$$

ψ_3 *La transition $q_3 \to q_1$ ne fait que des affectations donc le meilleur choix possible est*

$$\psi_3 \equiv \psi_1[i \leftarrow i+1] \equiv \forall k \in [0..i[,\ A[k] = B[k]$$

VÉRIF1 *Les propriétés ψ_2 et ψ_3 sont choisies. La transition $q_2 \to q_3$ nous permet d'effectuer une vérification. Nos choix d'invariants doivent satisfaire l'implication*

1. C'est ainsi que sont implantées les chaînes de caractères avec pour sentinelle le caractère `'\0'`.

$$\{\!|\forall k \in [0..i],\ A[k] = B[k]\ |\!\} : \psi_2\ \wedge\ \underbrace{A[i] \neq -1}_{test}$$

$$\stackrel{?}{\Longrightarrow} \psi_3 : \{\!|\forall k \in [0..i],\ A[k] = B[k]\ |\!\}$$

Preuve de l'implication Il faut montrer ψ_3 à l'aide de ψ_2 et du test. C'est immédiat puisque ψ_2 et ψ_3 sont identiques. □

$\boxed{\psi_5}$ *On souhaite démontrer qu'à l'état de sortie q_5 de l'algorithme on a*

$\psi_5 : \exists k \in [0..i],\ A[k] \neq B[k]$

VÉRIF2 *Les propriétés ψ_1 et ψ_5 sont choisies. La transition test $q_1 \to q_5$ nous permet d'effectuer une vérification. Nos choix d'invariants doivent satisfaire l'implication*

$$\{\!|\forall k \in [0..i-1],\ A[k] = B[k]\ |\!\} : \psi_1\ \wedge\ \underbrace{A[i] \neq B[i]}_{test}$$

$$\stackrel{?}{\Longrightarrow} \psi_5 : \{\!|\exists k \in [0..i],\ A[k] \neq B[k]\ |\!\}$$

Preuve de l'implication On réécrit ψ_5 sous une forme équivalent pour faire apparaître les termes $A[i]$ et $B[i]$.

$$\begin{array}{rl} \psi_5 : & \exists k \in [0..i],\ A[k] \neq B[k] \\ \equiv & \exists k \in [0..i-1], A[k] = B[k] \vee \underbrace{A[i] \neq B[i]}_{test} \end{array}$$

Il faut montrer ψ_5 à l'aide de ψ_1 et du test. Puisque ψ_5 s'écrit sous la forme d'une disjonction ($\vee$) il suffit de montrer l'une des parties de la disjonction. C'est immédiat puisque la partie droite correspond au test. □

$\boxed{\psi_0}$ *La transition $q_0 \to q_1$ nous permet de déterminer les conditions d'utilisation du programme. Elle fait que des affectations donc le meilleur choix possible est*

$$\begin{array}{l} \psi_0 \equiv \psi_1[i \leftarrow 0] \\ \equiv \forall k \in [0..-1],\ A[k] = B[k] \\ \equiv \{\!|\ |\!\} \end{array}$$

Exercice 2 : Calcul du pgcd de deux entiers

On considère l'algorithme, donné sous forme d'automate, qui caclule le *pgcd* (*« Plus Grand Diviseur Commun »*) de deux entiers A et B donnés. On prétend qu'à la sortie de l'automate on aura « $a = pgcd(A, B)$ » où $pgcd(A, B)$ représente le résultat mathématique qu'on essaie de calculer.

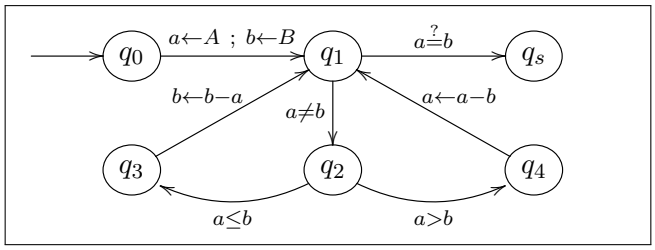


FIGURE 1 – Algorithme du *pgcd* sous forme d'automate

Propriétés du pgcd et principe de l'algorithme On note $x|y$ le fait que x divise y, c'est-à-dire que y est un multiple de x, soit encore $\exists k \in \mathbb{N}^*, y = k \times x$. On rappelle quelques propriétés du *pgcd* qui seront utiles pour prouver la correction de ce programme.

1) $pgcd(x, x) = x$
2) $p|x \ \wedge \ p|y \Longrightarrow p|pgcd(x, y)$
3) $p|x \ \wedge \ p|y \Longrightarrow p|x - y$
4) $p|x \ \wedge \ p|y \Longrightarrow p|x + y$
5) $p|q \ \wedge \ q|p \Longrightarrow q = p$

Ces propriétés permettent de démontrer les deux égalités suivantes

(i) $pgcd(a, b - a) = pgcd(a, b)$ si $a \leq b$

(ii) $pgcd(a - b, b) = pgcd(a, b)$ si $a \geq b$

sur lesquelles repose l'algorithme.

Q1. Complétez la preuve de (i) ci-dessous.

SOLUTION

Soit $p \stackrel{def}{=} pgcd(a, b)$ *et* $q \stackrel{def}{=} pgcd(a - b, b)$.
Puisque $p = pgcd(a, b)$ *alors* $p|a$ *et* $p|b$ *et donc* $p|a - b$ *(d'après 3) mais alors* $p|pgcd(a - b, b)$ *(d'après 2) et donc* $p|q$ *(par définition de* q*).*
Puisque $q \stackrel{def}{=} pgcd(a - b, b)$ *alors* $q|a - b$ *et* $q|b$ *et donc* $q|a$ *(d'après 4) mais alors* $q|pgcd(a, b)$ *(d'après 2) et donc* $q|p$ *(par définition de* p*).*
Conclusion : $q = p$ *(d'après 5), autrement dit* $pgcd(a - b, b) = pgcd(a, b)$. *On démontre de la même manière que* $pgcd(a, b - a) = pgcd(a, b)$.

Q2. Preuve de correction partielle

Indication : Vous prendrez pour invariant en q_1 une propriété de la forme : $pgcd(a, b) = pgcd(A, B)$

SOLUTION

$\boxed{\psi_s}$ *On souhaite démontrer qu'à la sortie de l'algorithme on a* $a = pgcd(A, B)$.
On prend $\psi_s : a = pgcd(A, B)$

$\boxed{\psi_1}$ *D'après la transition test* $q_1 \rightarrow q_s$ *on doit choisir* ψ_1 *telle que* $\psi_1 \ \wedge \ \underbrace{a = b}_{test} \Longrightarrow \psi_s$.

On choisit de prendre $\psi_1 : pgcd(a, b) = pgcd(A, B)$

Preuve de l'implication D'après la condition du test $a = b$ on a $pgcd(a, b) = pgcd(a, a)$ et $pgcd(a, a) = a$ d'après la propriété 1 du $pgcd$, ce qui donne $a = pgcd(a, b)$. En combinant cette égalité avec $\psi_1 : pgcd(a, b) = pgcd(A, B)$, on obtient l'égalité de ψ_s donc l'implication est valide avec ce choix de ψ_1. □

$\boxed{\psi_3}$ *La transition $q_3 \to q_1$ ne fait que des affectations donc le meilleur choix possible est*

$$\psi_3 \equiv \psi_1[b \leftarrow b - a] \equiv pgcd(a, b - a) = pgcd(A, B)$$

$\boxed{\psi_4}$ *La transition $q_4 \to q_1$ ne fait que des affectations donc le meilleur choix possible est*

$$\psi_4 \equiv \psi_1[a \leftarrow a - b] \equiv pgcd(a - b, b) = pgcd(A, B)$$

$\boxed{\psi_2}$ *D'après la transition $q_2 \to q_3$ on doit choisir ψ_2 telle que* $\psi_2 \ \wedge \ \underbrace{a \leq b}_{test} \Longrightarrow \psi_3 : pgcd(a, b - a) = pgcd(A, B)$

On choisit de prendre $\psi_2 : pgcd(a, b) = pgcd(A, B)$

Preuve de l'implication Avec la condition du test $a \leq b$ on sait d'après la propriété (i) que $pgcd(a, b - a) = pgcd(a, b)$. En combinant cette égalité avec $\psi_2 : pgcd(a, b) = pgcd(A, B)$, on obtient l'égalité de ψ_3 donc l'implication est valide avec ce choix de ψ_1. □

$\boxed{\textbf{VÉRIF1}}$ *Les propriétés ψ_2 et ψ_4 sont choisies. La transition $q_2 \to q_4$ nous permet d'effectuer une vérification. Nos choix d'invariants doivent satisfaire l'implication*

$$\{pgcd(a, b) = pgcd(A, B)\} : \psi_2 \ \wedge \ \underbrace{a > b}_{test} \Longrightarrow \psi_4 : \{pgcd(a - b, b) = pgcd(A, B)\}$$

Preuve de l'implication De la condition du test $a > b$ on peut déduire que $a \geq b$ et d'après la propriété (ii) que $pgcd(a - b, b) = pgcd(a, b)$. En combinant cette égalité avec $\psi_2 : pgcd(a, b) = pgcd(A, B)$, on obtient l'égalité de ψ_4 donc l'implication est valide. □

$\boxed{\textbf{VÉRIF2}}$ *Les propriétés ψ_1 et ψ_2 sont fixées. La transition $q_1 \to q_2$ nous permet d'effectuer une vérification. Nos choix d'invariants doivent satisfaire l'implication*

$$\{pgcd(a, b) = pgcd(A, B)\} : \psi_1 \ \wedge \ \underbrace{a \neq b}_{test} \Longrightarrow \psi_2 : \{pgcd(a, b) = pgcd(A, B)\}$$

Preuve L'implication est triviale puisque de la forme $Prop \ \wedge \ Test \Rightarrow Prop$: la propriété à prouver en conclusion de l'implication (partie droite) est présente en hypothèse de l'implication (partie gauche). □

$\boxed{\psi_0}$ *La transition $q_0 \to q_1$ ne fait que des affectations donc le meilleur choix possible est*

$$\psi_0 \equiv \psi_1[a \leftarrow A \ ; \ b \leftarrow B] \equiv pgcd(A, B) = pgcd(A, B) \equiv \{\}$$

L'invariant ψ_0 donne les conditions d'utilisation du programme qui garantissent qu'en sortie on aura $a = pgcd(A, B)$. Le fait que $\psi_0 \equiv \{\}$ indique que l'algorithme n'a pas de restriction d'utilisation pour garantir sa correction partielle.