Exercice 1 – Calcul des termes de la suite de Fibonacci – version avec seulement deux variables (9 pt)

```
i:=N-1 ; x:=1; y:=1 ;
while(i<>0){ y:=y+x ; x:=y-x ; i:=i-1; }
```

On prétend qu'à la sortie du programme le résultat y satisfait la propriété $y = fib(N)$ où

$$\begin{cases} fib(0) = 1 \\ fib(1) = 1 \\ fib(n+1) = fib(n) + fib(n-1) \end{cases}$$

Q1. (1 pt) Donnez les transitions de l'automate correspondant au programme ci-dessus (q_0 représente le point d'entrée et q_s celui de sortie de l'automate).

$$q_1 \xrightarrow{i=0} q_s$$
$$q_0 \xrightarrow{i:=N-1;x:=1;y:=1;} q_1$$
$$q_1 \xrightarrow{i\neq 0} q_2 \quad q_2 \xrightarrow{y\ :=\ x+y} q_3 \quad q_3 \xrightarrow{x\ :=\ y-x} q_4 \quad q_4 \xrightarrow{i\ :=\ i-1} q_1$$

Q2. (4 pt) **Preuve de correction partielle** Rédigez la preuve (en suivant la méthode de Floyd-Dijsktra-Hoare) qu'à la sortie du programme $y = fib(N)$. La qualité et la précision de la rédaction compte pour une grande part dans la note. Vous prendrez pour invariant en q_1 une propriété de la forme :

$$y = fib(N-i) \ \wedge \ \ x \ = fib(N-i-k)$$

où k est une constante que vous devrez déterminer

SOLUTION

1. *Transition* $q_1 \xrightarrow{i=0} q_s$

$$\psi_1 \ \wedge \ i = 0 \Longrightarrow \overbrace{y = fib(N)}^{\psi_s}$$

On choisit $\psi_1 \stackrel{def}{=} y = fib(N-i) \ \wedge \ x = fib(N-i-k)$
Lorsque $i = 0$, ψ_1 *nous donne* $y = fib(N)$ *(et d'autres propriétés qui ne sont pas utiles pour* ψ_s*).*

2. *Transition* $q_4 \xrightarrow{i:=i-1} q_1$

$$\begin{aligned} \psi_4 &\stackrel{def}{=} \psi_1[i := i-1] \\ &\stackrel{def}{=} y = fib(N-(i-1)) \ \wedge \ x = fib(N-(i-1)-k) \\ &\equiv y = fib(\,(N-i)\,+1) \ \wedge \ x = fib(\,(N-i)\,+1-k) \end{aligned}$$

3. *Transition* $q_3 \xrightarrow{x:=y-x} q_4$

$$\begin{aligned} \psi_3 &\stackrel{def}{=} \psi_4[x \leftarrow y-x] \\ &\stackrel{def}{=} y = fib(\,(N-i)\,+1) \ \wedge \ y-x = fib(\,(N-i)\,+1-k) \end{aligned}$$

4. *Transition* $q_2 \xrightarrow{y:=x+y} q_3$

$$\begin{aligned} \psi_2 &\stackrel{def}{=} \psi_3[y \leftarrow x+y] \\ &\stackrel{def}{=} x+y = fib(\,(N-i)\,+1) \ \wedge \ (x+y)-x = fib(\,(N-i)\,+1-k) \\ &\equiv x+y = fib(\,(N-i)\,+1) \ \wedge \ y = fib(\,(N-i)\,+1-k) \end{aligned}$$

5. *Transition* $q_1 \xrightarrow{i \neq 0} q_2$ VÉRIFICATION DES INVARIANTS CHOISIS *Il s'agit de vérifier que limplication suivante est valide.*

$$\overbrace{y = fib(N-i) \ \wedge \ x = fib(\,(N-i)\, - k)}^{\psi_1} \ \wedge \ \overbrace{i \neq N}^{garde}$$

$$\overset{?}{\Longrightarrow} \overbrace{\underbrace{x + y \overset{?}{=} fib(\,(N-i)\, + 1)}_{(**)} \ \wedge \ \underbrace{y \overset{?}{=} fib(\,(N-i)\, + 1 - k)}_{(*)}}^{\psi_2}$$

$(*)$ *Si on choisit* $k = 1$ *alors* $(*)$ *devient* $y \overset{?}{=} fib(\,(N-i)$ *et c'est alors le premier terme de la prémisse de l'implication, donc cette partie de la conclusion de l'implication est trivialement démontrée.*

$(**)$ *En prenant* $k = 1$ *on a alors en prémisse de l'implication* $y = fib(N-i)$ *et* $x = fib((N-i)-1)$, *on peut remplacer* x *et* y *par leurs valeurs dans le terme* $(**)$ *qui devient alors*

$$fib(N-i) + fib((N-i)-1) \overset{?}{=} fib((N-i)+1)$$

Cette égalité valide d'après la définition du $n^{ème}$ *terme de la suite de Fibonacci avec* $n = N - i$

Q3. (2 pt) Récapitulez les invariants

$\psi_1 \overset{def}{=}$

$\psi_2 \overset{def}{=}$

$\psi 3 \overset{def}{=}$

$\psi 4 \overset{def}{=}$

Q4. (1 pt) En déduire les conditions d'utilisation du programme. Détaillez vos étapes de calcul et de simplification.

SOLUTION

Transition $q_0 \xrightarrow{i:=N-1; x:=1; y:=1} q_1$

$$\begin{array}{rl} \psi_0 & \overset{def}{=} \ \psi_1[i \leftarrow N-1; x \leftarrow 1; y \leftarrow 1] \\ & \equiv \ \big(y = fib(N-i) \ \wedge \ x = fib(N-i-1)\big)\,[i \leftarrow N-1; x \leftarrow 1; y \leftarrow 1] \\ & \equiv \ \big(1 = fib(N - \,(N-1)\,) \ \wedge \ 1 = fib(N - \,(N-1)\, - 1)\big) \\ & \equiv \ \underbrace{1 = fib(1)}_{vrai} \ \wedge \ \underbrace{1 = fib(0)}_{vrai} \\ & \equiv \ \textit{vrai par définition de la suite de Fibonacci} \end{array}$$

Q5. (1 pt) Comment se comporte le programme pour $N = 0$? Expliquez pourquoi la condition $N > 0$ n'apparaît pas comme condition d'utilisation du programme ?