

$\frac{\square}{0.5\text{ pt}}$

Q2. Donnez la propriété ψ_s associée à l'état de sortie q_s du programme.

SOLUTION

$$\psi_s \stackrel{\text{def}}{=} r = \sum_{k=0}^N c[k] \times 10^k$$

$\frac{\square}{2\text{ pt}}$

Q3. Déterminez les propriétés (en justifiant votre choix)

$$\psi_5 \stackrel{\text{def}}{=} \psi_5 \stackrel{\text{def}}{=} \psi_s[i \leftarrow 42] \equiv \psi_s \text{ car } i \notin \psi_s$$

$$\psi_4 \stackrel{\text{def}}{=} \psi_5[r \leftarrow r + D[i]] \equiv r + D[i] = \sum_{k=0}^N D[k] \times 10^k$$

$\frac{\square}{2\text{ pt}}$

Q4. Choix de l'invariant de boucle

(a) Écrivez l'implication correspondant à la transition $q_1 \xrightarrow{i \leq 0} q_4$

(b) Choisissez la propriété ψ_1 .

Indication :

$$\psi_1 \stackrel{\text{def}}{=} r = \frac{\left(\sum_{k=i+1}^N D[k] \times 10^k \right)}{(10^i)} \wedge i \geq 0$$

(c) Prouvez l'implication.

SOLUTION

On choisit $\psi_1 \stackrel{\text{def}}{=} r = \frac{\sum_{k=i+1}^N D[k] \times 10^k}{10^i} \wedge i \geq 0$ qui permet de montrer l'implication $\psi_1 \wedge \overbrace{i \leq 0}^{\text{test}} \stackrel{?}{\implies} \psi_4$ associée à la transition $q_1 \xrightarrow{i \leq 0} q_4$.

Preuve :

$$\left\{ \begin{array}{l} i \geq 0 \\ r = \frac{\sum_{k=i+1}^N D[k] \times 10^k}{10^i} \end{array} \right\} = \psi_1 \wedge \overbrace{i \leq 0}^{\text{test}}$$

$$\implies \left\{ \begin{array}{l} i = 0 \dots \dots \dots \text{puisque } i \geq 0 \text{ et } i \leq 0 \\ r = \frac{\sum_{k=i+1}^N D[k] \times 10^k}{10^i} \text{ on remplace } i \text{ par } 0 \end{array} \right\}$$

$$\implies \left\{ \begin{array}{l} i = 0 \\ r = \frac{\sum_{k=1}^N D[k] \times 10^k}{10^0} \end{array} \right\}$$

$$\implies \left\{ \begin{array}{l} i = 0 \\ r + D[0] = \sum_{k=1}^N D[k] \times 10^k + D[0] \text{ on ajoute } D[0] \text{ de chaque côté} \end{array} \right\}$$

$$\implies \left\{ \begin{array}{l} D[0] = D[i] \dots \dots \dots \text{puisque } i = 0 \\ r + D[0] = \sum_{k=0}^N D[k] \times 10^k \text{ on remplace } D[0] \text{ par } D[i] \end{array} \right\}$$

$$\implies \left\{ r + D[i] = \sum_{k=0}^N D[k] \times 10^k \right\} \stackrel{\text{def}}{=} \psi_4$$

□

$\frac{\square}{3.5 \text{ pt}}$

Q5. Terminez la preuve de correction partielle du programme.

SOLUTION

$$\begin{aligned} \psi_3 &\stackrel{\text{def}}{=} \psi_1[i \leftarrow i - 1] \equiv r = \frac{\sum_{k=(i-1)+1}^N D[k] \times 10^k}{10^{i-1}} \wedge i - 1 \geq 0 \equiv r = \frac{\sum_{k=i}^N D[k] \times 10^k}{10^{i-1}} \wedge i \geq 1 \\ \psi_2 &\stackrel{\text{def}}{=} \psi_3[r \leftarrow 10 \times (r + D[i])] \equiv 10(r + D[i]) = \frac{\sum_{k=i}^N D[k] \times 10^k}{10^{i-1}} \wedge i \geq 1 \\ &\equiv r + D[i] = \frac{\sum_{k=i}^N D[k] \times 10^k}{10^i} \wedge i \geq 1 \\ &\equiv r = \frac{\sum_{k=i}^N D[k] \times 10^k - 10^i \times D[i]}{10^i} \wedge i \geq 1 \\ &\equiv r = \frac{\sum_{k=i+1}^N D[k] \times 10^k}{10^i} \wedge i \geq 1 \end{aligned}$$

$\frac{\square}{2 \text{ pt}}$

VÉRIFICATION

$$\underbrace{\underbrace{\frac{\sum_{k=i+1}^N D[k] \times 10^k}{10^i}}_{\psi_1} \wedge i \geq 0}_{\textcircled{1}} \wedge \underbrace{i > 0}_{\text{test}} \stackrel{?}{\implies} \underbrace{\underbrace{\frac{\sum_{k=i-1}^N D[k] \times 10^k}{10^i}}_{\psi_2} \wedge i \geq 1}_{\text{test}}_{\textcircled{1}}$$

$\frac{\square}{1 \text{ pt}}$

Q6. Donnez les conditions d'utilisations du programme et concluez.

SOLUTION

$$\psi_0 \stackrel{\text{def}}{=} \psi_1[i \leftarrow N ; r \leftarrow 0] \equiv 0 = \sum_{k=N+1}^{k=N} D[k] \times 10^k \wedge N \geq 0 \equiv 0 = 0 \wedge N \geq 0 \equiv N \geq 0$$

Conclusion : le programme fonctionne pour un tableau contenant au moins une case, la case 0.

$\frac{\square}{10 \text{ pt}}$

Exercice 2 : Calculs de factorielle

On considère la définition de la factorielle sous la forme d'une suite récurrente

$$\begin{cases} \text{fac}(0) &= 1 \\ \text{fac}(i+1) &= i \times \text{fac}(i) \quad \text{pour } i+1 > 0 \end{cases}$$

$\frac{\square}{1.5 \text{ pt}}$

Q1. Programme Donnez un programme qui calcule la factorielle de N

PROGRAMME

```

1  r:= 1 ; i:= 0 ;
2  while( i < N ){
3      i:=i+1 ;
4      r:=i*r ;
5  }
```

$\frac{\square}{0.5 \text{ pt}}$

Q2. Automate Donnez les transitions de l'automate correspondant à votre programme.

$$\begin{aligned} q_0 &\xrightarrow{r \leftarrow 1 ; i \leftarrow 0} q_1 \xrightarrow{i \geq N} q_s \\ q_1 &\xrightarrow{i < N} q_2 \xrightarrow{i \leftarrow i+1} q_3 \xrightarrow{r \leftarrow i \times r} q_1 \end{aligned}$$



Q3. Question de cours Expliquez ce que signifie qu'une propriété ψ est un invariant de l'état q

SOLUTION

La propriété ψ est un invariant de l'état q signifie que chaque fois que l'exécution de l'automate passe dans l'état q les variables du programme satisfont la propriété ψ .



Q4. Propriété de correction Donnez la propriété de correction du programme sous la forme d'un invariant d'état.



Q5. Preuve de correction partielle Montrez la correction partielle de votre programme à l'aide de la méthode de Floyd-Dijkstra-Hoare.

SOLUTION

$$\begin{aligned} \psi_s &\stackrel{def}{=} i = N \wedge r = fac(i) \equiv i = N \wedge r = fac(N) \\ \psi_1 &\stackrel{def}{=} 0 \leq i \wedge i \leq N \wedge r = fac(i) \\ \psi_3 &\stackrel{def}{=} \psi_1[r \leftarrow i \times r] \equiv 0 \leq i \wedge i \leq N \wedge i \times r = fac(i) \\ \psi_2 &\stackrel{def}{=} \psi_3[i \leftarrow i + 1] \equiv 0 \leq i + 1 \wedge i + 1 \leq N \wedge (i + 1) \times r = fac(i + 1) = (i + 1) \times fac(i) \end{aligned}$$



On ne peut pas simplifier ψ_2 . En effet, on ne peut pas diviser chacun des membres par $(i + 1)$ car on ne sait pas si $(i + 1) \neq 0$

Vérification

$$\overbrace{0 \leq i \wedge i \leq N \wedge r = fac(i)}^{\psi_1} \wedge \overbrace{i < N}^{test} \stackrel{?}{\implies} \overbrace{0 \leq i + 1}^{(*)} \wedge \overbrace{i + 1 \leq N}^{(**)} \wedge \overbrace{(i + 1) \times r = fac(i + 1)}^{(***)}$$

Preuve de (*): $0 \leq i \implies 0 < i + 1 \xrightarrow{OK} 0 \leq i + 1$ □

Preuve de ():** $i < N$ d'après le test et $i \in \mathbb{N}$ donc $i + 1 \leq N$ □

Preuve de (*):**

(i) $0 \leq i \implies 0 < i + 1 \implies 1 \leq i + 1$ donc, on peut appliquer la définition de fac :

$$fac(i + 1) = (i + 1) \times fac(i)$$

(ii) $0 \leq i \implies 0 < i + 1 \implies 0 \neq i + 1$ donc, on peut diviser par $(i + 1)$

D'après (i) et (ii), $(i + 1) \times r = fac(i + 1) \equiv (i + 1) \times r = (i + 1) \times fac(i) \equiv r = fac(i)$ qui fait partie de ψ_1 □



Q6. Conditions d'utilisation En déduire les conditions d'utilisation de votre programme qui garantissent qu'il calcule bien la factorielle de N .

SOLUTION

$$\psi_0 \stackrel{def}{=} \psi_1[r \leftarrow 1 ; i \leftarrow 0] \equiv 0 \leq 0 \wedge 0 \leq N \wedge 1 = fac(0) \equiv 0 \leq N$$



10 pt

Exercice 3 : Calcul des termes de la suite de Fibonacci – version avec seulement deux variables

PROGRAMME

```

1  i:=N-1 ; x:=1; y:=1 ;
2  while(i<>0){
3      y:=y+x ;
4      x:=y-x ;
5      i:=i-1 ;
6  }
```

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

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



1 pt

Q1. 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).

$$\begin{array}{l} q_1 \xrightarrow{i=0} q_s \\ q_0 \xrightarrow{i \leftarrow N-1 ; x \leftarrow 1 ; y \leftarrow 1} q_1 \\ q_1 \xrightarrow{i \neq 0} q_2 \quad q_2 \xrightarrow{y \leftarrow x+y} q_3 \quad q_3 \xrightarrow{x \leftarrow y-x} q_4 \quad q_4 \xrightarrow{i \leftarrow i-1} q_1 \end{array}$$



6 pt

Q2. Preuve de correction partielle Rédigez la preuve (en suivant la méthode de Floyd-Dijkstra-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 $q_1 \xrightarrow{i=0} q_s$

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

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 $q_4 \xrightarrow{i \leftarrow i-1} q_1$

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

3. Transition $q_3 \xrightarrow{x \leftarrow y-x} q_4$

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

4. Transition $q_2 \xrightarrow{y \leftarrow x+y} q_3$

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

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

$$\begin{aligned} &\overbrace{y = fib(N-i) \wedge x = fib((N-i)-k)}^{\psi_1} \wedge \overbrace{i \neq N}^{garde} \\ \Rightarrow &\overbrace{x+y \stackrel{?}{=} fib((N-i)+1)}^{(**)} \wedge \overbrace{y \stackrel{?}{=} fib((N-i)+1-k)}^{(*)} \end{aligned}$$

Preuve (*): Si on choisit $k = 1$ alors (*) devient $y \stackrel{?}{=} fib((N-i))$ 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. \square

Preuve ()**: 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

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

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

\square
1 pt

Q3. En déduire les conditions d'utilisation du programme. Détaillez vos étapes de calcul et de simplification.

SOLUTION

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

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

\square
2 pt

Q4. Comment se comporte le programme pour $N = 0$? Expliquez pourquoi la condition $N > 0$ n'apparaît pas comme condition d'utilisation du programme?