NOM .... Prénom .... Groupe ...
ou NUMÉRO D'ÉTUDIANT .... si l'examen est anonyme

RICM3 – Automates et Grammaires

Durée : 1h30, sans documents.

- Tous les appareils électroniques sont interdits à l'exception des montres
- Le barème est donné à titre indicatif
- Le sujet comporte 3 exercices indépendants
- Il est noté sur 30
- Répondez sur votre copie sauf pour les questions avec pointillés
- N'oubliez pas de mettre votre nom ou votre numéro d'étudiant sur le sujet
- Commencez par lire tout le sujet pour repérer les questions faciles

10 pt

Exercice 1 : Schéma de Horner

On considère un tableau `D[0..N]` remplit de $N+1$ digit (appartenant à $\{0,\ldots,9\}$). On peut voir un tableau de digits comme un entier en considérant que le digit de case 0 correspond aux unités, le digit de la case 1 aux dizaines, celui de la case 2 aux centaines et plus généralement le digit de la case i correspond au coefficient de puissance 10^i.

Exemple : le tableau

i	0	1	2	3	4	5	....	$N-1$	N
$D[i]$	5	0	7	3	1	0	...0...	0	0

correspond à l'entier 13705

De manière générale, le tableau

i	0	1	2	3	...	$N-1$	N
$D[i]$	d_0	d_1	d_2	d_3	...	d_{N-1}	d_N

correspond à l'entier

$$d_0 + d_1 \times 10 + d_2 \times 10^2 + d_3 \times 10^3 + \ldots + d_{N-1} \times 10^{N-1} + d_N \times 10^N = \sum_{k=0}^{N} D[k] \times 10^k$$

On prétend que le programme suivant calcule l'entier correspondant au tableau de digits `D[0..N]`, et qu'à la sortie du programme la variable r contient la valeur $\sum_{k=0}^{N} D[k] \times 10^k$.

Dans le corrigé complet on considère le programme suivant plus difficile à prouver :

PROGRAMME

```
i:=N ; r:=0 ;
while(i>0){
 r := 10 * (r + D[i]) ;
 i := i-1 ;
}
r := r +  D[0]
u  := 42 ;
```

1 pt

Q1. Donnez l'automate correspondant au programme.

$$q_0 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} q_1$$

$$q_1 \xrightarrow{\ldots\ldots\ldots} q_4 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} q_5 \xrightarrow{\ldots\ldots\ldots\ldots} q_s$$

$$q_1 \xrightarrow{\ldots\ldots\ldots} q_2 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} q_3 \xrightarrow{\ldots\ldots\ldots\ldots\ldots} q_1$$

$\frac{\quad}{0.5\,pt}$

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

SOLUTION

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

$\frac{\quad}{2\,pt}$

Q3. Déterminez les propriétés (en justifiant votre choix)

$\psi_5 \stackrel{def}{=}$

$\psi_4 \stackrel{def}{=}$

$\frac{\quad}{2\,pt}$

Q4. Choix de l'invariant de boucle

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

(b) Choisissez la propriété ψ_1.

Indication :

$$\psi_1 \stackrel{def}{=} r = \frac{\left(\sum_{k=\cdots\cdots}^{N} D[k] \times 10^k\right)}{\left(10^{\cdot\,\cdot}\right)} \wedge \cdots\cdots$$

(c) Prouvez l'implication.

SOLUTION

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

Preuve :

$$\left\{\begin{array}{l} i \geq 0 \\ r = \dfrac{\sum_{k=\,.......}^{N} D[k] \times 10^k}{10^{.}} \end{array}\right\} = \psi_1 \;\wedge\; \overbrace{i \leq 0}^{test}$$

$$\Longrightarrow \left\{\begin{array}{ll} i = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots & \textit{puisque } i \geq 0 \textit{ et } i \leq 0 \\ r = \dfrac{\sum_{k=\,.......}^{N} D[k] \times 10^k}{10^{.}} & \textit{on remplace } i \textit{ par 0} \end{array}\right\}$$

$$\Longrightarrow \left\{\begin{array}{l} i = 0 \\ r = \dfrac{\sum_{k=\,.}^{N} D[k] \times 10^k}{10^{.}} \end{array}\right\}$$

$$\Longrightarrow \left\{\begin{array}{ll} i = 0 & \\ r + D[0] = \sum_{k=\,.}^{N} D[k] \times 10^k + D[0] & \textit{on ajoute } D[0] \textit{ de chaque côté} \end{array}\right\}$$

$$\Longrightarrow \left\{\begin{array}{ll} D[0] = D[i] \ldots\ldots\ldots\ldots\ldots\ldots & \textit{puisque } i = 0 \\ r + D[0] = \sum_{k=\,.}^{N} D[k] \times 10^k & \textit{on remplace } D[0] \textit{ par } D[i] \end{array}\right\}$$

$$\Longrightarrow \left\{\begin{array}{l} r + D[0] = \sum_{k=\,.}^{N} D[k] \times 10^k \end{array}\right\} \stackrel{def}{=} \psi_4$$

□

3.5 pt

Q5. Terminez la preuve de correction partielle du programme.

SOLUTION

$$\psi_3 \stackrel{def}{=} \psi_1[i \leftarrow i-1] \equiv r = \frac{\sum_{k=\,..........+1}^{N} D[k] \times 10^k}{10^{.......}} \;\wedge\; .... \geq 0 \equiv r = \frac{\sum_{k=\,.}^{N} D[k] \times 10^k}{10^{.......}} \;\wedge\; . \geq 1$$

$$\psi_2 \stackrel{def}{=} \psi_3[r \leftarrow 10 \times (r + D[i])] \equiv .... = \frac{\sum_{k=\,.}^{N} D[k] \times 10^k}{10^{.......}} \;\wedge\; . \geq 1$$

$$\equiv .... = \frac{\sum_{k=\,.}^{N} D[k] \times 10^k}{10^{.}} \;\wedge\; . \geq 1$$

$$\equiv r = \frac{\sum_{k=\,.}^{N} D[k] \times 10^k - 10^{.} \times ......}{10^{.}} \;\wedge\; . \geq 1$$

$$\equiv r = \frac{\sum_{k=\,.......}^{N} D[k] \times 10^k}{10^{.}} \;\wedge\; . \geq 1$$

2 pt

VÉRIFICATION

$$\overbrace{r = \underbrace{\frac{\sum_{k=\ldots\ldots\ldots}^{N} D[k] \times 10^k}{10^{\cdot}}}_{①} \wedge\ i \geq 0}^{\psi_1} \bigwedge \overbrace{i > 0}^{test} \overset{?}{\Longrightarrow} \overbrace{r = \underbrace{\frac{\sum_{k=\ldots\ldots\ldots}^{N} D[k] \times 10^k}{10^{\cdot}}}_{①} \wedge\ \underbrace{i \geq 1}_{test}}^{\psi_2}$$

1 pt

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

SOLUTION

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

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

10 pt

Exercice 2 : Calculs de factorielle

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

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

1.5 pt

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

PROGRAMME

```
r:= ...    ; i:= ...    ;

while( ................   ){

    ....................    ;

    ....................    ;
}
```

0.5 pt

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

$$q_0 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} q_1 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots} q_s$$

$$q_1 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots} q_2 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} q_3 \xrightarrow{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} q_1$$

1 pt

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.

0.5 pt

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

5.5 pt

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

SOLUTION

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

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

$$\overbrace{0 \leq i \ \wedge \ i \leq N \ \wedge \ r = fac(i)}^{\psi_1} \bigwedge \overbrace{i < N}^{test} \stackrel{?}{\Longrightarrow} \overbrace{\underbrace{0 \leq i+1}_{(*)} \ \wedge \ \underbrace{i+1 \leq N}_{(**)} \ \wedge \ \underbrace{(i+1) \times r = fac(i+1)}_{(***)}}^{\psi_2}$$

Preuve de $(*)$**:** $0 \leq i \Longrightarrow 0 < i+1 \stackrel{\text{OK}}{\Longrightarrow} 0 \leq i+1$ $\square$

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

Preuve de $(***)$**:**

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

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

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

1 pt

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

SOLUTION

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

10 *pt*

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

PROGRAMME

```
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

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

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).

$$q_1 \xrightarrow{\;..............\;} q_s$$

$$q_0 \xrightarrow{\;..\;} q_1$$

$$q_1 \xrightarrow{\;..............\;} q_2 \quad q_2 \xrightarrow{\;... \leftarrow ....\;} q_3 \quad q_3 \xrightarrow{\;... \leftarrow ....\;} q_4 \quad q_4 \xrightarrow{\;.. \leftarrow ....\;} q_1$$

6 *pt*

Q2. **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 :

$$y = ... \;\wedge\; .... = fib(................ - k)$$

où k est une constante que vous devrez déterminer

SOLUTION

1. *Transition* $q_1 \xrightarrow{\;.............\;} q_s$

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

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

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

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

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

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

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

5. *Transition* $q_1 \xrightarrow{i \neq 0} q_2$ Vérification des invariants choisis *Il s'agit de vérifier que limplication suivante est valide.*

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

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

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 $n^{ème}$ terme de la suite de Fibonacci avec $n = N - i$ $\square$

1 pt

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

SOLUTION

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

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

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 ?