NOM .... Prénom ....

RICM3 – Automates et Grammaires

Durée : 1h00, sans documents.

- Tous les appareils électroniques sont interdits à l'exception des montres, et encore, à condition qu'elles ne communiquent pas.
- Le barème est donné à titre indicatif.
- Le sujet est sur 20 points et comporte 2 exercices indépendants.

SUJET A

Exercice 1 : Fonctions tabulées en bijection

11 pt

Un tableau $G[0..N-1]$ permet de représenter une fonction g définie sur $[0..N-1]$. Il suffit de placer dans la case k du tableau G la valeur de la fonction g en k.

Exemple : Considérons $g(k) = (k+2)$ *mod* N. Le tableau G contient les valeurs précalculées de g.

k	0	1	2	3	4	...	k	...	$N-2$	$N-1$
$G[k]$	2	3	4	5	6	...	$g(k)$	...	0	1

En pratique on utilise la technique des fonctions tabulées pour calculer rapidement $\sin, \sin^{-1}$,... au moyen de tableaux préremplis SIN, SIN_INV,....

Dans cet exercice on considère des fonctions g et f de $[0..N-1] \to [0..N-1]$, tabulées respectivement dans les tableaux G et F. On aimerait déterminer si g et f forment une bijection, c'est-à-dire si $\forall k \in [0..N-1],\ g(f(k)) = k$. Avec les versions tabulées G et F des fonctions, cela devient

G et F forment une bijection si

$$\forall k \in [0..N-1],\ G[F[k]] = k$$

0.5 pt **Q1.** Complétez le tableau F de sorte que G et F forment une bijection :

k	0	1	2	3	4
$G[k]$	2	3	4	0	1

k	0	1	2	3	4
$F[k]$	..	..	..	..	..

On considère le programme suivant qui prend en entrée deux tableaux $G[0..N]$ et $F[0..N]$ déjà remplis et qui répond vrai si les tableaux sont en bijection et faux sinon.

PROGRAMME

```
j:= 0 ;
while(j < N){
  if ( G[F[j]] ≠ j)
  then { return false }
  else { j:=j+1 }
}
return true
```

1 pt **Q2. Automate** Complétez les transitions de l'automate correspondant au programme

Indication : On rappelle que l'instruction « `return valeur` » se traduit par une transition vers l'état de sortie portant une affectation « `result := valeur` ».

$$q_0 \xrightarrow{\;\dots\dots\dots\dots\;} q_1 \xrightarrow{\;\dots\dots\dots\dots\;} q_2 \xrightarrow{\;result\,\dots\dots\dots\dots\dots\;} q_s$$

$$q_1 \xrightarrow{\;\dots\dots\dots\dots\;} q_3 \xrightarrow{\;\dots\dots\dots\dots\dots\dots\dots\;} q_4 \xrightarrow{\;\dots\dots\dots\dots\dots\dots\;} q_1$$

$$q_3 \xrightarrow{\;\dots\dots\dots\dots\dots\dots\dots\;} q_5 \xrightarrow{\;result\,\dots\dots\dots\dots\dots\;} q_s$$

0.5 pt **Q3. Propriété de correction** Complétez la propriété ψ_s qui caractérise le fait que le résultat du programme indique **si G et F forment une bijection**.

$$\psi_s : \begin{cases} result = \dots\dots\dots & \Longrightarrow \; (\forall k \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots) \\ result = \dots\dots\dots\dots & \Longrightarrow \; (\exists k \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots) \end{cases}$$

Indication : Inspirez vous de la propriété de correction de l'algorithme de recherche par dichotomie, vue en cours.

8 pt **Q4. Preuve de correction** Montrez par la méthode de Floyd-Dijkstra-Hoare qu'à la sortie du programme la propriété ψ_s est vérifiée.

Indication :

1. Vous indiquerez toujours la transition considérée. Vous prendrez soin de rédiger complétement une *transition test* et une *transition affectation, la vérification et sa conclusion*.
2. Vous aurez besoin de réécrire les $\forall$ sous une forme qui fait apparaître explicitement le terme qui vous intéresse. Aidez vous de l'équivalence

   $\forall k \in [a..b], \textit{propriété}(k)$ **équivaut à** $\big(\forall k \in [a..b-1], \textit{propriété}(k)\big) \bigwedge \textit{propriété}(b)$
3. On vous conseille de déterminer les invariants dans l'ordre suivant : $\psi_2, \psi_1, \psi_4, \psi_3, \psi_5$
4. Vous prendrez comme invariant en q_1 une propriété de la forme

$$\psi_1 : \begin{cases} \dots\dots\dots\dots \\ \forall k \in [\dots\dots\dots\dots],\; G[F[k]] = k \end{cases}$$

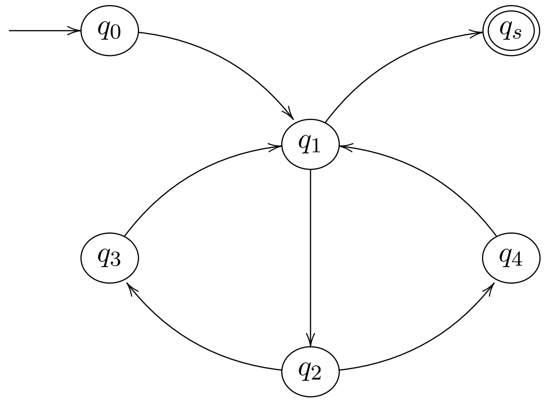
1 pt **Q5. Conditions d'utilisation** En déduire les conditions d'utilisation du programme.

9 pt Exercice 2 : Preuve de correction partielle de l'algorithme d'exponentiation rapide

Algorithme

```
x:=A ; y:=N ; r:=1 ;
while(y>0){
  if (y%2=0){ x:=x*x; y:=y/2; }
  else { r:=r*x; y:=y-1; }
}
```

$\frac{}{1\,pt}$ **Q1.** Donnez les transitions de l'automate pour qu'il corresponde au programme précédent.



$\frac{}{1\,pt}$ **Q2.** Complétez l'exécution de l'automate en adoptant la présentation ci-dessous.

a) Pour les valeurs $A = 3,\ N = 3$

état	q_0	q_1	q_2	....	....	....	....	....	....	....	....	....
x	..	3	..	..	..	..	..	..	..	..	..	..
y	..	3	..	..	..	..	..	..	..	..	..	..
r	..	..	..	..	..	..	..	..	..	....	....	....

b) Même question pour les valeurs $A = 2,\ N = 0$

$\frac{}{0.5\,pt}$ **Q3.** Donnez la propriété de correction qui exprime qu'à la sortie de l'automate la variable `r` contient la valeur de A^N.

$\frac{}{5\,pt}$ **Q4. Preuve de correction partielle (soignez la rédaction !)** Donnez et démontrez l'implication associée à chaque transition et donnez les 5 invariants $(\psi_s, \psi_1, \psi_3, \psi_4, \psi_2)$ associés aux états q_s, q_1, q_3, q_4, q_2.

Indication : $\psi_1 \overset{def}{=} r\ldots\ldots\ldots = A^N \ \wedge\ \ldots \geq \ldots$

$\frac{}{1.5\,pt}$ **Q5.** En déduire les conditions d'utilisations du programme.