Université de Lorraine
DIPLÔME: Telecom Nancy 2A - Apprentissage
Épreuve: MVSI Première épreuve (session 1)
Durée du sujet : 2 h 00
Date : Mercredi 11 décembre 2019 de 14 h 00 à 16 h 00
Nom du rédacteur : Dominique Méry
Documents personnels autorisés

Il est recommandé de bien lire les questions. Les explications et les justifications doivent être aussi simples et claires que possible. Les documents sont autorisés à l'exclusion des documents qui vous seraient transmis durant l'épreuve. Le sujet comprend cinq (5) exercices.

Second écrit

Exercice 1

Soit l'annotation suivante. On suppose que a et b sont des constantes entières et que x, y, z et t sont des variables entières.

$$\ell_1 : x = a \wedge z = x \wedge y = b * b \wedge t = b$$
$$y := x * z + y + z * t$$
$$\ell_2 : y = (x + t)^2$$

Question 1.1 Donner une condition liant a et b pour que cette annotation soit correcte.
On rappelle que l'annotation est définie comme suit
$\forall v, v'.P_\ell(v) \wedge cond_{\ell,\ell'}(v) \wedge v' = f_{\ell,\ell'}(v) \Rightarrow P_{\ell'}(v')$. Vous devez répondre en énonçant et en démontrant les *conditions de vérification* c'est-à-dire en indiquant les differents pas de transformation.

Question 1.2 Nous allons utiliser la fonctionnalité de traduction d'un algorithme PlusCal en un module TLA, pour vérifier l'annotation précédente. Ecrire un module TLA contenant une expression de l'algorithme et ensuite on donnera la question à poser pour vérifier que cette annotation. On rappelle qu'un algorithme peut s'écrire ainsi en PlusCal.

```
1 —algorithm xxx {
2 variables ???;
3 {
4 ......
5 l1: ...
6 l2:...
7 .....
8 };
9 }
```

Exercice 2

On suppose que a et b sont des constantes entières et que x, y et z sont des variables entières. Soit l'annotation suivante:

$$\ell_1 : x = a \wedge y = b * b \wedge z = y$$
$$y := x * z$$
$$\ell_2 : y = (z + x)^2$$

Question 2.1 Donner une condition pour que cette annotation soit correcte.

Question 2.2 On décide d'utiliser Rodin et le langage Event-B pour vérifier cette propriété d'annotation correcte. Ecrire les éléments suivants:

- la liste des variables de la machine
- L'invariant
- les événements
- les axiomes à utiliser.

Exercice 3

Soit p un nombre positif différent de 0, 7, 14, 21, 28, ..., c'est-à-dire non multiple positif de 7.

$$\begin{array}{l}\ell_1 : x = p \wedge y = 35 \wedge z = 35 * p \wedge x = y \\ x := p * y \\ \ell_2 : x = z \wedge \wedge z = 35 * p\end{array}$$

Etudier la correction de cette annotation et prÃ©ciser si elle est correcte ou non.

Exercice 4

Soit l'annotation suivante:

$$\begin{array}{l}\ell_1 : x = a \wedge y = b * b \wedge z = y \wedge t = b \wedge y = t^2 + x^2 + z \\ y := x * z + z * t \\ \ell_2 : y = (t + x)^2\end{array}$$

On suppose que a et b sont des constantes entièresnon nuls et que x, y, z et t sont des variables entières. On rappelle que les conditions de vérifications sont énoncées comme suit:

$\forall v, v'.P_\ell(v) \wedge cond_{\ell,\ell'}(v) \wedge v' = f_{\ell,\ell'}(v) \Rightarrow P_{\ell'}(v')$. Vous devez répondre en énonçant et en démontrant les *Conditions de vérification.*

Question 4.1 Montrer que cette annotation est correcte ou incorrecte.

Question 4.2 On change $y := x * z + z * t$ en $y := x * x + t * t$. Montrer que cette annotation est correcte ou incorrecte et de même nature que l'annotation de la question précédente.

Exercice 5 Nous proposons d'analyser un algorithme dont les annotations sont partielles.

PRECONDITION [...]
POSTCONDITION [...]
LOCAL VARIABLES $int\ v, w, t, u;$
$l0 : [...]$
$u = 0; z = 0; v = 0; w = 1; t = 3;$
$l1 : w = 3 * u + 1 \wedge v = 3 * u^2 \wedge z + v + w = (u + 1)^3 \wedge v + t = 3 * (u + 1)^2 \wedge t = 3 * (2 * u + 1) \wedge u = 0$
WHILE $(u < x)$
$l2 : [...]$
$z = z + v + w;$
$l3 : \left(\begin{array}{l} w = 3 * u + 1 \wedge v = 3 * u^2 \wedge z = (u + 1)^3 \wedge v + t = 3 * (u + 1)^2 \\ \wedge t = 3 * (2 * u + 1) \wedge [...] \end{array} \right)$
$v = v + t;$
$l4 : \left(\begin{array}{l} w = 3 * u + 1 \wedge v = 3 * (u + 1)^2 \wedge z = (u + 1)^3 \wedge v = 3 * (u + 1)^2 \\ \wedge t = 3 * (2 * u + 1) \wedge [...] \end{array} \right)$
$t = t + 6;$
$l5 : \left(\begin{array}{l} w = 3 * u + 1 \wedge v = 3 * (u + 1)^2 \wedge z = (u + 1)^3 \wedge v = 3 * (u + 1)^2 \\ \wedge t - 6 = 3 * (2 * u + 1) \wedge [...] \end{array} \right)$
$w = w + 3;$
$l6 : [...]$
$u = u + 1;$
$l7 : \left(\begin{array}{l} w = 3 * u + 1 \wedge v = 3 * u^2 \wedge z + v + w = (u + 1)^3 \wedge v + t = 3 * (u + 1)^2 \\ \wedge z = u^3 \wedge t = 3 * (2 * u + 1) \wedge 0 \leq u \wedge u \leq x \end{array} \right)$
END-WHILE
$l8 : w = 3 * x + 1 \wedge v = 3 * x^2 \wedge z + v + w = (x + 1)^3 \wedge v + t = 3 * (x + 1)^2 \wedge z = x^3$

Au cours d'une expérimentation avec l'outil ToolBox, on peut extraire cet invariant à partir de l'annotation et on peut le vérifier.

$$
\begin{array}{l}
pc \in L \\
z, v, w, t, u \in \mathbb{Z} \\
pc = "l0" \Rightarrow [\ldots] \\
pc = "l1" \Rightarrow \left(\begin{array}{l} w = 3 * u + 1 v = 3 * u^2 \wedge z + v + w = (u+1)^3 \wedge v + t = 3 * (u+1)^2 \\ \wedge t = 3 * (2 * u + 1) \wedge u = 0 \end{array} \right) \\
pc = "l2" \Rightarrow [\ldots] \\
pc = "l3" \Rightarrow [\ldots] \\
pc = "l4" \Rightarrow [\ldots] \\
pc = "l5" \Rightarrow [\ldots] \\
pc = "l6" \Rightarrow [\ldots] \\
pc = "l7" \Rightarrow \left(\begin{array}{l} w = 3 * u + 1 \wedge v = 3 * u^2 \wedge z + v + w = (u+1)^3 \wedge v + t = 3 * (u+1)^2 \\ \wedge z = u^3 \wedge t = 3 * (2 * u + 1) \wedge 0 \leq u \wedge u \leq x \end{array} \right) \\
pc = "l8" \Rightarrow w = 3 * x + 1 \wedge v = 3 * x^2 \wedge z + v + w = (x+1)^3 \wedge v + t = 3 * (x+1)^2 \wedge z = x^3
\end{array}
$$

Question 5.1 Enoncer la précondition et la postcondition de cet algorithme..

Question 5.2 En vous aidant des annotations présentes, complétez les annotations qui manquent (ℓ_0, ℓ_2, ℓ_6) et compléter les annotations en ℓ_3, ℓ_4 et ℓ_5. Pour cela, vous utiliserez la condition de vérification suivante

$P_\ell(a) \wedge cond_{\ell,\ell'}(a) \wedge a' = f_{\ell,\ell'}(a) \Rightarrow P_{\ell'}(a')$ où a est la liste des variables de votre algorithme; cette condition de vérification justifiera votre réponse.

Question 5.3 On traduit cet algorithme sous la forme d'un module TLA en introduisant la variable implicite du contrôle notée pc. Donner la question à poser au système pour vérifier la correction partielle de cet algorithme.

Question 5.4 Expliquer comment peut-on procéder à partir de ces annotations pour déduire l'absence d'erreurs à l'exécution. On supposera qu'il existe un ensemble des entiers informatiques $\mathbb{I}_i = min..max$ où min est une valeur minimale négative et max est une valeur maximale positive des variables.

Question 5.5 Déduire un algorithme qui calcule la racine cubique entière d'un nombre x c'est-à-dire la valeur a satisfaisant la propriété $a^3 \leq x < (a+1)^3$.